

**3D Face Structure Extraction from Images at Arbitrary Poses and under
Arbitrary Illumination Conditions**

A Thesis

Submitted to the Faculty

Of

Drexel University

By

Cuiping Zhang

In partial fulfillment of the

Requirements for the degree

Of

Doctor of Philosophy

October 2006

Acknowledgments

I would like to thank my advisor, Dr. Fernand S. Cohen for his support through all these years. Without his common-sense, knowledge and guidance, I would never have finished.

I am grateful to all of my dissertation members, including Dr. Oleh J. Tretiak, Dr. Jaudelice Cavalcante de Oliveira, Dr. Ali Shokoufandeh, and Dr. Ko Nishino. Thank them for serving on my committee and for their advice.

My thanks go to my close friends, which are too many to mention. They took time to listen to me and give me advice and asked me over and over again “How is your research recently?” .

I would give my deepest love to my mother and father, who always support me unconditionally all these years!

Table of Contents

TABLE OF CONTENTS	IV
TABLE OF FIGURES.....	VII
TABLE OF TABLES	IX
ABSTRACT	X
CHAPTER 1 : INTRODUCTION.....	1
1.1 INFLUENTIAL FACTORS FOR FACE APPLICATIONS	2
1.2 PROBLEMS AND MOTIVATIONS	3
1.3 FACE IMAGES USED IN THIS WORK	5
1.4 CONTRIBUTION OF THE THESIS	5
1.5 OUTLINE OF THIS THESIS	7
CHAPTER 2 : COMPONENT-BASED ACTIVE APPEARANCE MODEL.....	8
2.1 LITERATURE	8
2.2 PRELIMINARY BACKGROUND: THE ACTIVE APPEARANCE MODEL.....	12
2.3 COMPONENT-BASED AAM	16
2.4 LOCAL PROJECTION MODELS FOR FACE BOUNDARY DETECTION.....	20
2.5 COMPONENT-BASED SEARCH	23
2.6 EXPERIMENTS.....	25
2.6.1 <i>Component-based Search</i>	26
2.6.2 <i>Face Contour Detection with Local Projection Models</i>	28
2.7 CONCLUSIONS	31
CHAPTER 3 : 2D FACE MODELING WITH A HYBRID CONSTRAINED OPTIMIZATION ALGORITHM	32
3.1 LITERATURE	32
3.2 NORMALIZED INVERSE COMPOSITIONAL AAM ALGORITHM.....	36
3.3 UNBIASED ERROR EVALUATION FUNCTION	39
3.4 DIRECT SHAPE ESTIMATE FROM MOTION ESTIMATION.....	41
3.5 CONSTRAINED GRADIENT DESCENT OPTIMIZATION	44
3.6 EXPERIMENT RESULTS AND DISCUSSION	46
3.6.1 <i>Constrained Hybrid Model Fitting Optimization</i>	46
3.6.2 <i>Experiments on the JAFFE face database</i>	51
3.7 CONCLUSION	54
CHAPTER 4 : FROM 2D TO 3D: 3D FACE STRUCTURE EXTRACTION.....	55
4.1 LITERATURE	55
4.1.1 <i>Shape Reconstruction by Modeling</i>	56

4.1.1.1 Generic Face Modeling	58
4.1.1.2 Statistical Face Modeling	63
4.2 OUR APPROACHES	68
4.2.1 3D Generic Model	70
4.2.2 Weak Perspective Projection	71
4.2.3 2D Feature Extraction with the View-based AAMs	74
4.2.4 Cubic Morphing: Revisited	77
4.2.4.1 Basic Cubic Polynomial Function	77
4.2.4.2 Cubic Morphing Reformulated as a Linear Operation	78
4.2.4.3 Regulate Cubic Morphing Parameters	79
4.2.5 Distance Map: Revisited	80
4.3 MORPHING AND POSE PARAMETER ESTIMATION	83
4.3.1 Partial Linear Optimization Algorithm	83
4.3.2 Optimization with Marquardt-Levenberg algorithm	85
4.3.3 Incorporate Contour Constraints to the Optimization	87
4.3.4 Refine the Parameter Estimation with Distance Mapping	90
4.4 EXPERIMENTS	91
4.4.1 View-based AAM	91
4.4.2 3D Modeling Experiments	93
4.4.2.1 Partial Linear Optimization versus LM Optimization Algorithms	93
4.4.2.2 Incorporation of Face Contour Constraints	95
4.5 CONCLUSIONS	97
CHAPTER 5 : FROM 2D TO 3D: ILLUMINATION-FREE TEXTURE EXTRACTION WITH THE SPHERICAL HARMONIC ILLUMINATION MODEL	99
5.1 LITERATURE	100
5.1.1 When Illumination is not considered: Face Texture Mapping Techniques	100
5.1.2 Basic Illumination Models for Photorealistic Rendering in Computer Graphics	102
5.1.2.1 Light Sources	103
5.1.2.2 Basic Illumination Models	103
5.1.3 Illumination Modeling for Face Recognition	105
5.1.3.1 PCA-based Low-dimensional Linear Subspace Representation	107
5.1.3.2 Illumination Modeling based on Theoretical Analysis of Lambertian Reflectance	107
5.2 PRELIMINARY BACKGROUND: SPHERICAL HARMONICS AND THEIR APPLICATIONS IN ILLUMINATION MODELING	110
5.2.1 The Spherical Harmonic Analysis	111
5.2.2 Illumination modeling by Spherical Harmonic Analysis: from the Lighting function to the Reflectance Function and Basis Images	113
5.3 EXTRACTION OF ILLUMINATION INVARIANT TEXTURE MAP FROM IMAGES	116
5.3.1 View-dependent Illumination Editing and Normalization	117
5.3.2 Extraction of View-independent Illumination-free Texture Map	119
5.4 EXPERIMENTS	121
5.4.1 View-based Illumination Analysis	122
5.4.1.1 Illumination Editing	123
5.4.1.2 Rotation in the same Illumination Environment	124

5.4.1.3 Illumination Regulation by “Copying” Illumination Effect	125
5.4.1.4 Synthesis of Novel Faces	126
5.4.2 <i>View-Independent Illumination Analysis</i>	127
5.4.2.1 Extraction of Texture Map.....	127
5.4.2.2 Synthesis of Novel Faces	130
5.5 <i>Conclusion</i>	131
CHAPTER 6 : 3D FACE RECOGNITION FOR IMAGES AT ARBITRARY POSES AND UNDER ARBITRARY ILLUMINATION CONDITIONS	133
6.1 LITERATURE	134
6.2 FACE RECOGNITION BASED ON THE EXTRACTED 3D STRUCTURE AND THE ILLUMINATION-FREE TEXTURE MAP	138
6.2.1 <i>Face Modeling Phase</i>	138
6.2.2 <i>Testing Phase</i>	139
6.3 EXPERIMENTS AND DISCUSSIONS	142
6.3.1 <i>A Typical Face Recognition Example</i>	144
6.3.2 <i>Complete Recognition Results for Different Pose Categories</i>	149
6.4 CONCLUSIONS	159
CHAPTER 7 : CONCLUSIONS AND FUTURE WORK.....	160
7.1 INCORPORATING TEXTURE INFORMATION TO EXTRACT FACE FEATURES.....	160
7.2 INCORPORATING ILLUMINATION INFORMATION TO REFINE SURFACE DETAILS	162
7.3 TRACKING FACES	163
7.4 FACIAL EXPRESSION MODELING AND RECOGNITION	163
LIST OF REFERENCES.....	165
APPENDIX A: AN EXISTING PROBLEM ABOUT THE TAGENT SPACE COORDINATE ALIGNMENT ALGORITHM.....	172
APPENDIX B: DIRECT EXHAUSTIVE SEARCH FOR INITIAL MODEL PARAMETERS .	174
APPENDIX C: 3D FACE MODEL NORMALIZATION	176
APPENDIX D: SYNTHESIZED FACE IMAGES BY VARYING THE FIRST SHAPE MODE	178
APPENDIX E: SYNTHESIZED FACE IMAGES BY VARYING THE FIRST TEXTURE MODE	179
VITA	180

Table of Figures

Figure 2.1 (a) Definition of landmark points. (b) Face mesh. (c) Shapeless texture. (d) Base face mesh.....	13
Figure 2.2 (a) Landmark points inside face are divided into 3 groups. (b) Left eyebrow and eye in one group. (c) Right eyebrow and eye. (d) The nose and mouth.....	18
Figure 2.3 Updating rule for the ASM algorithm.....	20
Figure 2.4 Building a local projection model.....	21
Figure 2.5 Triangulation of face landmark points: a) Original shape of the person in Fig. 2.1. b) Mean shape of training set.....	22
Figure 2.6 A parallelogram associated with a triangle on the face boundary. a) Original image frame. b) Mean shape frame. c) Standard pair.....	23
Figure 2.7 Flowchart of steps in an iterative AAM search.....	25
Figure 2.8 Sample images in our face database.....	26
Figure 2.9 AAM (top row) versa AAM_CA (bottom row). (a) Training set. (b) Test set. c) JAFFE.....	27
Figure 2.10 AAM_CA (top row) versa AAM_CA_LPM (bottom row). (a) Training set. (b) Test set. (c) JAFFE.....	28
Figure 2.11 Curves of convergent rate versa error threshold. (a) Training set. (b) Test set. (c) JAFFE database.....	31
Figure 3.1 Example of a piece-wise affine transform from the image frame (left) to the base frame (right).....	40
Figure 3.2 From left to right: a) Synthesized image. b) Input image. c) Synthesized face overlapped on the original face with current landmark points.....	43
Figure 3.3 Comparison of hybrid search and original inverse compositional AAM search. (a)-(c) Hybrid search process at iteration 1, 2 and 6. (d) Inverse Compositional AAM search. (e) Constrained hybrid search. (f) Evolution of error curve.....	47
Figure 3.4 Fitting errors on (a) Training set. (b) Test set.....	49
Figure 3.5 Cumulative functions.....	50
Figure 3.6 Model fitting results: a) Inverse compositional algorithm. b) Constrained hybrid algorithm.....	52
Figure 3.7 Model fitting errors on JAFFE.....	53
Figure 3.8 Cumulative density functions.....	53
Figure 4.1 Candide-3 face model: a) Frontal view. b) Profile view.....	59
Figure 4.2 Face model created at Univ. of Washington. a) Frontal view. b) Profile view.....	60
Figure 4.3 Revised generic model: (a) Mesh model. (b) Solid model. (c) Selected features for structure estimation.....	70
Figure 4.4 Camera model.....	72
Figure 4.5 The 3D face model and 4 view-based AAM models.....	75
Figure 4.6 Four face images of different viewpoints on the first row. Their corresponding distance maps are shown on the second row.....	82
Figure 4.7 Face meshes used in the view-based AAM: pose category 2 and 3 from left to right.	

.....	87
Figure 4.8 Candidates for contour points on the generic model viewed from two different angles	89
Figure 4.9 Nine normal lines on the face contour	89
Figure 4.10 Average faces for pose category 1 to 4 (from left to right)	92
Figure 4.11 Face alignment results of different poses	93
Figure 4.12 Modeling results: (a)(d)(g): the partial linear method shown as frontal, half-profile and full profile; (b)(e)(h) the LM method shown from three angles; (c)(f)(i): the regularized LM method from three angles	94
Figure 4.13 Modeling results with and without the contour constraint for pose category 2 ...	96
Figure 4.14 Modeling results with and without the contour constraint for pose category 3 ...	97
Figure 5.1 The generic face model (left) and the texture space (right)	120
Figure 5.2 Four views of a texture face based on the reconstructed 3D face	122
Figure 5.3 Simulation of different illumination effects by editing coefficients directly	123
Figure 5.4 Simulation of different illumination effects another viewing angle	124
Figure 5.5 Simulation of rotating the face in the same illumination environment	125
Figure 5.6 Illumination "copying" example: top row: two original images; bottom row: two regulated images that copied illumination effect from each other	126
Figure 5.7 Synthesized images (bottom row) versus original images (top row)	127
Figure 5.8 Original face images on the texture space	128
Figure 5.9 Weight functions for the images in Fig.5.8	129
Figure 5.10 An illumination-free texture map after 20 iterations	130
Figure 5.11 Synthesized images (bottom row) versus original images	131
Figure 5.12 Synthesized images of arbitrary poses	131
Figure 6.1 Face modeling flowchart	139
Figure 6.2 One example view of the 38 people	143
Figure 6.3 A typical test image	145
Figure 6.4 Comparison of the synthesized images and the test image: a) Synthesized illumination-free image. b) Illuminated image. c) Test image	145
Figure 6.5 More examples of texture matching	147
Figure 6.6 Three matching error curves	148
Figure 6.7 Example views of synthesized images and test images for pose 2 to pose 4 (from first row to last row)	149
Figure 6.8 Face recognition based on distance map error	150
Figure 6.9 A misclassified case: Left image is the test image. The best match is the person in the right image.	151
Figure 6.10 Face recognition based on texture error	152
Figure 6.11 Face recognition based on illumination-normalized texture error	152
Figure 6.12 Face recognition based on the combined error	154
Figure 6.13 Recognition results for pose categories 2 to 4 (from the leftmost column to the last column)	155
Figure 6.14 Recognition results based on raw texture error for pose categories 1 to 4	158

Table of Tables

Table 2.1 Average point to edge error (contour points excluded)	27
Table 2.2 Average point to edge error(contour points only).....	29
Table 3.1 Average point to edge error for different algorithms	51
Table 3.2 Average point to edge error for different algorithms	53
Table 4.1 Azimuthal range for 4 different view-based models.....	75
Table 4.2 Our face database	91
Table 4.3 Average modeling errors for different methods.....	95
Table 6.1 Misclassification rate for 38 people	156

Abstract

3D Face Structure Extraction from Images at Arbitrary Poses and under Arbitrary Illumination Conditions

Cuiping Zhang

Fernand S. Cohen, Supervisor, Ph.D.

With the advent of 9/11, face detection and recognition is becoming an important tool to be used for securing homeland safety against potential terrorist attacks by tracking and identifying suspects who might be trying to indulge in such activities. It is also a technology that has proven its usefulness for law enforcement agencies by helping identifying or narrowing down a possible suspect from surveillance tape on the crime scene, or quickly by finding a suspect based on description from witnesses.

In this thesis we introduce several improvements to morphable model based algorithms and make use of the 3D face structures extracted from multiple images to conduct illumination analysis and face recognition experiments. We present an enhanced Active Appearance Model (AAM), which possesses several sub-models that are independently updated to introduce more model flexibility to achieve better feature localization. Most appearance based models suffer from the unpredictability of facial background, which might result in a bad boundary extraction. To overcome this problem we propose a local projection models that accurately locates face boundary landmarks. We also introduce a novel and unbiased cost function that casts the face alignment as an optimization problem, where shape constraints obtained from direct motion estimation are incorporated to achieve a much higher convergence rate and more accurate alignment. Viewing angles are roughly categorized to four different poses, and the customized view-based AAMs align face images in different specific pose categories. We also attempt at obtaining individual 3D face

structures by morphing a 3D generic face model to fit the individual faces. Face contour is dynamically generated so that the morphed face looks realistic. To overcome the correspondence problem between facial feature points on the generic and the individual face, we use an approach based on distance maps. With the extracted 3D face structure we study the illumination effects on the appearance based on the spherical harmonic illumination analysis. By normalizing the illumination conditions on different facial images, we extract a global illumination-invariant texture map, which jointly with the extracted 3D face structure in the form of cubic morphing parameters completely encode an individual face, and allow for the generation of images at arbitrary pose and under arbitrary illumination.

Face recognition is conducted based on the face shape matching error, texture error and illumination-normalized texture error. Experiments show that a higher face recognition rate is achieved by compensating for illumination effects. Furthermore, it is observed that the fusion of shape and texture information result in a better performance than using either shape or texture information individually.

CHAPTER 1 : INTRODUCTION

Human face recognition is an inter-disciplinary subject that utilizes various technologies to analyze and identify human faces from images, video sequences, range data etc. [Face recognition has an extensive range of potential applications](#). For law enforcement agencies it can be of great assistance to hunt down criminals. The verification of a real person and his ID photo in the driver's license or passport would somehow prevent ID fraud. In the future, face recognition is expected to be the core technology in intelligent surveillance systems for banks and customs. With the blooming of online business, companies have to deal with more and more online frauds as personal information in a traditional sense are easy to be hacked. Real time face verification with input video sequences from a webcam seems to be a promising method for increased transaction safety.

Besides face, other features that have been explored for recognition are fingerprint, iris, retinal, vein, voice etc. Together, these automated recognition methods belong to the big category of biometrics, which in general refers to identifying a person based on a physiological or behavioral characteristic. Compared to popular biometric features like fingerprint and iris, face images are easy to obtain without the cooperation or awareness from the targeted people.

It is the most natural thing to identify a person by his face. It is a fundamental function of a human being to memorize and identify thousands of different faces. However, this task is more difficult for computers of current generation. This is caused by their different underlying working mechanism. Computers are good at complicated computation, but have limited logic ability. Scientists are still trying hard to understand how a human being's brain works.

Physiological and psychological studies of the human vision system lead to some conclusions that stimulate the development of face recognition systems. First of all, vision data needs to be encoded compactly and later reconstructed (as a brain does) and classified based on the a priori knowledge of all classes. A human brain dynamically interacts with the outside world to update its knowledge. Secondly, both the overall appearance and facial parts contribute to the recognition process. Eyes and mouths are believed to be more important than noses. However for a specific person, the most active feature varies. What a caricature does is to magnify the most distinct feature of a face, so that people can easily recognize the person.

1.1 Influential Factors for Face Applications

In general, a face recognition algorithm extracts certain information from face images or video sequences of unknown identity and look up a face database for its closest match. Usually this process is composed of the following steps: segment the face from the image(s), extract and analyze features, followed by the verification or identification. Different face applications might focus on different steps.

For almost all face applications, following is a summary of some important factors:

- 1) Face image quality. Unlike face images taken under controlled environments, video surveillance usually outputs face images of low resolution, therefore makes it hard to obtain face details.
- 2) Unpredictable background. To separate a face from a complicated background is not an easy task. For video sequences, this could be relieved with motion estimation algorithms.
- 3) Lighting condition. Face images of the same person might look very different under

different illumination environments, especially when shadow is present.

- 4) Pose of the face. Even under a controlled condition, different people might still show slightly different poses, let alone images from video surveillance tapes.
- 5) Facial expression. It is easier to analyze the facial expression given the human-computer interaction. However, facial expression presented in one or just several images is hard to separate, causing a lower recognition rate.
- 6) Disguise and partial occlusions. With different hairstyle, mustache, sunglasses and other cosmetic effects, even human eyes might be deceived. These disguises are not reversible.
- 7) Age. Human face transforms slowly with age due to the growth of facial bones and healthy state of facial skin. Aging effect is extremely difficult to predict as it is affected by both internal and external environment.

In reality, no such an almighty face recognition system has been developed to cover all aspects mentioned above. Facial expression recognition and age, gender recognition has more or less developed into independent topics. This thesis mainly considers face alignment and recognition under different poses, backgrounds and lighting conditions.

1.2 Problems and Motivations

At the early stage of face recognition study, the mainstream effort was devoted to extract geometrical features for recognition. This approach is not effective, especially on a big face database. Ever since the 90s, the prevailing algorithms have been those that are based on the whole appearance. Typical algorithms include eigenface algorithms [1], elastic graph matching algorithms [2], Hidden Markov Models [3] and Neural Network algorithms etc. In a

survey paper [4] in 1993, appearance-based methods are compared with geometrical feature-based methods and the author concluded that appearance-based methods have better performance. Model matching algorithm is one of the earliest appearance-based methods. With years, it has evolved from simple rigid model matching algorithms to more complicated flexible models. Algorithms of this kind share two common features: shape and texture of a face are separately encoded; face modeling task is accomplished by seeking for the optimal parameters that can synthesize a face which best imitates the unknown face. In this thesis, flexible appearance-based models mainly serve to align facial feature points of an unknown face.

Face is a nearly convex three dimensional (3D) object. So naturally one would expect to see a lot of face applications from the 3D perspective. However before the late 90s, the overwhelming algorithms were carried on two dimensional (2D) face images without explicitly recovering the underlying 3D face geometry. One of the main causes is the inability to reconstruct a 3D face accurately. Early 3D research in computer vision includes 3D shape from X, where X refers to shading, stereo etc. At the same time, people in computer graphics community have the continuous fever of working toward synthesis of novel realistic view and advanced 3D animation. In fact, some big achievements have been made by the computer graphics community. Triangle mesh is widely adopted to represent face structure.

The big breakthrough is the 3D morphable model [5] proposed by T. Vetter in 1999. 3D morphable model is trained based on real 3D dense face data. It is a generative model that can synthesize new face based on the statistical properties of the training face set. There are extensive studies on this model and it proves to be very successful for face recognition on

large databases with various poses and illuminations. However, the 3D morphable model needs a 3D face database which is generated with special instruments like a 3D laser scanner.

The modeling process is still too slow for most real face applications.

This thesis will address several aspects of a face recognition system: alignment of facial feature points based on improved appearance models; 3D face modeling using cubic morphing and 2D alignment results. [The recovered 3D model makes it possible to conduct illumination modeling based on the spherical harmonics analysis \[6\]](#), a recently proposed illumination model that can model illumination space with only 9 parameters. As applications, our model can generate novel faces under arbitrary views. The extracted shape and texture information could be used separately or together for recognition purpose.

1.3 Face Images Used in This Work

The face images we collected are from open online face databases. All images under study have been preprocessed and segmented to the standard size 256 by 256. All faces have neutral expressions and slight or no occlusions. Age is not considered though most of the faces are female and male adults of different ethnic backgrounds. Besides Caucasian, some Asian and Black faces are selected. There is no limitation with regard to the face backgrounds, illuminations and poses. In a word, it is a very diverse face database.

1.4 Contribution of the Thesis

This thesis makes the following contributions:

- 1) For the task of aligning face feature points, the standard Active Appearance Model [7] is enhanced by introducing the idea of sub-models for better feature localization. Sub-models

allow for independent local shape updating and they are closely integrated into one global appearance model.

- 2) Local projection models are adopted to accurately locate face boundary landmark points as unpredictable background poses problem for most appearance models.
- 3) Face alignment task is reinterpreted from the perspective of optimization. This thesis presents a novel cost function that is an unbiased evaluation of face alignment quality. Shape constraints from motion estimation and local projection models are added to the optimization function to achieve a much higher convergent rate.
- 4) Face surface reconstruction problem is tackled by cubic morphing and view-based face alignment results. Partial linear optimization and L-M optimization are experimented and compared. Face contour is dynamically generated and added to the optimization process.
- 5) The illumination condition of face images is not controlled. Spherical harmonics are adopted to approximate illumination space for each person. An illumination independent texture map is extracted.

3D face modeling is the best approach to estimate face poses. With a reconstructed 3D face model, synthesis of new images is straightforward. Face illumination is also analyzed based on the 3D face structure. With extra symmetry constraint, illumination could give us shape information about the underlying face just as shape from shading algorithms do. This thesis mainly deals with extraction of 3D shape information and illumination-free texture information and manifests their applications for face synthesis and recognition.

1.5 Outline of this thesis

In the rest of this thesis, the problem of face reconstruction and recognition will be addressed in detail. This thesis starts with the face alignment task on face images in Chapter 2 and Chapter 3. In Chapter 2, a Component-based Active Appearance Model is proposed to align a morphable face mesh to any face image. Sub-model analysis aims at better local details, while local projection models deals with accurate face boundary detection. Chapter 3 introduces a hybrid constrained optimization method that incorporates several shape constraints for fast and more accurate face alignment. Experiments show that direct shape estimate is very efficient and accelerates the canonical optimization procedure. Chapter 4 addresses the 3D surface reconstruction problem from face alignment results in 2D. For realistic view, face contour is also considered. Chapter 5 emphasizes the illumination model based on the 3D structure from Chapter 4 and the spherical harmonics theory for illumination subspace modeling. Chapter 6 shows how the extracted shape and texture information could be used for synthesis and face recognition purposes. Chapter 7 concludes this thesis and discusses about possible extensions for future work.

CHAPTER 2 : COMPONENT-BASED ACTIVE APPEARANCE MODEL

In this chapter, a novel component-based AAM is presented to align facial feature points on an unknown face image. Without confusion, terms “face modeling (in 2D)” and “face alignment” might be abused to refer to the same task. The AAM statistically models shape and appearance based on principal component analysis. It is a powerful tool for modeling a class of objects such as faces. However, the AAM also suffers from two major drawbacks as a statistical model. First, it is common to see a far from optimal local alignment when attempting to model a face that is quite different from training faces. By adopting three sub-models inside the face area, then combining them with a global AAM, face alignment could achieve both local as well as global optimality. Secondly, it is well known that background information is invariably encoded into AAM’s updating rule, which leads to substantially degraded performance for faces with unseen background. In fact for most appearance model-based algorithms, face contour points are especially hard to locate due to 2 facts: face cheek is almost textureless; face background is unpredictable and unreliable. Local projection models are utilized to accurately locate face contour points. They prove to be effective and computationally efficient by making use of intermediate piecewise affine transforms between face meshes.

2.1 Literature

Detecting a face and aligning facial features are usually the first step for any face recognition system. Therefore it is crucial for all face applications. The earliest approaches favored by

researchers are general low-level edge detection. Later several parametric models that are more specific to human face were developed. For example, circles are adopted to model and locate the eyeball locations from the edge map of a given face image and a parabolic curve is used to extract and describe the mouth [9]. The active contour [10], also known as the Snake algorithm was introduced by M. Kass etc. Starting with an initial position, the parameterized curve is updated iteratively and moved towards an optimal contour in the face image so that an objective energy function is minimized. Usually the energy function models the smoothness of the curve as its internal force and edge strength as its external force. Its applications include edge detection, motion tracking etc. Level set method [11] is a more advanced algorithm compared to Snake. It is a generic numerical method for evolving fronts in an implicit form. It handles topological changes of the evolving interface and defines the problem in one higher dimension. It lays the groundwork of its kind and researchers could add their own constraints for their specific applications in image segmentation, enhancement or registration.

Though the smoothness of the curve is considered into the energy function, Snake has little control over the overall shape. For level set method, some latest papers incorporate shape prior into the level set function and it leads to improved performance. Unlike Snake or level set, statistical models directly learn the shape and/or texture distribution and only allow for admissible shapes/textures. The Active Shape Model (ASM) [12] is such a model. It is proposed by T. Cootes in 1994. Face shape is represented with a set of landmark points. With the help of a subspace analysis method like Principal Component Analysis (PCA), a compact representation of the shape can be obtained. The distribution of shape in the subspace is

learned from a training set. To extract the face shape from an unknown image, the ASM analyzes the local distribution for each landmark point and moves the point so that it is in accordance with a typical local distribution. By confining the overall face shape within the face space, the ASM generates flexible yet well-controlled face shapes.

In 1998, Cootes came up with a similar model, the Active Appearance Model (AAM), which differs from the ASM with its updating mechanism. For each face, a shapeless texture is extracted from the image and projected into the texture subspace. Ultimately face model parameters are then generated from both the shape and texture subspace coefficients. Seeking optimal model parameters for a given face image is an iterative procedure. In each step a fixed gradient decent matrix is used to update the model parameters. As a successor of the ASM, the AAM is computationally efficient and has been intensively studied by many researchers. T. Cootes compared the ASM with the AAM in [13] and concluded that the ASM has better localization, while the AAM has better face texture interpretation.

Several variations of the original AAM algorithm have been proposed. The Active Wavelet Networks (AWN) [14] replaces the AAM's texture model with Gabor networks. Due to the localization property of the wavelet basis, the AWN is less sensitive to illumination and possible partial occlusion. The Texture Constrained – Active Shape Model (TC-ASM) [15] unites the ASM and the AAM under a Bayesian framework. The resulting shape is a hybrid weighted prediction from both the ASM and the AAM. The Direct Appearance Model (DAM) [16] predicts shape parameters directly from texture parameters, based on mutual dependency of shape, texture and model parameters. The Inverse Compositional AAM [17] is proposed as a simple, yet theoretically more correct model. Better performance is reported in terms of rate

of convergence and model fitting accuracy. The Inverse Compositional AAM will be explained in detail in next chapter. M. Stegmann [18] studied various AAM extensions in his thesis and compared different optimization schemes.

Multi-view faces alignment algorithms have been developed for the AAM, the AWN and the DAM [19] [20] [21]. S. Romdhani etc used a nonlinear Kernel PCA [22] based on Support Vector Machines to handle the nonlinear model transformation in their multi-view face ASM algorithm. The AAM is also applied to direct 3D modeling. Li etc [23] presented a 3D face model and iteratively solved 3D model parameters by fitting projected 2D models to video sequence or images of different viewpoints. F. Dornaika etc [24] trained the Candide 3D face model using PCA to track faces in video sequences. In CVPR 2004, J. Xiao etc [25] showed their latest progress with the Inverse Compositional AAM. A 3D morphable model is added to constrain the 2D alignment. 3D and 2D model parameters are solved simultaneously as a result of the real-time aligning procedure.

The original AAM has several inherent drawbacks as a global appearance based model. First, it has a simple linear update rule stemming from a first order Taylor series approximation of an otherwise complex relationship between the model parameters and the global texture difference. Clearly, any factor that contributes to the global texture will affect the AAM's performance (examples are global illumination, partial occlusion, etc.). In a converged AAM modeling scenario, the local alignment results may need further refinement to meet the accuracy requirement of many applications. In some cases, the AAM results in a local minimum where some landmark points are far away from their real locations. It is desirable to pull the AAM out of the local minimum. Secondly, gradient descent information near the face

contour seeps the background pattern in the training set. Hence, the AAM can not perform well for test face images with unseen backgrounds.

This chapter tries to cope with all these problems associated with the AAM with a component-based AAM, which groups landmark points inside the face into three natural components in addition to the globally defined AAM. The independence of the local component AAMs adds flexibility and leads to a more accurate local alignment result. For landmark points on the face contour, a strategy similar to the ASM is adopted. The ASM updates any landmark point by analyzing the profile along local normal direction that needs to be adjusted accordingly in the iterative procedure. Our new method makes full use of what's already available during the AAM procedure. Instead of getting local distributions directly from the original image, the proposed algorithm works directly on the edge map in a standard shape frame. That leads to a huge reduction in computation. Another advantage is that our new projection models are automatically proportioned to the scale of the whole face. Together, the revised projection models with the component-based AAM results in an improved performance, especially on the test set.

2.2 Preliminary Background: the Active Appearance Model

For each face in the training set, 73 key landmark points are picked and the sequence of their coordinates forms a shape vector. Let $\bar{\mathbf{X}}$ be the mean shape of all the training face shapes. For each training image, the face patch inside the convex hull of the landmark points is warped to a base face mesh to form a texture vector. Usually the base face mesh is just the mean face mesh. Fig. 2.1(a) shows landmark points on a face image and the resulting

shapeless texture is in Fig. 2.1(c).

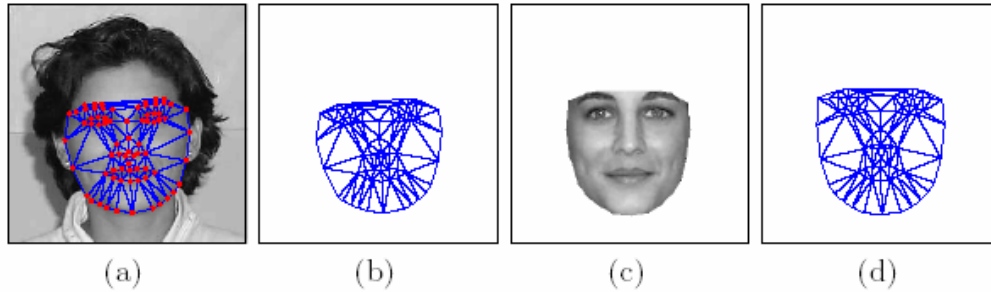


Figure 2.1 (a) Definition of landmark points. (b) Face mesh. (c) Shapeless texture. (d) Base face mesh.

All shape vectors are normalized to a common coordinate system before proceeding to the subspace analysis. In the original AAM, all shapes are aligned to the tangent space of the aligned set mean, as this normalization introduces less non-linearity compared to other methods [8]. This normalization needs to be implemented iteratively. Our experiment exposed a potential convergence problem of this normalization method. As it hasn't been addressed before, Appendix A gives a detailed explanation. To discriminate different coordinate systems, capital letters are used to notate shape and texture vectors in the image frame and low-case letters for the model frame. The coordinate normalization introduces a similarity transform between a face vector \mathbf{X}_{im} in the image frame and the corresponding normalized one \mathbf{x} in the model frame. The similarity transform could be characterized by translations in x and y direction as t_x , t_y , a scaling factor s and a rotation angle θ . These 4 parameters form a 2D pose parameter set for a specific face image. Let the pose parameter set Ψ be $\{s, \theta, t_x, t_y\}$. The texture vector \mathbf{G}_{im} is also scaled and zero centered so that the transformed \mathbf{g} is aligned with the tangent space of the set mean in the model frame. PCA is

adopted to model the shape variation and texture variation in the training set. In the low dimensional shape space and texture space, the shape \mathbf{x} and the texture \mathbf{g} are respectively

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s * \mathbf{b}_s \quad (2-1)$$

and

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g * \mathbf{b}_g \quad (2-2)$$

where \mathbf{P}_s is a matrix whose columns are the principal orthogonal modes of variation of the shape space with \mathbf{b}_s being the vector of shape parameters. \mathbf{P}_g is a matrix describing the modes of variation of the texture space with \mathbf{b}_g being the projected texture parameters. As there might be correlations between shape and texture variations, the vector \mathbf{b}_s and \mathbf{b}_g are concatenated for further de-correlation using PCA.

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \cdot \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} \quad (2-3)$$

where \mathbf{W}_s is a diagonal weight matrix, the vector \mathbf{b} is projected into the subspace as

$$\mathbf{b} = \mathbf{Q} \cdot \mathbf{c} \quad (2-4)$$

The combined parameter vector \mathbf{c} encodes both the shape and texture information. We rewrite \mathbf{Q} in the form of 2 sub-matrices \mathbf{Q}_s and \mathbf{Q}_g as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_s \\ \mathbf{Q}_g \end{pmatrix} \quad (2-5)$$

So that \mathbf{Q}_s has the same number of rows as \mathbf{b}_s . The reconstruction of the shape vector \mathbf{x} and the texture vector \mathbf{g} from the combined parameter vector \mathbf{c} is as follows:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \cdot \mathbf{W}_s \cdot \mathbf{Q}_s \cdot \mathbf{c} \quad (2-6)$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \cdot \mathbf{Q}_g \cdot \mathbf{c} \quad (2-7)$$

The complete appearance model parameter set \mathbf{p} includes 2D pose parameter set Ψ and

the combined parameter vector \mathbf{c} . $\mathbf{p} = \{\Psi, \mathbf{c}\}$. The offsetting and scaling operations needed to transform the texture vector from the image frame to the model frame are easy to estimate, so they are not included.

Searching for an unknown face in an image is equivalent to seeking for the model parameter set that best describes the face. This procedure is realized iteratively. Given the current estimate of the combined parameter vector \mathbf{c} and the 2D pose parameters Ψ , the shape vector \mathbf{X}_{im} in the image frame can be easily computed. We warp the face patch enclosed by \mathbf{X}_{im} to the base mesh and align the resulted texture vector \mathbf{G}_s to the model frame to generate the texture vector \mathbf{g}_s . The difference between \mathbf{g}_s and the model texture \mathbf{g}_m directly reconstructed from the model parameters is:

$$\mathbf{r}(\mathbf{p}) = \mathbf{g}_s - \mathbf{g}_m \quad (2-8)$$

The mean squared error of $\mathbf{r}(\mathbf{p})$ indicates the matching quality. It measures how good the current model parameters describe the unknown face. This thesis follows Cootes' notation for the difference vector here. $\mathbf{r}(\mathbf{p})$ is the texture residue. In some papers, it is also called the difference image though it is not an image in common sense. Notation $\delta\mathbf{g}$ is also used by a lot of researchers instead of $\mathbf{r}(\mathbf{p})$. The main contribution of the AAM is that it assumes a linear relationship between the texture residue $\mathbf{r}(\mathbf{p})$ and the update $\delta\mathbf{p}$ for the model parameter vector:

$$\delta\mathbf{p} = -\mathbf{R} \cdot \mathbf{r}(\mathbf{p}) \quad (2-9)$$

\mathbf{R} is the gradient descent matrix as [8]:

$$\mathbf{R} = \left(\left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}} \right)^T \frac{\partial \mathbf{r}}{\partial \mathbf{p}} \right)^{-1} \left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}} \right)^T \quad (2-10)$$

where $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ is the derivative of $\mathbf{r}(\mathbf{p})$ w.r.t. the model parameter vector \mathbf{p} . It is assumed to

be fixed and could be approximated numerically by systematically adding disturbance to model parameters from their optimal values and taking the average over the whole training set.

$\mathbf{r}(\mathbf{p})$ is still in high dimensional space. X. Hou etc [16] showed that using principal components of $\mathbf{r}(\mathbf{p})$ instead of raw data leads to a more robust alignment result.

$$\mathbf{r}(\mathbf{p}) = \mathbf{A} \cdot \mathbf{r1}(\mathbf{p}) \quad (2-11)$$

Then the new updating rule is:

$$\delta \mathbf{p} = -\mathbf{R} \cdot \mathbf{r}(\mathbf{p}) = \mathbf{R} \cdot \mathbf{A} \cdot \mathbf{r1}(\mathbf{p}) = \mathbf{R1} \cdot \mathbf{r1}(\mathbf{p}) \quad (2-12)$$

In [8], it is mentioned that background could cause some problem in the process of training for the gradient descent matrix. A suggested solution is to use a random background so that

\mathbf{R} is independent of the background patterns in the training set. However, experiment shows that the searching process has visibly worse performance locating the face contour as a result.

In fact, background information can help lock to the right face contour when the unknown face is presented with a background mostly seen in the training set. An efficient solution to the face contour problem will be addressed later.

2.3 Component-based AAM

The AAM is a global appearance-based model and has the ability to model both shape and texture with only a few parameters. After a predicted update for the model parameters is calculated given the current texture residue, almost all existing AAM algorithms adopt the same strategy to guarantee the convergence of the AAM. That is, along the gradient descent

direction $\delta \mathbf{p}$, different step sizes are tested until a smaller matching error is achieved. While the computational cost is proportional with the number of tries, experiment also indicates that it is useless to try for more than 4 different step sizes. In that case, it could either fail to converge, or give a little improvement and fail within the next one or two iterations. Some other action should be taken when the AAM fails to work. The ASM is less affected by global illumination and occlusion compared to the AAM, but the ASM also has its own drawbacks despite its good localization. It could fall into local minimum easily. Building a local model for every landmark point and updating landmark points one by one is time consuming. Based on the fact that local shape depends only on local appearance pattern, a Component-based AAM is proposed in an effort to gain better feature localization while keeping the AAM's merit of good appearance modeling. The basic idea is to group landmark points to components and train the local models independently. To avoid possible confusion, the original AAM is referred as the global AAM. During the modeling process, different components are updated separately as independent sub models. Meanwhile, they are united and confined with a global AAM. In this way, error propagation between local components is reduced and modeling ability is enhanced locally. Three components in the mean shape frame are shown as highlighted areas in Fig. 2.2(a). Landmark points are naturally grouped to balance the added computational cost and the algorithm efficiency. From columns 2.2(b) to 2.2(d), components of the person in Fig. 2.1(a) are shown. The top row shows original facial patches and the bottom row shows warped shapeless textures.

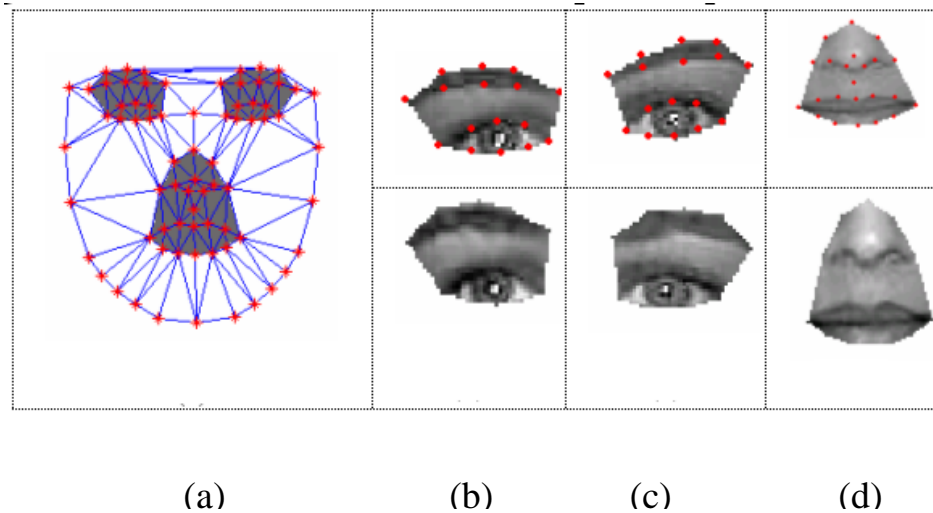


Figure 2.2 (a) Landmark points inside face are divided into 3 groups. (b) Left eyebrow and eye in one group. (c) Right eyebrow and eye. (d) The nose and mouth.

Our component-based AAM is a combination of one global AAM and three sub-models. As part of the global face patch, all components are normalized to the same common coordinate system as that for the global face. This establishes clear correspondence between the global model and the sub-models. Not only all sub-models share the same 2D pose parameters (translations, scaling and rotation) as the whole face, but the component shapes, textures and texture residuals are just fixed entries in their counterparts of the global model. Normalizing the components differently would introduce extra variables and only complicates the problem.

Three sub-models are trained separately for the local components. For the i th component, its combined parameter vector is obtained by projecting appearance pair $\{\mathbf{x}_i, \mathbf{g}_i\}$ to a local eigen-subspace. The local model parameter sets are expressed as $\{\mathbf{p} = \{\Psi, \mathbf{c}_i\}, i = 1, 2, 3\}$. Each component has its own gradient descent matrix \mathbf{R}_i generated by adding disturbance to optimal component model parameters and averaging over the whole training set.

Since the components have the same 2D pose parameters as the global face patch, derivatives of i th texture residue $\mathbf{r}_i(\mathbf{p})$ w.r.t. the pose parameters are simply the corresponding entries in $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$. The sub-models work on the components in the same way the global AAM does on the whole face, except that the optimization criterion for the sub-models is the same global matching error.

During the process of modeling a novel face, the Component-based AAM switches between the global model and the sub-models alternatively. After every iteration of a global AAM, the global appearance parameter set $\mathbf{p} = \{\Psi, \mathbf{c}\}$ is sufficient to reconstruct a series of measurements: the global shape \mathbf{x} , texture \mathbf{g} , shape in the image frame \mathbf{X}_{im} , texture residual $\mathbf{r}(\mathbf{p})$ and global matching error is e_0 . The various steps to model local components are detailed as follows:

For the i th component ($i = 1$ to 3), repeat the following steps:

- 1) **Global to local mapping:** Generate the sub-model shape \mathbf{x}_i , texture \mathbf{g}_i and texture residual $\mathbf{r}_i(\mathbf{p}_i)$ by looking up fixed entries in \mathbf{x} , \mathbf{g} , and $\mathbf{r}(\mathbf{p})$. Project $\{\mathbf{x}_i, \mathbf{g}_i\}$ onto local subspaces. $\mathbf{r}_i'(\mathbf{p}_i)$ are the principal components of $\mathbf{r}_i(\mathbf{p}_i)$ that capture 98% of the variations of the combined feature space.
- 2) **Local AAM prediction:** Apply the local AAM to obtain a new sub-model shape vector \mathbf{x}_i' , texture vector \mathbf{g}_i' and local 2D pose Ψ_i .
- 3) **Local to global mapping:** Use $\{\mathbf{x}_i', \mathbf{g}_i'\}$ to update corresponding entries of the global texture vector \mathbf{g} and local landmark points in the image frame.
- 4) **Decision making:** If the new global parameters result in a smaller matching error, accept the update for the current component.

In summary, three independent sub-models allow components to transform separately for an optimal global matching. In [26], sub-models are constructed to model vertebra. It bears similar idea as ours. However, they basically repeat the same sub-model for a sequence of triplet vertebrae and propagate their results, therefore different from our approach.

2.4 Local Projection Models for Face Boundary Detection

When a test face is presented in a background unseen in the training set, the AAM often fails, especially for face contour points. Since the landmark points on the face contour are usually the strongest local edge points, it stimulates us to develop a method similar to the ASM to complement our component based AAM. First, let's quickly review how the ASM is able to adjust a landmark point to its desired location as illustrated in Fig. 2.3. Each landmark point is pulled toward the strongest edge point along the local normal direction. The right figure shows the edge strength pattern along the dashed line in the left figure. The landmark point A is moved to A' based on the updating rule.

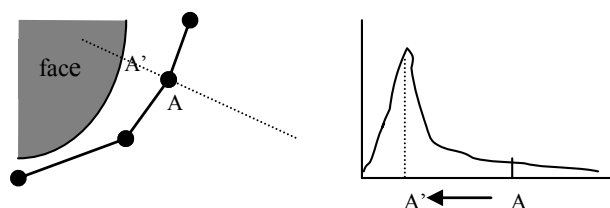


Figure 2.3 Updating rule for the ASM algorithm

Using the edge strength along the profile directly is very sensitive to image noise. Edge information would be more prominent and stable after taking the local average. This could be

implemented by opening a narrow window and accumulating the edge map along the boundary direction to create a projection model for this landmark point. This is shown in Fig. 2.4.

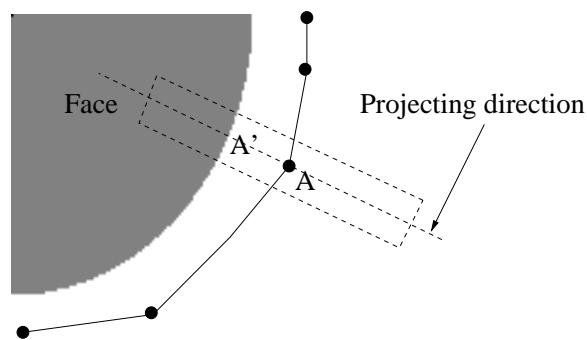


Figure 2.4 Building a local projection model

Building local projection models in the original face image, as the ASM does, would be a time consuming task considering all the contour points that need to be updated within a single iteration. The normal direction at a landmark point depends on this specific point and its 2 neighbor points. Any update to one of the 3 points will lead to a different normal direction. Secondly, the scale of the local projection model should be made proportional to the size of the whole face. These two problems are easily solved in our approach by associating the local projection models with the triangulation result of landmark points. Fig. 2.5 (a) is the mesh of landmark points for the person in Fig. 2.1. Fig. 2.5(b) shows the mean shape.

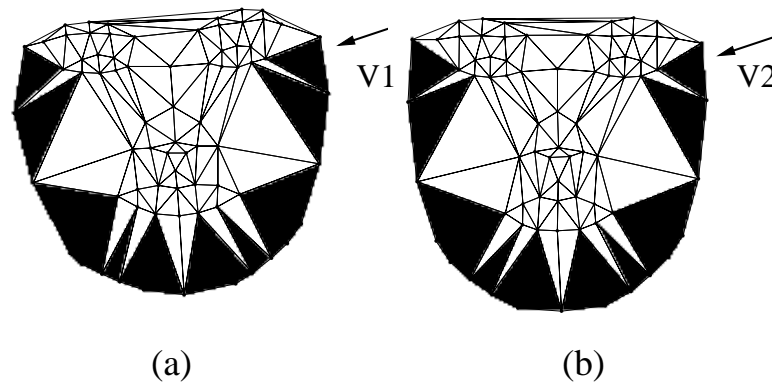


Figure 2.5 Triangulation of face landmark points: a) Original shape of the person in Fig. 2.1. b) Mean shape of training set.

Triangles sitting on the face boundary are filled with black color. Their bottom sides form the face contour. Assume each black triangle is associated with a parallelogram with the bottom side of the triangle being the parallelogram's middle line. Our local projection models are built based on the analysis of the edge map inside these parallelograms.

Instead of working on the edge map in the original face image, the analysis is conducted on the warped image edge map. Suppose a triangle in the original mesh is $\mathbf{V}_1 = \{\mathbf{v}_{11}, \mathbf{v}_{12}, \mathbf{v}_{13}\}$, where $\mathbf{v}_{11} = (x_{11}, y_{11}, 1)'$ is one of the vertices under homogeneous coordinate system. The corresponding transformed triangle in the mean shape is $\mathbf{V}_2 = \{\mathbf{v}_{21}, \mathbf{v}_{22}, \mathbf{v}_{23}\}$. An isosceles triangle is introduced as a standard triangle $\mathbf{V}_0 = \{\mathbf{v}_{01}, \mathbf{v}_{02}, \mathbf{v}_{03}\}$. As long as the bottom side of the triangle is transformed to the horizontal or vertical position, the projection along the face contour direction in the face image is now simplified to summation along the x (or y axis) after 2 transforms. Fig. 2.6 illustrates how a triangle-parallelogram in the face image is warped to the mean shape frame and subsequently to the standard triangle-rectangle pair.

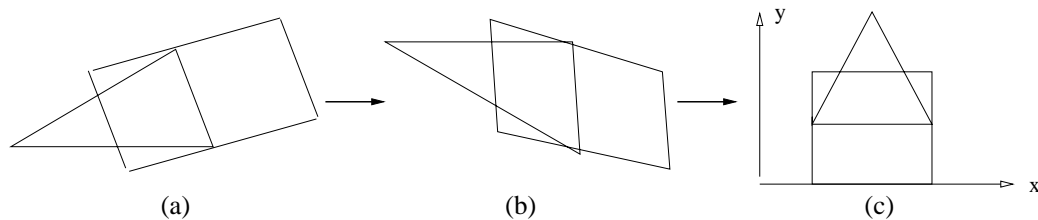


Figure 2.6 A parallelogram associated with a triangle on the face boundary. a) Original image frame. b) Mean shape frame. c) Standard pair.

$$\mathbf{V}_1 = \mathbf{A} \cdot \mathbf{V}_2 \quad (2-13)$$

$$\mathbf{V}_2 = \mathbf{B} \cdot \mathbf{V}_0 \quad (2-14)$$

$$\mathbf{V}_1 = \mathbf{A} \cdot \mathbf{B} \cdot \mathbf{V}_0 \quad (2-15)$$

The piece-wise affine transform parameters are available in the AAM step (to generate a shapeless texture vector from the current face patch). The transformations between \mathbf{V}_0 and all the triangles in the mean shape could also be computed in advance. Clearly, with the help of mean shape and a standard triangle, the local projection models can lock the face contour points to the locally strongest edge points. It is much faster and easier compared to the ASM. The regions of interest for our local projection models are defined according to current face landmark points. Therefore there is no scaling problem at all.

2.5 Component-based Search

The AAM search on an unknown face image is an iterative procedure to find the best model parameters. The update for the model parameters is driven by a linear prediction model. When the texture residue is beyond the linear approximation range, a first order gradient descent matrix has less or no prediction power, especially when the matrix is a fixed matrix throughout the whole search. A good initial model parameter set is necessary for the AAM to

converge successfully. A common initialization starts with the average shape, texture and 2D pose parameters. As different initialization strategies could significantly affect the face alignment performance, it is worthwhile to pay a little attention to the initialization method. A direct exhaustive searching strategy is used. After sampling the face subspace, 900 prototype models are generated which vary in pose, shape and appearance. Given an unknown face image, one of the prototype models is quickly chosen. The direct search is very fast as most of the computation is irrelevant with any specific image, therefore needs to be done only once. The detailed initialization method is given in Appendix B.

Like most other AAM algorithms, our component-based AAM works in a pyramidal way. To roughly locate the face and facial features, in the first stage it is enough to work on the face image at low resolution. This helps prevent the search from being trapped in a local minimum. The alignment result of lower resolution is then passed to the original image as initial parameters. In the iterative realization, the global AAM is run first and when it fails to converge, local AAMs is launched, followed by the local projection models to lock current model boundary points to the nearby strongest edge points. The search will stop in any of the 3 cases: 1). No more improvement could be made from last iteration; 2). Maximum number of iterations has reached; 3). Matching error is below a pre-defined threshold ε . Fig. 2.7 is a flowchart illustrating all steps.

The search in the original image is more complex and time consuming compared to the search involved in a smaller image. Practically human interference should be added when the search in the first stage fails. All experiments in the next section were however carried out without any human interference.

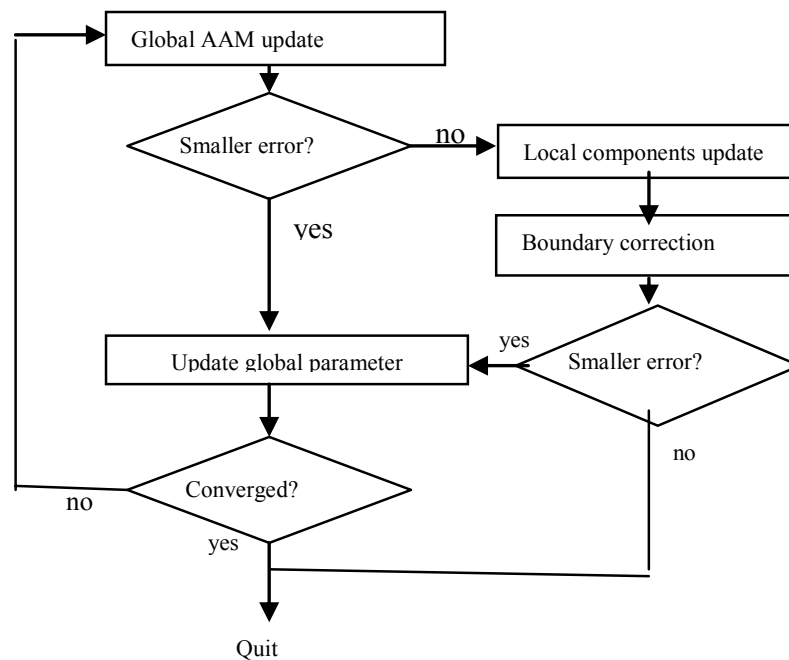


Figure 2.7 Flowchart of steps in an iterative AAM search

2.6 Experiments

Our face database includes 138 nearly frontal images from various face databases [27] [28] [29]. [The set consists of neutral faces and](#) all images were roughly resized and cropped to 256 by 256. Various lighting conditions and image background patterns are challenging to the proposed algorithm. Nevertheless, a versatile face database is the best way to test the robustness. 80 images are sequentially picked to train face shape subspace and rest of the images constitute the test set. Some samples from our database are shown in Fig. 2.8.

The proposed algorithm is also tested on the Japanese Female Facial Expression database (JAFFE) [30], which contains 213 images of 7 facial expressions of 10 female models.

[Though our algorithm is only designed to deal with neutral faces, its performance on the](#)

JAFFE turns out to be quite good. The only pre-processing that is conducted is to scale the original 200 by 200 images to standard size 256 by 256.

Though the texture error is used as the evaluation function, it does not reflect model fitting quality strictly. For all images in our database and JAFFE database, all landmark points are manually labeled and a distance map is created for each image. The model fitting quality is then measured by the average point-to-edge distance. Within the same framework, three different algorithms are tested and compared: the AAM search; the AAM with Component analysis (AAM_CA); the AAM with component analysis and local projection models (AAM_CA_LPM).



Figure 2.8 Sample images in our face database.

2.6.1 Component-based Search

Fig. 2.9 compares the AAM and the AAM_CA model fitting results. Apparently face components inside the face area have better feature localization with the enhanced

component-based search.

As expected, the converged global AAM couldn't achieve optimal local alignment results.

Better localization of facial feature points could be seen on the bottom row. Note there is no boundary correction involved here. Table 2.1 shows the average point-to-edge errors for algorithms with and w/o component analysis. Only face component points are considered.

Table 2.1 Average point to edge error (contour points excluded)

Algorithms	Training set	Test set	JAFFE
AAM	2.0661	3.5513	3.1696
AAM_CA	1.8988	3.2429	2.9377

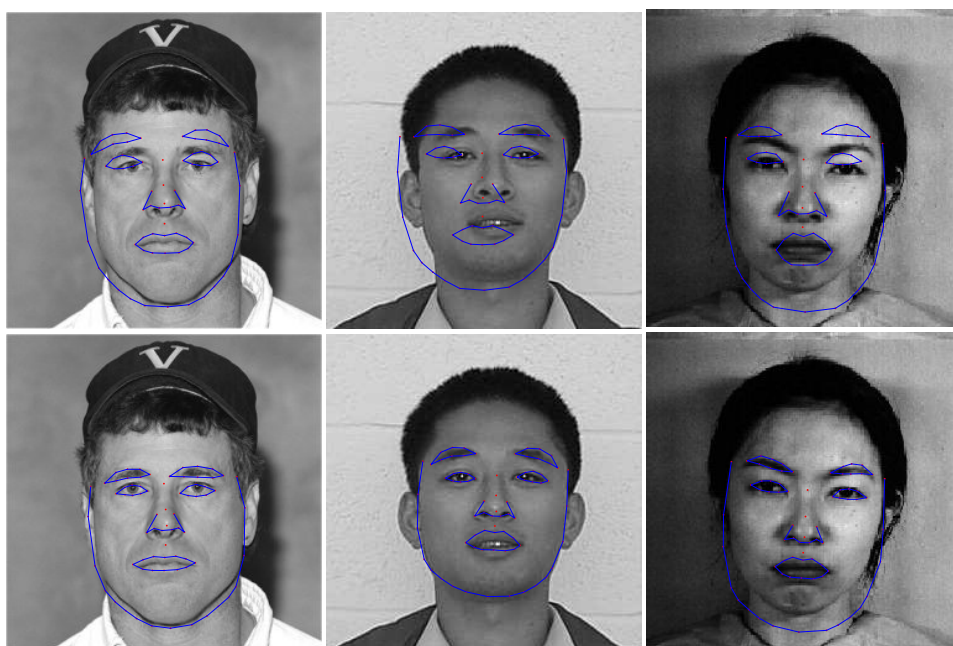


Figure 2.9 AAM (top row) versus AAM_CA (bottom row). (a) Training set. (b) Test set. (c) JAFFE

2.6.2 Face Contour Detection with Local Projection Models

The AAM_CA and the AAM_CA_LPM fitting results are compared to show how the integration of local projection models can help solve the boundary problem. Fig. 2.10 shows some examples.

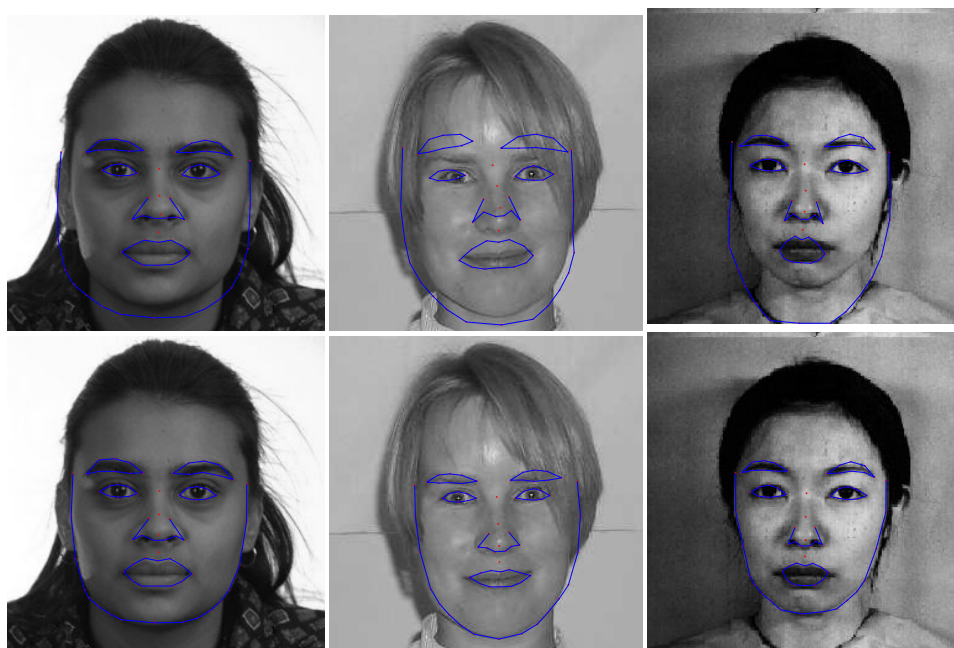


Figure 2.10 AAM_CA (top row) versus AAM_CA_LPM (bottom row). (a) Training set. (b) Test set. (c) JAFFE

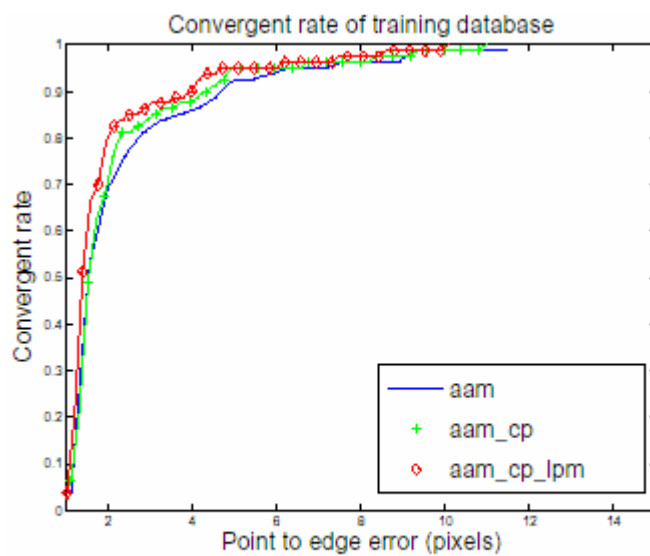
Table 2.2 shows the average point-to-edge errors for algorithms with and without face contour correction. The eye-catching error of 7.4741 indicates an almost total failure of face contour detection. Clearly the AAM trained on one database has problem detecting the right face contour on a different database. This error is efficiently lowered to 4.1356 with the proposed algorithm.

Table 2.2 Average point to edge error(contour points only)

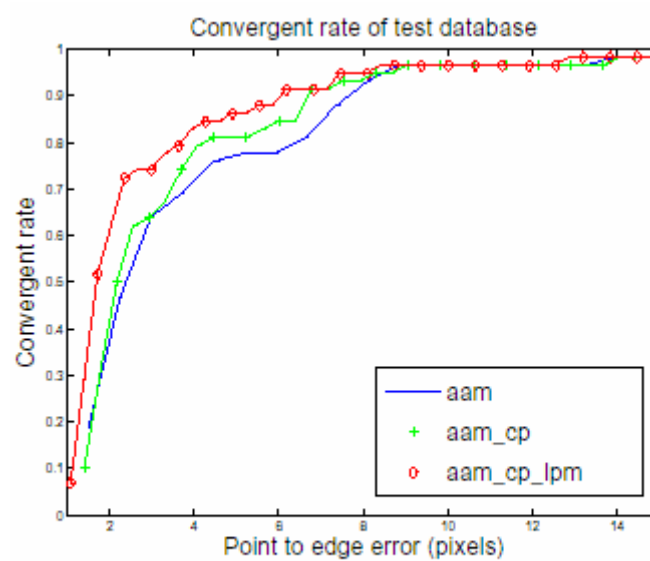
Algorithms	Training set	Test set	JAFFE
AAM_CA	3.5298	4.7153	7.4741
AAM_CA_LPM	3.2909	3.8430	4.1356

It is interesting to see in Fig. 2.9(b), boundary points are correctly aligned due to component analysis. Also Fig. 2.10(b) has correct component points. Clearly the integration of local AAM analysis and local projection models makes our fitting algorithm more accurate and robust.

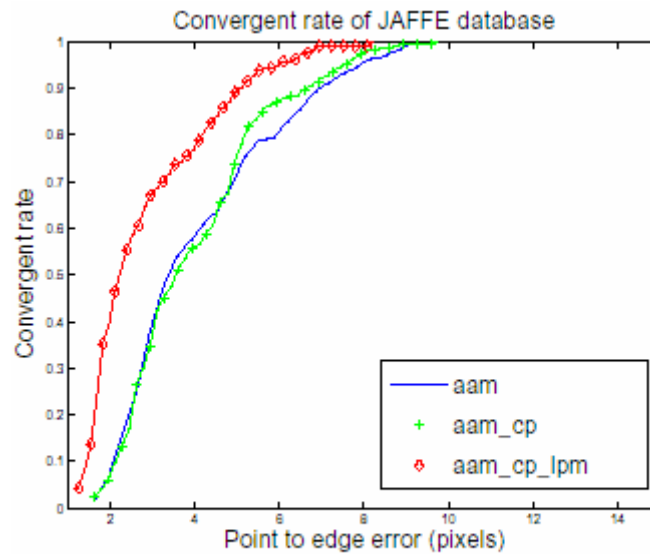
At last, the AAM, the AAM_CA and the AAM_CA_LPM are compared w.r.t. their convergent rate curves as shown in Fig.2.11. A good approximation of an error PDF function can be obtained with the histogram of point errors for all images in a database. The cumulative function also represents the convergent rate. i.e., given a point to edge error ϵ in x-axis, y-axis gives the percentage of images with errors smaller or equal to ϵ . Clearly the AAM_CA_LPM has the best performance and the improvement is especially prominent for the JAFFE database.



(a)



(b)



(c)

Figure 2.11 Curves of convergent rate versa error threshold. (a) Training set. (b) Test set. (c) JAFFE database.

2.7 Conclusions

In this chapter, a component-based AAM algorithm is proposed to deal with the lack of feature localization problem in the original AAM. Model points are naturally grouped and sub-models are combined with the global AAM model. In this way, the simplicity and efficiency of the AAM model is preserved, as well as the ASM's good localization ability. The background problem is solved by using local projection models which attract boundary model points toward the strongest edge points locally. All component sub-models and local projection models are tightly combined and smoothly interact with the global AAM model by sharing their intermediate results. As a result, our algorithm is efficient and shows steady performance for images from different sources.

CHAPTER 3 : 2D FACE MODELING WITH A HYBRID CONSTRAINED OPTIMIZATION ALGORITHM

In Chapter 2, a component-based AAM is addressed for face modeling. The improved performance is due to the inherent localization of the proposed model. However, *it only* refines the alignment result when the face is already approximately located. When it fails to locate the face even roughly, it has a high probability to fail. For this reason, in this chapter, a new face alignment algorithm will be presented from a different perspective. A constrained hybrid optimization algorithm incorporates several shape constraints into a gradient-descent procedure using a novel unbiased cost function. Shape constraints are heuristically derived from face images where the face shape can be directly estimated based on "motion" analysis. To better locate face contour points regardless of the background, local projection models are used. Experiments show that our algorithm benefits significantly from these shape constraints and achieves a much higher convergent rate compared to the inverse compositional optimization algorithm. Our algorithm is tested on different face databases, and its robustness is fully demonstrated in the presence of varying illumination, background, and facial expressions.

3.1 Literature

A widely adopted strategy in face analysis and recognition is analysis by synthesis, i.e., describing and analyzing a human face through modeling. In recent years, many flexible model-based algorithms have been proposed based on the analysis by synthesis approach and

have been shown to be fruitful in a wide area of applications ranging from face coding, to face reconstruction, to facial expression recognition etc. Two such models that have received a great deal of attention are: the Multidimensional Morphable Model (MMM) [31] and the AAM. Both are generative parametric models that have been customized to model a class of objects.

The AAM, the MMM and most of their variants, follow several basic rules. Raw face shape and texture are extracted from the face image and are stored (in a vector form) as two distinct measurements. Face parameters are formed by treating face shape (or texture) as a linear combination of a set of exemplar basis shapes (or textures). Fitting a face model to a face image is an optimization process that minimizes a cost function. Modeling quality is evaluated using the cost function, which usually measures the minimum mean square error (MSE) between a synthesized model face and input face.

Face modeling amounts to finding the global minimum of the cost function, and is usually attained with standard gradient descent algorithms. It is well known that for the solution to be a desired global minimum, the cost function has to be convex, which is hard to meet in reality. Besides, heavy computation is inevitable as gradient and Hessian information of the cost function need to be updated iteratively. The MMM originally adopts a stochastic gradient descent algorithm, which is comparatively fast and can to some extent avoid the trapping in local minima. The AAM takes a totally different approach, which has been explained in the last chapter. It assumes a fixed linear relationship between the texture error and the necessary update in the parameter space. This relationship is learned from a training set. The AAM is very fast, but it also has obvious drawbacks. First, the assumption of a fixed linear

relationship is incorrect [17]. Secondly, the face image background is encoded, which may result in degraded performance when modeling a novel face with an unseen background. Matthews etc. [17] proposed an inverse compositional AAM that is as efficient as the AAM, yet is more theoretically founded. Its main advantage over standard gradient descent algorithms is its computational efficiency due to constant gradient and Hessian information. The inverse compositional AAM is reported to have similar convergence rate as the original AAM [32], and is therefore used as a test bed to compare with our algorithm.

Global algorithms attempt to find the bottom of the deepest valley for the cost function over a region of parameter space. Simulated annealing algorithms and genetic algorithms are two typical global optimization methods. Global algorithms have broader view of the cost function's terrain and seek for a possibly better solution.

Treating the fitting problem as a general function minimization problem and seeking an analytical solution can hardly achieve both a decent convergent rate and algorithm efficiency. Fortunately there are alternative methods from a different perspective. The Active Shape Model (ASM) [12] is such a good example. It fully utilizes the heuristic information from face image and moves facial landmark points in a way so that extracted local terrain features conform to typical local distributions. The Active Morphable Model (AMM) [33] directly estimates face shape using a standard optical flow algorithm. Its model fitting algorithm is robust with a large region of convergence. However its iterative hierarchical optical flow estimation inside each iterative step makes it a less efficient model.

In this chapter, a constrained hybrid optimization algorithm is proposed to efficiently fit a morphable model to frontal face images. This algorithm takes advantage of abundant heuristic

information in face image while preserving the gradient descent power of an analytical cost function. Based on our observation of one weakness of the widely adopted cost function, a novel unbiased error evaluation function is proposed. Two different algorithms are adopted to estimate face shape directly from face image. One is the block 'motion' estimation algorithm that adjusts crucial landmark points for better local fitting. Another one is the local projection model that has been discussed in the previous chapter. The directly estimated shape is incorporated as shape constraint within the framework of the standard inverse compositional fitting algorithm. At the initial phase of our model fitting process, the directly estimated shape dominates so that the search has better chance to move toward a global solution. As the model fitting error becomes smaller, gradient descent direction of the cost function takes control. Experiment indicates that a much higher convergence rate is achieved. The hierarchical motion estimation algorithm helps lock to the neighborhood of true minimum. In this sense, it is a global optimization scheme. Yet, it is much more efficient than general global optimization algorithms like simulated annealing algorithms and genetic algorithms. Average shape from training set is good enough to serve as the initial model estimate. Another merit of our algorithm is that it is truly background independent compared to various AAM or MMM algorithms.

The chapter is organized as follows: Section 3.2 briefly introduces the normalized inverse compositional optimization method; Section 3.3 starts with a novel definition of error evaluation function, followed by the block motion estimation in Section 3.4. Section 3.5 shows how the estimated shape is incorporated to the inverse compositional optimization scheme; Section 3.6 gives our experimental results and analysis, followed by the conclusion

part.

3.2 Normalized Inverse Compositional AAM Algorithm

The Normalized Inverse Compositional AAM (NIC_AAM) is a very important variant of the original AAM. Notations here will be slightly different from those in the previous chapter.

The NIC_AAM is proposed by a group of people in CMU, who have a long research history in the area of image alignment. They used a set of notations that are more consistent with their own convention in their previous study in the inverse compositional image alignment algorithm. This chapter will follow their notations as it is easier to understand.

Let \mathbf{S}_{img} be the raw image shape which is a sequence of coordinates of the predefined landmark points. Without confusion, the same notation \mathbf{S}_{img} is used to refer to the face shape, the triangular mesh of the landmark points and the set of all enclosed pixels. Let \mathbf{S}_0 be the base shape (or mesh), which usually takes the form of the average shape in the training set. Let \mathbf{p} be the shape parameters in the shape subspace, $\boldsymbol{\lambda}$ be the texture parameters in the texture subspace and \mathbf{q} be the similarity parameters needed for coordinate normalization purpose. $\{\mathbf{S}_k\}$ and $\{\mathbf{T}_k\}$ are respectively the set of basis for the shape subspace and the texture subspace. Unlike the AAM, the NIC_AAM doesn't mix the shape and texture parameters to create combined parameters to de-correlate shape and texture. $\{\mathbf{q}, \mathbf{p}, \boldsymbol{\lambda}\}$ forms the complete face model parameter set. A model face can be rendered by warping a synthesized texture \mathbf{T} from base mesh \mathbf{S}_0 to the reconstructed face mesh \mathbf{S}_{img} .

Aligning a face in a given image is equivalent to finding the optimal parameters so that the difference between the model face and the given face image is minimized. Any pixel \mathbf{u} inside

the base face \mathbf{S}_0 is mapped to a new position $\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q})$ in the image according to the shape \mathbf{p} and the similarity parameters \mathbf{q} . The cost function usually takes the form of the sum of the squares of the texture residue in accordance with:

$$G(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = \sum_{\mathbf{u} \in S_0} \left[\mathbf{T}_0(\mathbf{u}) + \sum_{k=1}^{m_0} \lambda_k \cdot \mathbf{T}_k(\mathbf{u}) - \mathbf{I}(\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q})) \right]^2 \quad (3-1)$$

In the original AAM, the difference image is directly used to update the current model parameters using a linear regression model. Such an updating rule is efficient and empirically driven, but it doesn't fall into any traditional optimization category.

Given the function $G(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda})$, a variety of gradient descent optimization algorithms could be applied. Generally, given a current estimate of model parameters, the function $G(\mathbf{p} + \Delta \mathbf{p}, \mathbf{q} + \Delta \mathbf{q}, \boldsymbol{\lambda} + \Delta \boldsymbol{\lambda})$ can be approximated to be a quadratic function of the incremental updates $\Delta \mathbf{p}$, $\Delta \mathbf{q}$ and $\Delta \boldsymbol{\lambda}$, using a Taylor series expansion, and closed form solutions and updates of the current estimate in the parameter space are found. This operation is repeated till a minimum is reached. Unfortunately, gradient descent algorithms are slow due to heavy computation. The inverse compositional algorithm is a principled gradient descent algorithm that performs model fitting very efficiently. It is derived from the canonical Lucas-Kanade image alignment algorithm. Assume that there is no texture variation from the base texture image \mathbf{T}_0 to an input image \mathbf{I} , and the cost function is:

$$E(\mathbf{p}, \mathbf{q}) = \sum_{\mathbf{u} \in S_0} [\mathbf{T}_0(\mathbf{u}) - \mathbf{I}(\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q}))]^2 \quad (3-2)$$

Basically in the iteration, the base texture image \mathbf{T}_0 is warped according to the underlying parameters $\Delta \mathbf{p}$ and $\Delta \mathbf{q}$ so that the warped base image resembles the input image at the current warp $w(\mathbf{u}; \mathbf{p}, \mathbf{q})$. Mathematically, it equals to minimize:

$$E(\Delta\mathbf{p}, \Delta\mathbf{q}) = \sum_{\mathbf{u} \in S_0} [\mathbf{T}_0(\mathbf{W}(\mathbf{u}; \Delta\mathbf{p}, \Delta\mathbf{q})) - \mathbf{I}(\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q}))]^2 \quad (3-3)$$

The real warp between the base texture image and the input image is then updated as a composition of the current warp and the inverse of the incremental warp parameters resulting from minimizing (3). The updating rule in (4) also explains how the inverse compositional AAM got its name.

$$\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q}) \leftarrow \mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q}) \circ \mathbf{W}(\mathbf{u}; \Delta\mathbf{p}, \Delta\mathbf{q})^{-1} \quad (3-4)$$

The cost function $E(\mathbf{p}, \mathbf{q})$ doesn't consider the texture variation between the base image and the input image. In the normalized inverse compositional algorithm, the texture parameters are iteratively updated given the pose parameters \mathbf{q} and the shape parameters \mathbf{p} . The closed form solution is:

$$\lambda_i = \sum_{\mathbf{u} \in S_0} \mathbf{T}_k(\mathbf{u}) \cdot [\mathbf{I}(\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q})) - \mathbf{T}_0(\mathbf{u})] \quad (3-5)$$

The difference image is accordingly adjusted for the inverse compositional algorithm.

The merit of the inverse compositional AAM is its efficiency. In the Gaussian-Newton gradient descent algorithm, approximating $\mathbf{T}_0(\mathbf{W}(\mathbf{u}; \Delta\mathbf{p}, \Delta\mathbf{q}))$ in (3-3) with its first order Taylor expansion requires the evaluation of $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at the warp $\mathbf{W}(\mathbf{u}; \mathbf{0}, \mathbf{0}) = \mathbf{u}$, the identity transform.

This leads to constant gradient descent images and Hessian matrix. Therefore, the computation time is cut greatly. It's not been demonstrated or proved in [17] that inverse compositional AAM might be superior to other gradient descent algorithms in terms of its convergent rate. It is just a special form of general gradient descent algorithm and fast enough for real-time applications.

When a global minimum is desired, gradient descent methods are suitable only when the objective function is convex. However, the sum of squared texture error is far from convex

function. Most likely gradient descent methods will end up in an undesired local minimum. It is of great interest to design an efficient and global optimization method for fitting this kind of morphable models.

3.3 Unbiased Error Evaluation Function

Intuitively, our goal is to seek for the model parameters so that a synthesized model face best resembles the face in the unknown image. Naturally the fitting quality should be evaluated as summed squares of the texture error on the image frame. For most appearance-based models, however, the fitting error is computed on the base shape frame as sum of squared difference of synthesized texture and shapeless texture warped from input unknown image. Though measuring the fitting error on a standard base frame is straightforward and efficient, it can't reflect the real model fitting quality. This is caused by the piecewise affine transform during the image warping. As a result, the underlying optimization process is affected. Therefore, we propose a revised error function so that the error computed on the standard base frame could impartially reflect the fitting quality on the image frame.

On the test image frame, assume the n^{th} triangle \mathbf{L}_n has area $\tau_n(\mathbf{p}; \mathbf{q})$. All pixels inside this triangle contribute to the sum of squared error as:

$$\varepsilon_n(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = \sum_{\mathbf{u} \in \mathbf{L}_n} \left[\mathbf{T}_0(\mathbf{u}) + \sum_{k=1}^{m_0} \lambda_k \cdot \mathbf{T}_k(\mathbf{u}) - \mathbf{I}(\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q})) \right]^2 \quad (3-6)$$

The total error on the image frame is then the sum over all n_t triangles:

$$G_1(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = \sum_{n=1}^{n_t} \varepsilon_n(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) \quad (3-7)$$

Accordingly, denote the corresponding n^{th} triangle \mathbf{L}_n' on the standard base frame has area τ_n' .

It is easy to see that under affine transform mapping, the pixels inside this triangle have the same mean squared error as its counterpart triangle on the image frame. Apparently, the canonical error function on standard frame shown in (3-1) can be expressed as:

$$G(\mathbf{p}, \mathbf{q}, \lambda) = \sum_{n=1}^{n_i} \frac{\tau_n'}{\tau_n(\mathbf{p}, \mathbf{q})} \varepsilon_n(\mathbf{p}, \mathbf{q}, \lambda) \quad (3-8)$$

As an effort to minimize (3-8), general optimization algorithms might tend to maximize $\tau_n(\mathbf{p}; \mathbf{q})$ for large $\varepsilon_n(\mathbf{p}; \mathbf{q})$, therefore are more likely to converge to a local minimum. Such an effect is not desirable since the fitting error on the image frame is still large, indicating a bad model fitting. Fig. 3.1 shows how the piece-wise affine transform might affect the overall fitting error.

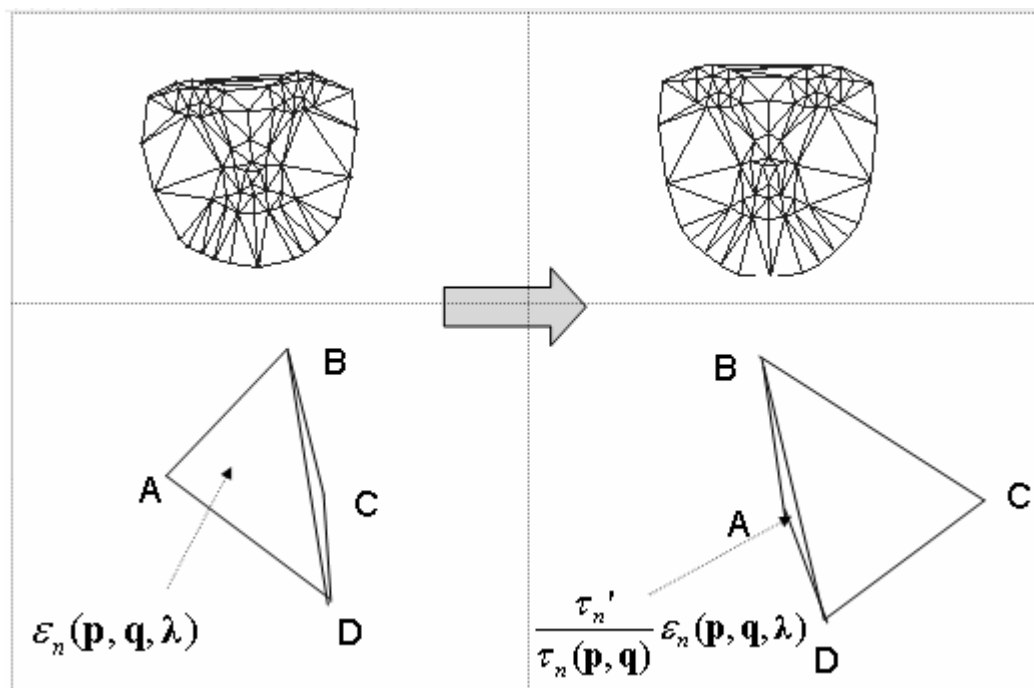


Figure 3.1 Example of a piece-wise affine transform from the image frame (left) to the base frame (right)

This problem is inherent in all model fitting algorithms based on the shapeless texture error measurement. On the other hand, computing a cost function on the standard frame is computationally efficient and better controlled. It is not difficult to rewrite $G_1(\mathbf{p}, \mathbf{q}, \boldsymbol{\alpha})$ on the standard frame as:

$$G_1(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = \sum_{\mathbf{u} \in S_0} \frac{\tau_n(\mathbf{p}, \mathbf{q})}{\tau_n'} \cdot \sum_{\mathbf{u} \in L_n'} \left[\mathbf{T}_0(\mathbf{u}) + \sum_{k=1}^{m_0} \lambda_k \mathbf{T}_k(\mathbf{u}) - \mathbf{I}(\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q})) \right]^2 \quad (3-9)$$

Because the total area of the face mesh on the image frame depends on the parameters \mathbf{p} and \mathbf{q} , it is more reasonable to use mean squared error (MSE) to evaluate the model fitting quality.

The MSE on the image frame can then be computed on the standard model frame as:

$$G_{mse}(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = \sum_{\mathbf{u} \in S_0} \rho_n(\mathbf{p}, \mathbf{q}) \cdot \left\{ \mathbf{T}_0(\mathbf{u}) + \sum_{k=1}^{m_0} \lambda_k \mathbf{T}_k(\mathbf{u}) - \mathbf{I}(\mathbf{W}(\mathbf{u}; \mathbf{p}, \mathbf{q})) \right\}^2 \quad (3-10)$$

G_{mse} is our new cost function. It is nothing but a weighted error function. The weight function varies for different triangles. In [34], it is explained how to minimize a weighted error function under the framework of inverse compositional algorithm. In short, Gradient and Hessian matrix of the new cost function are now respectively weighted sum of gradient and Hessian matrices for all triangles, which can be pre-computed before the iterative optimization

3.4 Direct Shape Estimate from Motion Estimation

Motion estimation is one of the many applications of morphable models. Our focus here is however the opposite. We would like to examine how a general motion estimation algorithm could help the model fitting process of our parameterized face model.

Motion estimation, literally, means estimating the motion of an object or camera from

consecutive video images. In a standard morphable model fitting process, a parameterized model is fitted to an unknown image in an analysis-by-synthesis fashion. A cost function is minimized so that the synthesized model image is aligned with the input image. In other words, the fitting process tries to estimate the arbitrary “motion” between the synthesized image and the unknown image. Since facial landmark points are carefully defined in salient facial areas with a rich texture. It is of particular interest to estimate the motions of blocks centered at all landmark points.

Block motion model is one of the fundamental motion estimation methods. For a block centered at current landmark point \mathbf{u}_0 in the synthesized image \mathbf{I}_s , assume that its counterpart best-matching block in the unknown image \mathbf{I}_0 is displaced by a flow vector \mathbf{v} .

Vector \mathbf{v} can then be obtained by minimizing the sum of squared error as follows:

$$E_{\beta}(\mathbf{u}) = \sum_{\mathbf{u} \in \beta_s} (\mathbf{I}_s(\mathbf{u}) - \mathbf{I}_0(\mathbf{u} + \mathbf{v}))^2 = \sum_{\mathbf{u} \in \beta_s} (\Delta \mathbf{I} - \nabla \mathbf{I}_0 \cdot \mathbf{v})^2 \quad (3-11)$$

where $\Delta \mathbf{I} = \mathbf{I}_s(\mathbf{u}) - \mathbf{I}_0(\mathbf{u})$ is the difference block image. $\nabla \mathbf{I}_0$ is the gradient of the unknown image. The solution to (3-11) is:

$$\left[\sum (\nabla \mathbf{I}_0)^T (\nabla \mathbf{I}_0) \right] \cdot \mathbf{v} = \sum (\nabla \mathbf{I}_0)^T \Delta \mathbf{I} \quad (3-12)$$

The incremental flow vector will be accepted if it leads to a smaller block fitting error. Figure 3.2 shows a synthesized model image and how it looks like when overlapped to the unknown image.

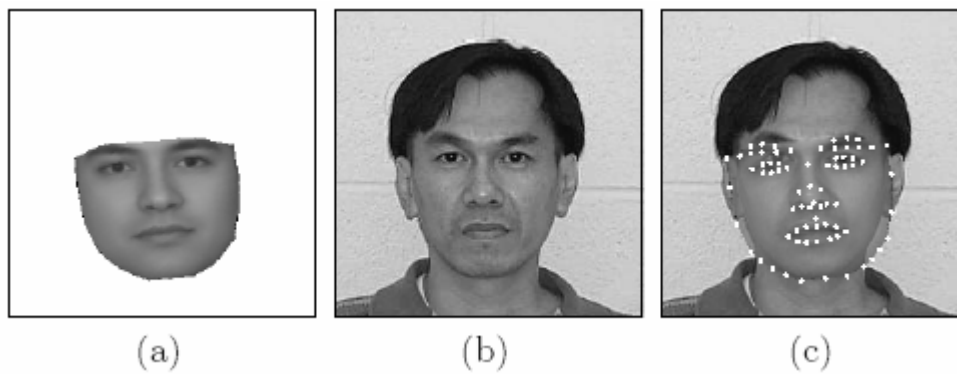


Figure 3.2 From left to right: a) Synthesized image. b) Input image. c) Synthesized face overlapped on the original face with current landmark points.

This simple block motion estimation algorithm is very efficient and fast. In (3-11), the block motion error is computed on the image frame and only the difference image and the gradient of the unknown image are required to estimate the flow vector. Applying the motion estimation with various block sizes hierarchically on a down-sampled image generates more robust estimation. In a standard gradient descent optimization procedure, the motion estimation procedure only adds little extra computation.

Another merit of conducting motion estimation on the image frame is the background independence. Synthesized face is confined to the convex hull of landmark points as shown in Fig.3.2. This convex hull serves as a face mask. During the block motion estimation, blocks are examined against this face mask so that only inner facial pixels participate in the motion estimation procedure. In this way, background independence is achieved. On the other hand, by subjecting the whole unknown image to motion estimation, heuristic information is fully explored.

3.5 Constrained Gradient Descent Optimization

Assume from the “motion estimation” in 3.4 and local projection models (refer to Chapter 2.4 for details), the direct face shape estimate is \mathbf{S}^* . It is desirable to minimize the distance between parameterized face shape $\mathbf{W}(\mathbf{S}_0; \mathbf{p}, \mathbf{q})$ and \mathbf{S}^* . Mathematically, we construct a new cost function $F(\mathbf{p}, \mathbf{q})$ as sum of squared distance:

$$F(\mathbf{p}, \mathbf{q}) = \|\mathbf{W}(\mathbf{S}_0; \mathbf{p}, \mathbf{q}) - \mathbf{S}^*\|^2 \quad (3-13)$$

In section 3.1, our cost function is defined as weighted mean squared texture error. Since $F(\mathbf{p}, \mathbf{q})$ is squared shape error, it provides complementary knowledge. To effectively benefit from both functions, we construct a hybrid function as combination of $F(\mathbf{p}, \mathbf{q})$ and $G_{mse}(\mathbf{p}, \mathbf{q}; \boldsymbol{\alpha})$:

$$Z(\mathbf{p}, \mathbf{q}) = G_{mse}(\mathbf{p}, \mathbf{q}; \boldsymbol{\alpha}) + K \cdot F(\mathbf{p}, \mathbf{q}) \quad (3-14)$$

The function $Z(\Delta\mathbf{p}, \Delta\mathbf{q})$ is minimized to generate the incremental update $\Delta\mathbf{p}$, $\Delta\mathbf{q}$. Optimization based purely on texture error function $G_{mse}(\mathbf{p}, \mathbf{q}; \boldsymbol{\alpha})$ has a small region of convergence by nature. In the iterative realization, weight K starts with a large initial value, so that shape error function $F(\mathbf{p}, \mathbf{q})$ plays a major role at the initial stage. As search continues, K decreases till texture error function $G_{mse}(\mathbf{p}, \mathbf{q}; \boldsymbol{\alpha})$ totally dominates the optimization. The gradient-descent generative function $Z(\mathbf{p}, \mathbf{q})$ is discriminated from the fitting error evaluation function $G_{mse}(\mathbf{p}, \mathbf{q}; \boldsymbol{\alpha})$. New parameters are accepted when they imply a smaller fitting error.

The inverse compositional optimization algorithm is chosen to minimize function $Z(\mathbf{p}, \mathbf{q})$ due to its efficiency and effectiveness. The general framework about how to minimize a constrained function like (3-14) is introduced in [34]. So the problem left here is how to minimize (3-14) when the shape transform is a piece-wise affine transform for morphable face models.

A unique feature of the inverse compositional algorithm is that model parameters are updated indirectly according to (3-4). The real incremental updates $(\Delta\mathbf{p}', \Delta\mathbf{q}')$ to current model parameters (\mathbf{p}, \mathbf{q}) are $(\Delta\mathbf{p}', \Delta\mathbf{q}') = \mathbf{J} \cdot (\Delta\mathbf{p}, \Delta\mathbf{q})$, where \mathbf{J} is the Jacobian matrix as follows:

$$Z(\Delta\mathbf{p}, \Delta\mathbf{q}) = G_{mse}(\Delta\mathbf{p}, \Delta\mathbf{q}) + K \cdot F((\mathbf{p}, \mathbf{q}) + \mathbf{J} \cdot (\Delta\mathbf{p}, \Delta\mathbf{q})) \quad (3-15)$$

Let \mathbf{U}_s be a matrix whose columns are orthogonal prototype shapes, $\mathbf{U}_s = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_{m0}]$. Let \mathbf{U}_g be a matrix of orthogonal global shape basis so that $\mathbf{S} = \mathbf{U}_g \cdot \mathbf{q}$ implies the equivalent global affine transform (refer to [17] for details). Assume that the j^{th} triangle \mathbf{V}_0 in the base face mesh \mathbf{S}_0 is mapped to its counterpart triangle \mathbf{V}_1 in the image frame by a general affine transform as $\mathbf{V}_1 = \mathbf{R}_j \cdot \mathbf{V}_0 + \mathbf{b}_j$. Then it is easy to find the Jacobian matrix \mathbf{J} to be

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{p}}{\partial \Delta \mathbf{p}} & 0 \\ 0 & \frac{\partial \mathbf{q}}{\partial \Delta \mathbf{q}} \end{bmatrix} \quad (3-16)$$

where $\frac{\partial \mathbf{p}}{\partial \Delta \mathbf{p}}$ is the derivative of the shape parameters \mathbf{p} w.r.t. $\Delta \mathbf{p}$ and $\frac{\partial \mathbf{q}}{\partial \Delta \mathbf{q}}$ is the derivative of \mathbf{q} w.r.t. $\Delta \mathbf{q}$. In detail,

$$\begin{cases} \frac{\partial \mathbf{p}}{\partial \Delta \mathbf{p}} = -\mathbf{U}_s' \cdot \mathbf{A}_p \cdot \mathbf{U}_s \\ \frac{\partial \mathbf{q}}{\partial \Delta \mathbf{q}} = -\mathbf{U}_g' \cdot \mathbf{A}_q \cdot \mathbf{U}_g \end{cases} \quad (3-17)$$

where \mathbf{A}_p is a matrix of all zeros except two by two sub-matrices along the main diagonal direction:

$$\overline{\mathbf{R}}^i = \begin{bmatrix} \mathbf{A}_p(2i-1, 2i-1) & \mathbf{A}_p(2i-1, 2i) \\ \mathbf{A}_p(2i, 2i-1) & \mathbf{A}_p(2i, 2i) \end{bmatrix} \quad \text{for } i = 1, 2, \dots, n_v \quad (3-18)$$

where n_v is the number of mesh vertices. $\overline{\mathbf{R}}^i$ is the affine transform matrix associated with the i^{th} vertex of the face mesh (superscript is used to discriminate it from \mathbf{R}_j , which denotes

the transformation matrix for j^{th} triangle). $\overline{\mathbf{R}}^i$ is generated by averaging the affine transforms of all triangles associated with this specific vertex. \mathbf{A}_q is defined in the same way, but it is much simpler as it corresponds to a single global affine transform, so $\overline{\mathbf{R}}^i = \mathbf{R}_j = \mathbf{R}$ for \mathbf{A}_q . In a typical morphable model fitting procedure, all the piecewise affine transforms are already computed. Computing the Jacobian matrix \mathbf{J} is easy and fast based on (3-17) and (3-18). Because $Z(\Delta\mathbf{p}, \Delta\mathbf{q})$ is a sum of two summed squared measurements, it is easy to see that its Gaussian-Newton Hessian is the sum of Gaussian-Newton Hessian for the weighted texture error function and K times the Gaussian-Newton Hessian of the shape error function. The same conclusion is true for the steepest gradient descent images. With the computed Jacobian matrix, the optimization procedure is nothing special other than a normal inverse compositional algorithm.

3.6 Experiment Results and Discussion

The same face database is used to conduct our experiments for this chapter. 40 shape parameters are used to capture 98% of the shape variation, and texture subspace has a dimension of 66 to account for 98% of the texture variation. With four extra global pose parameters, we have a total of 44 model parameters.

3.6.1 Constrained Hybrid Model Fitting Optimization

In our constrained hybrid optimization scheme, shape is directly estimated from block motion estimation and local projection models. Integration of such heuristic information is the key for a high convergent rate with only around 10% extra computation. Our algorithm is compared to standard inverse compositional algorithm. Fig. 3.3(d) shows the fitting with standard

inverse compositional algorithm. Almost all other landmark points are displaced. It is commonly seen that model eye points converge to eyebrow area in the image. Fitting with our algorithm is shown in 3.3 (e). Figures 3.3 (a) to (c) show a typical scenario of our hybrid search. Fig. 3.3 (f) compares texture error evolution curve of our constrained hybrid model fitting algorithm with standard normalized inverse compositional AAM algorithm.

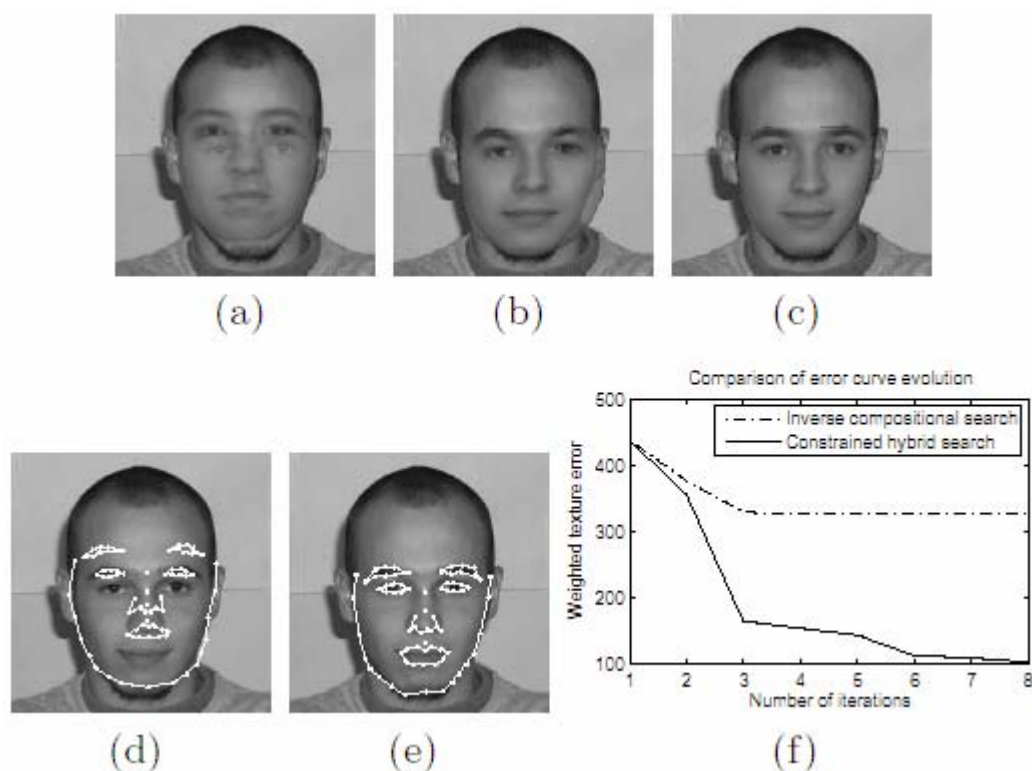
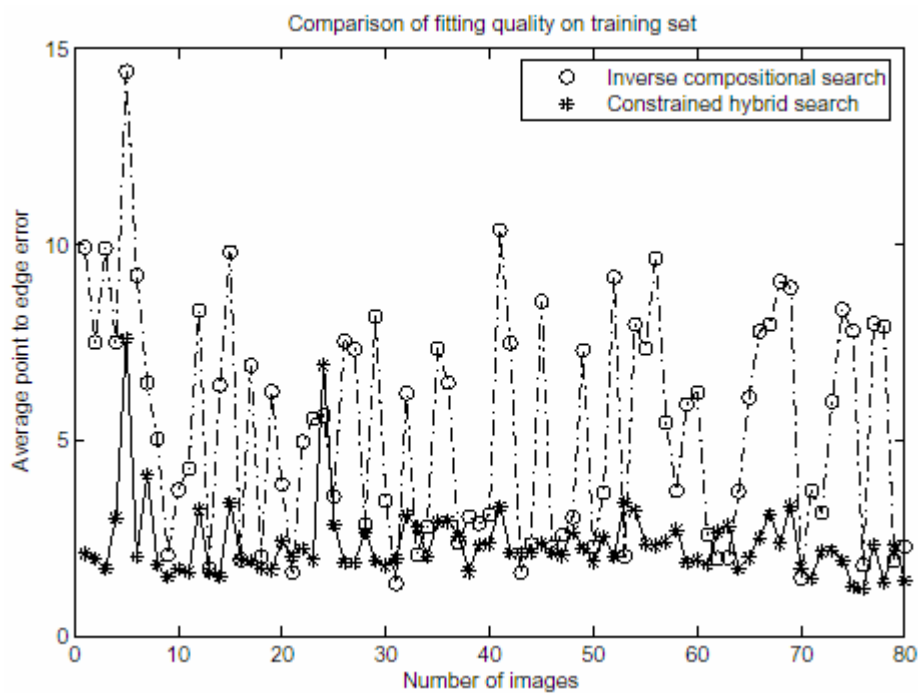


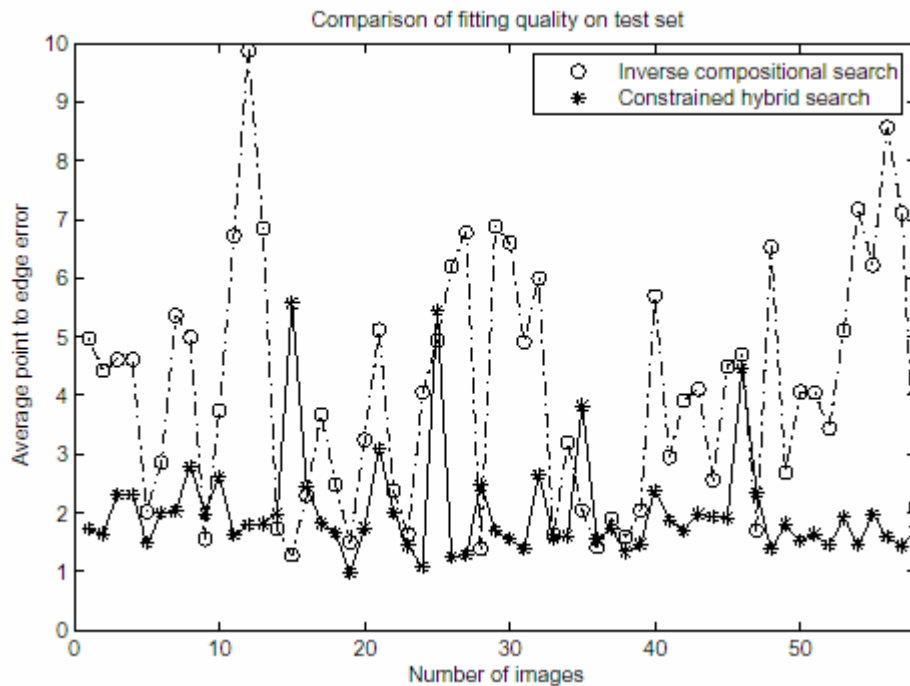
Figure 3.3 Comparison of hybrid search and original inverse compositional AAM search. (a)-(c) Hybrid search process at iteration 1, 2 and 6. (d) Inverse Compositional AAM search. (e) Constrained hybrid search. (f) Evolution of error curve

Though the weighted texture error is used as the evaluation function, it does not reflect the model fitting quality strictly, especially when the novel face texture pattern is beyond the representation power of the trained texture subspace. In that case, the texture reconstruction

error might remain large even when all landmark points are perfectly located. A reasonable evaluation is to measure how good the model points converge to their desired positions in the image. For all 138 images, we manually labeled the landmark points and created a distance map for each image. The model fitting quality is then measured by the average point to edge distance. Fig. 3.4 compares the performance of the inverse compositional algorithm and our proposed algorithm by plotting the average point to edge error for all images in the training set and the test set. The superiority of our constrained hybrid algorithm is clearly manifested.



(a)



(b)

Figure 3.4 Fitting errors on (a) Training set. (b) Test set.

All the tests are conducted with exactly the same initial model parameters and stop criterions.

Inverse compositional search has a quite bad performance due to the fact that our face database consists of images originating from various sources. With our constrained hybrid optimization method, the average point to edge error reduces from 4.3900 to 2.0359 on the training set, and 5.3869 to 2.4713 on the test set. Both show considerable improvement.

Based on the results in Fig.3.4, we could further generate error density functions. All 138 images are used to estimate their probability density functions of the fitting error. Their cumulative functions are plotted in Fig. 3.5. Given a threshold from x-axis, we can read a number from y-axis, which is the percentage of images that have equal or less fitting errors than the threshold. If we assume the model fitting is successful when the point to edge error is

below a threshold, then Fig. 3.5 is actually a plot of the convergence rate versus threshold. Apparently our hybrid optimization has a significantly higher convergence rate compared to standard inverse compositional algorithm.

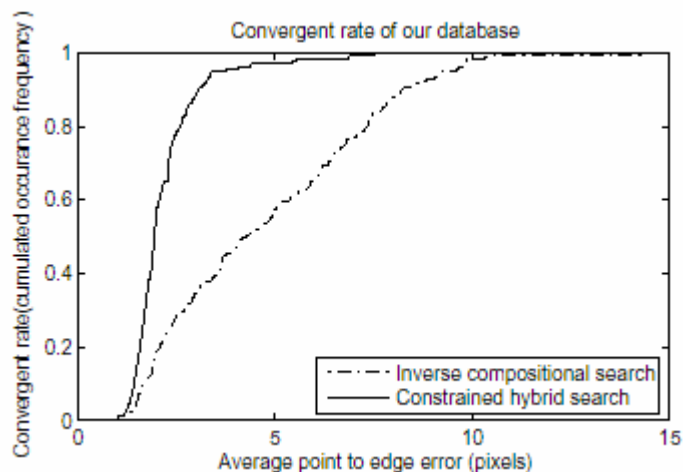


Figure 3.5 Cumulative functions

There are 3 key components in our constrained hybrid optimization scheme: the unbiased evaluation function, the integration of direct shape estimation from block motion estimation, and the local projection models. To see how each of these components plays a role in the hybrid optimization, 4 different algorithms are compared: the original Normalized Inverse Compositional algorithm (NIC), the NIC with motion integration (NIC_MO), the NIC with both motion integration and local projection models (NIC_MO_LPM), our hybrid model with unbiased evaluation function (NIC_MO_LPM_cr). Table 3.1 summarized their performance with average point to edge errors.

Table 3.1 Average point to edge error for different algorithms

Database	NIC	NIC_MO	NIC_MO_LPM	NIC_MO_LPM_cr
Training set	4.3900	2.3924	2.3440	2.0359
Test set	5.3869	2.7506	3.0055	2.4713

Table 3.1 shows that our constrained hybrid optimization scheme has the best performance.

Unbiased error function proves to be a better evaluation function than traditional error function as it truly reflects the fitting quality on the image frame. An exception is NIC_MO_LPM error is larger than NIC_MO error on the test set. [This is statistically normal fluctuation considering the fact that we only have 55 test images.](#) In fact, if we look at this problem from a different perspective, 27 images of the 55 perform better with local projection models, comparing to 20 of the 55 perform better without projection models. It is still justified that local projection models improve the overall performance.

3.6.2 Experiments on the JAFFE face database

[Though our constrained hybrid model is trained on the 80 neutral faces in our training set, the model fitting algorithm performs very well on the face images with rich facial expressions in the JAFFE database.](#) The only pre-processing we conducted is to scale original 200 by 200 images to standard size 256 by 256.

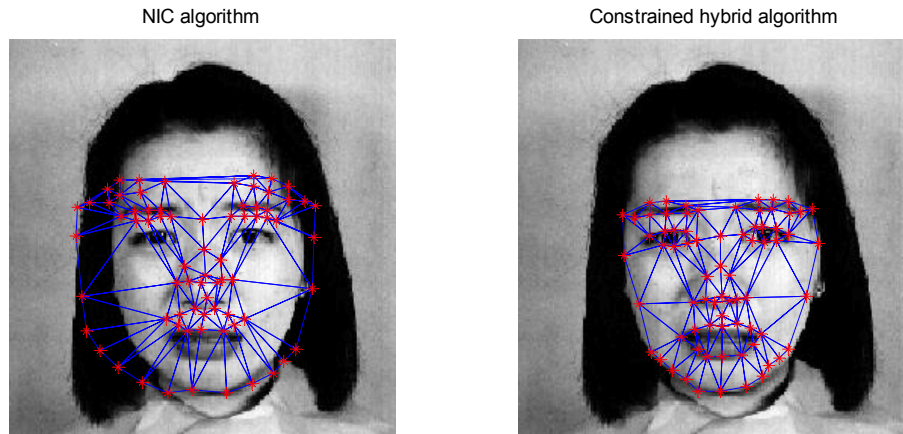


Figure 3.6 Model fitting results: a) Inverse compositional algorithm. b) Constrained hybrid algorithm

Fig. 3.6 shows an example of converged face model overlapped to images. Fig. 3.7 plots average point to edge error for all images in JAFFE. Fig. 3.8 is the plot of cumulative functions. Table 3.2 shows average point to edge error for different algorithms. From our face database to JAFFE, the performance only degenerates very slightly with an average point to edge error of 2.9. It proves to be efficient and robust even in the presence of rich facial expressions, unseen in our training set.

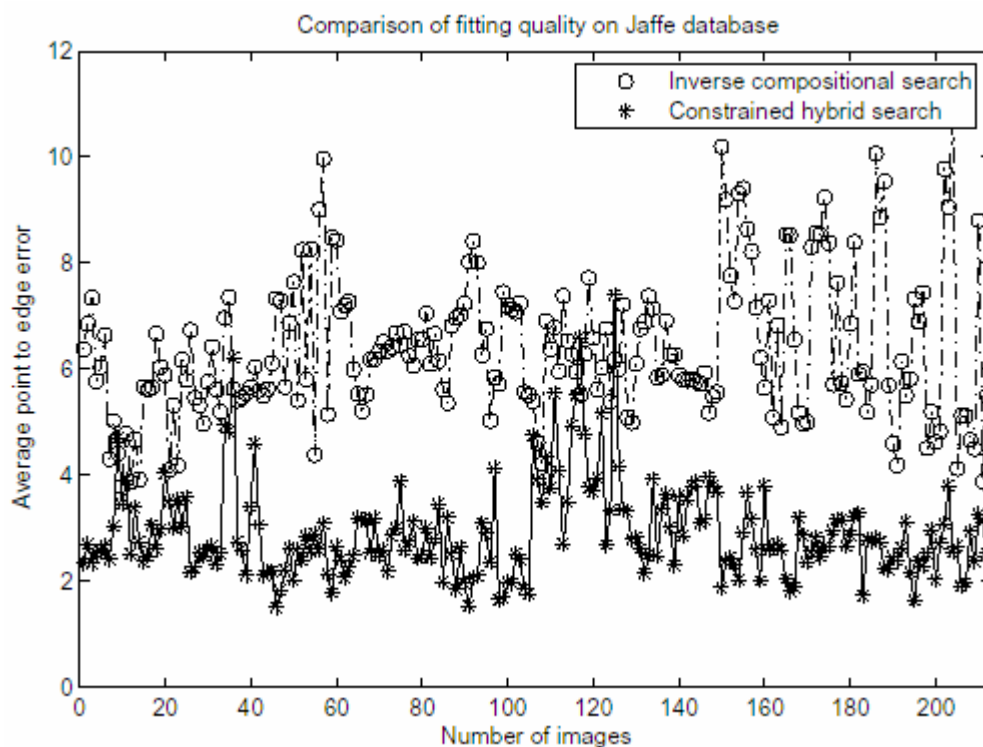


Figure 3.7 Model fitting errors on JAFFE

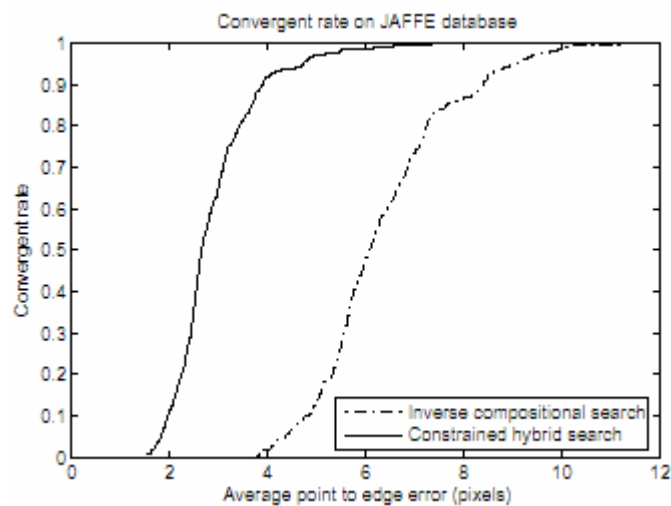


Figure 3.8 Cumulative density functions

Table 3.2 Average point to edge error for different algorithms

Database	NIC	NIC_MO	NIC_MO_LPM	NIC_MO_LPM_cr
JAFFE	6.3646	4.1144	3.3958	2.9133

3.7 Conclusion

In this chapter, we presented a constrained hybrid optimization algorithm to solve the general morphable model fitting problem. Designing a robust and efficient modeling algorithm is very important as feature detection is an inevitable step for a lot of face applications. Our constrained hybrid algorithm features a novel error evaluation function which is an unbiased error estimate of model fitting quality on the image frame. Shape estimate from block motion estimation and local projection models is incorporated into the gradient-descent optimization procedure. They play a role in the parameter updating by acting as a shape constraint in the optimization process. As a result, our model fitting algorithm performs much better than general optimization algorithms that purely rely on analytic solutions. One apparent conclusion is that heuristic information in an image itself is abundant and could see better of image local terrains. Blindly minimizing a texture error is inefficient and error-prone. Experiments on our face database and JAFFE database shows that our constrained hybrid model fitting algorithm could achieve a high convergent rate even when presented with images of a large variety of illuminations, image backgrounds and facial expressions. Large image scaling, rotation and partial occlusion are not tested.

CHAPTER 4 : FROM 2D TO 3D: 3D FACE STRUCTURE

EXTRACTION

We live in a three dimensional world. For any normal human being, perceiving and understanding the 3D world is just a piece of cake and it is as natural as other spontaneous human activities like eating and sleeping. Such a simple activity, however, seems formidable for even the most advanced computer in the world. In fact, the ability to infer 3D from 2D projections is the result of a complex mechanism inside human brain that hasn't been resolved so far. Nevertheless, many 3D algorithms have been proposed and some commercial software has been developed in the past decades. For most 3D face applications, the reconstruction of 3D face surface is essential. Current 3D applications include 3D face motion estimation, 3D pose estimation, 3D face animation and 3D recognition etc. In this chapter, we will present our 3D face structure extraction algorithm given aligned faces in 2D images. Different aspects of the 3D reconstruction problem, like different poses, possible occlusions, uncertain feature correspondence etc, are discussed. In particular, face contours are dynamically generated to match extracted face contours in 2D face images. [As a result the reconstructed 3D faces are more realistic compared to the results in \[27\].](#)

4.1 Literature

Obtaining accurate 3D face surface information is crucial for 3D face applications. There are some special devices that can generate 3D range data directly from human faces. Typical devices include stereoscopic cameras and laser-based cylindrical scanners, such as those

produced by Cyberware [35]. The resulted range data yields accurate face depth information. It can be very useful for realistic face animation and face modeling. However, even though these devices are now not as expensive as they used to be, most of their applications are still in the computer graphics field for the purpose of vivid face animation. It is possible that one day 3D photographing will be common enough for daily lives. Until then, photos and videos are the main media to preserve face information and they will continue to be the sources for the mainstream research on 3D face modeling.

4.1.1 Shape Reconstruction by Modeling

We briefly introduced several Shape-from-X methods in the previous section. Generally, shape-from-X refers to techniques to reconstruct a scene without any assumption about the target. Object structure can be inferred from shading, texture and other natural properties. If we have some prior knowledge about the structure or the texture pattern of the object, surface reconstruction can be realized by constructing a 3D model that best fits the description of the object. In our scenario, face surface reconstruction then amounts to the adaptation of a 3D face model to a face image viewed from a possibly arbitrary viewpoint. There are some commercial software packages available that enable a user to construct personalized 3D face models. Modeling a face with 2D or 3D generic face models from images or video sequences has been carried on for several decades, almost as long as the history of automatic face recognition. Face modeling has a lot of applications in the entertainment industry. Driven by the big market, face modeling techniques have been commercialized and widely applied in the developing of interactive video games and animated movies in Hollywood, while face

modeling for recognition purpose is relatively immature and still at the developing stage.

Covering a detailed history of face modeling is beyond our topic here. In the following, we will mainly focus on the aspect of 3D face modeling and skip those 2D related face modeling issues. Face surface could be represented with any typical surface descriptors like Coons surface patch, B-spline surface, triangle mesh etc. The most popular and practical form is triangle mesh. Face modeling usually includes three steps. First, feature points are extracted from the face images, and point correspondences are built up (by either dense or sparse feature matching techniques). Second, the head pose, the camera parameters and the 3D feature points are computed based on the extracted feature correspondence in the images. At last, a 3D face model is fitted to the 3D feature point set. Sometimes 3D feature points are not explicitly reconstructed. Instead, a 3D face model is morphed so that its projections best match the extracted 2D feature point set directly. Face modeling algorithms could be classified according to the model descriptors, morphing techniques, matching features (dense or sparse, point or texture etc) etc. From the statistical point of view, most 3D face modeling algorithms fall into two categories: unsupervised generic modeling algorithms and statistical modeling algorithms. In another word, generic modeling algorithms adopt some empirical expert-defined models and edit model points directly or follow some morphing functions to generate desired individual models. Throughout the modeling process, no training or supervision is involved. On the other hand, statistical modeling algorithms first learn the distribution of all kinds of variations from some real face samples in the training phase. Then typical face parameters are chosen to initialize the face model. The model morphing process is governed by the learned distribution of model parameters, assuming the training faces are

general enough to represent the human population, or at least the population of the specific interest. Compared to the unsupervised face modeling algorithms, statistical modeling algorithms have constrained parameter spaces, therefore morphed face models are still reasonable faces. However, statistical modeling algorithms need a training face database and some labor work might also be necessary. Moreover, it is questionable whether the training faces could be universal enough.

There also exist some algorithms fusing both modeling techniques [49]. They adopt empirical generic models and use training databases to compute typical distributions for model parameters.

4.1.1.1 Generic Face Modeling

There are several famous face models that are widely used by researchers in the computer vision and computer graphics fields. CANDICE [52] is a 3D wire frame model developed at Linkoping University in the eighties. Fig. 4.1 shows frontal and profile view of this model.

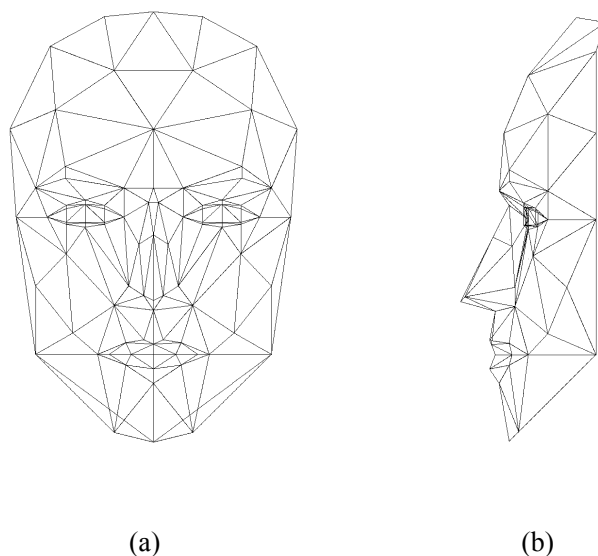


Figure 4.1 Candide-3 face model: a) Frontal view. b) Profile view.

CANDICE only consists of approximately 113 vertices and 183 triangles. Such a small number of vertices allow for fast face reconstruction with moderate computation. Vertices are controlled by global and local action units (AU). Both inter-personal variations (due to different people) and intra-personal variations (facial animation) are considered in the design of AUs. A 3D shape is represented as:

$$x = \bar{x} + S\sigma + A\alpha \quad (4-1)$$

where \bar{x} is the mean shape, Matrix S is the predefined shape units, whose columns are a set of shape basis. Similarly, A consists of the animation units. Shape units enable operations such as the widening (or narrowing) of eyes. Animation units allow the deformation of the face mesh to some predefined facial animations. A 3D face is completely determined by shape coefficient vector σ and animation coefficient vector α . CANDICE is a popular 3D face model as it can model both intra-personal and inter-personal variations. It has been widely applied to reconstruct face structures and analyze facial expressions from images or video sequences [53] [54].

In the computer graphics area, there are quite some different 3D face models. Besides CANDICE, there is EPFL [55] developed at University of Geneva. EPFL uses a hierarchical configuration to generate different facial expressions arising from speeches and emotions.

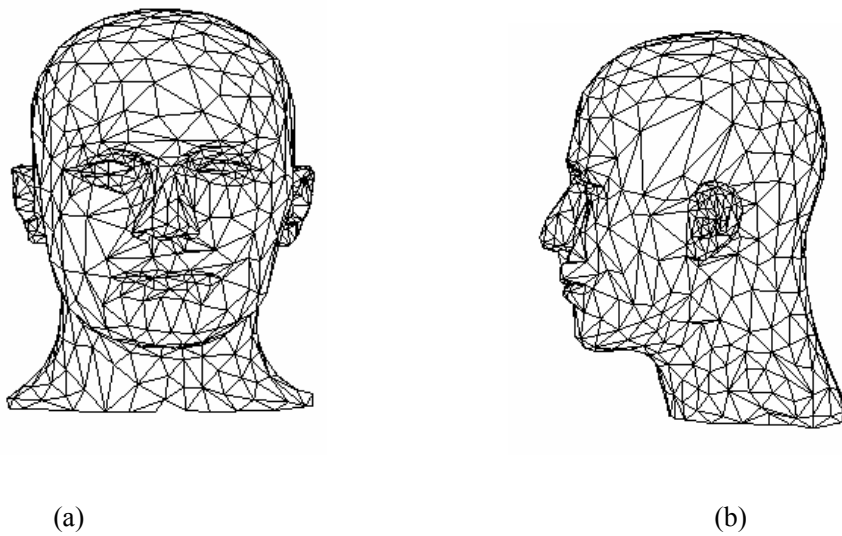


Figure 4.2 Face model created at Univ. of Washington. a) Frontal view. b) Profile view

Fig. 4.2 shows the 3D face mesh model created at University of Washington [56]. Unlike CANDIDE whose model parameters are explicitly associated with physical face characteristics and facial expressions, this model tries to catch as much face details as possible by using 689 vertices and 1355 triangular facets. In fact, we work with a face model revised based on this 3D face model. It balances the requirements for accurate surface description and acceptable computational complexity.

Each 3D face model is made up of surface patches. Each surface patch is a triangle with three control points (vertices) to control the position and orientation of that specific patch. Neighboring triangles share control points (or triangle edges) to form a closed surface. Usually such a face surface is described by two files. The first file contains a list of the 3D coordinates of all triangle vertices. The vertices file is typically arranged as follows:

$V_1 \quad X_1 \ Y_1 \ Z_1$

$V_2 \quad X_2 \ Y_2 \ Z_2$

$V_3 \quad X_3 \ Y_3 \ Z_3$

...

The second file indicates the indices of three vertices for each triangle. The order of vertices of a triangle is given in either clockwise or anticlockwise manner so that any surface normal is pointed outwards. The triangles file appears as follows:

Triangle₁ $V_3 \ V_1 \ V_4$

Triangle₂ $V_2 \ V_1 \ V_5$

Triangle₃ $V_1 \ V_2 \ V_6$

...

Given a 3D object, there are many ways to approximate its surface with triangular patches and generate the above two description files. One would desire a triangulation scheme that can preserve as much shape information as possible of the original object. Therefore, sampling should be dense around surface locations of high curvature, while a minimal number of points are enough for a nearly flat surface. Building optimal 3D face mesh out of an object with fixed number of vertices is another interesting research topic in the computer graphics field. It is beyond our scope. Here we just assume the generic 3D face model we will use is already optimal for face surface description and our focus is mainly the model morphing and fitting task.

Besides face shape, texture is another important face feature. The irradiance at an image point measured by a camera is affected by many aspects, including surface properties such as surface roughness, absorption, reflectance, and also the surrounding illumination environment. Usually a Lambertian reflectance model is assumed to model the illuminating process. Under

this assumption, a surface irradiance is the product of the surface albedo and the cosine of the angle between the lighting incident direction and the surface normal at that specific location. Recovering surface texture is a complicated procedure. Usually it is tackled with texture-mapping techniques for the purpose of realistic synthetic faces from novel viewpoints. Generic face models have been widely applied for face analysis and recognition. In the following, several related papers are briefly reviewed based on different image source formats.

1. Modeling from video sequences

One of its most popular applications is to estimate face structure from motion and use the modeling result for semantic coding. For example, in [54], CANDIDE face model is adapted to a person's face in a video sequences. The adaptation algorithm starts with global fitting to adjust the model size and orientation, followed by locally adapting the model details to match those detected facial feature points in each video frame.

P. Koch [57] estimated face structure from motion by employing analysis-by-synthesis approach. Basically, a face mesh is imposed on a video sequence and refined by minimizing the intensity differences between the real face images and the synthesized ones.

2. Modeling from still images of different viewpoints.

Pighin et al [58] uses customized face models to synthesize different facial expressions. Feature points are labeled manually in 5 or more images of the same person. 3D coordinates of these points and pose parameters are estimated using a partially linear least square method. A generic model is adapted to the reconstructed 3D points so that these 3D points are exactly matched, while other points in the generic model are morphed using the RBF interpolation

function. Face texture is handled using either view-independent texture mapping or view-dependent texture mapping techniques. Once a 3D structure and a global texture are known, images with novel expressions are synthesized or ‘copied’ by interpolating local geometries and textures of known 3D models.

In [59], a generic face model is morphed with a complex deformation function. The 3D model is a composite of 3 models: edge model, color model and a wire frame model. Edge detection and color segmentation algorithms are adopted to generate edge and color fields respectively. The model matching process is essentially the minimization of an objective energy function. The energy function includes 3 components: edge and color energy functions that measure the quality of matching for edge and color features, as well as a 3rd component measuring the deformation cost in order to prevent the 3D model from deforming too much. The wire frame model is used to deal with occlusion and generate face contours.

In [27], a 3D face modeling algorithm is proposed to extract 3D face structure from images. A generic face is morphed to generate a specific face structure according to an explicit cubic polynomial in 3D. Choosing of the cubic morphing function is dictated by its morphing ability, as well as the consideration of the algorithm complexity. The feature correspondence problem is bypassed with the help of the distance map technique, which bears similar ideas as the edge field and the color field in [59].

4.1.1.2 Statistical Face Modeling

Unlike generic face modeling techniques, statistical face modeling techniques take advantage of statistical information of typical human faces. Most statistical face modeling algorithms follow the analysis-by-synthesis protocol. In another word, statistical face models have to be

generative models. Under this protocol, image interpretation could be formulated as face matching problem. A 3D face model is adjusted and morphed so that it could generate an imaginary face image which is most similar to the unknown face image. Apparently, face models of this kind should be general enough to generate any arbitrary plausible human face. On the other hand, any synthesized instance has to be a legal face. Generative face models should rarely instantiate unlikely faces.

For statistical face models, a crucial assumption is that any face can be generated by a linear combination of a set of prototype faces. This linear combination is not a simple addition of face images at the pixel level. Addition of raw images usually won't create a valid face image, but a face-like image with double face contours and blurred facial components. In order to justify the linearity of the face space, training faces have to be in full correspondence with each other. A complete face model should have the ability to model various faces in terms of face shape and face texture. Only independent linear operations of face shape and face texture could possibly satisfy the linearity requirement for the face vector space. For statistical face modeling, the construction of a morphable model requires the establishment of correspondences across all training face images, followed by analyzing and learning the variations of face shape and texture. The training face database is assumed to be general enough to reflect variations among human faces in reality. After a generative model is constructed, the alignment and analysis of an unknown face image is carried out by seeking the optimal model parameters so that the synthesized face best resembles the unknown face.

Some 2D statistical face models were proposed around a decade ago. The most popular of them are the AAM [8], the ASM [12] and the MMM [31] etc. Brief introduction of them could

be found in Chapter 2 and Chapter 3. Though the exact definition of face shape might be different and various minimization algorithms are used to match unknown faces, these 2D statistical modeling algorithms have some characteristics in common: shapeless textures are extracted by warping face patches to a common shape frame. Face shape and texture form different vector spaces respectively; the matching process is formulated as an optimization problem, and implemented as an iterative procedure using either traditional gradient descent optimization algorithms or linear regression algorithms. 2D statistical models have been successfully applied to interpret medical images as well as face images. There exist a variety of extended algorithms based on the AAM, the ASM etc. As a result, face modeling performance is improved from different aspects. For instance, in [14], wavelet analysis is adopted to replace the PCA analysis for robust face matching. Inverse compositional alignment algorithm is incorporated in [17] for real-time applications etc.

Though intrinsically, differences among face images are caused by different face identities. The appearance of a face in an image is also affected by the changes in pose, illumination, facial expression etc. In nature, 2D models are awkward to distinguish different poses and illumination conditions. In [22], a nonlinear PCA is used to tackle this problem. However that algorithm is complicated and inefficient. Since any 3D face geometry can be easily associated with its projected 2D images given a projection model (perspective projection model for example), modeling a face in 3D is the natural solution to the different pose problem. According to physical imaging models, any image intensity value can be interpreted as a function of the corresponding surface albedo, the surface normal and the surrounding illumination environment. Apparently, a 3D morphable model has the advantage of applying

physical laws to model both different poses and illumination conditions.

Some 3D face models are natural extensions of 2D statistical face models. An algorithm is presented in [60] to model faces from images or video sequences with large pose variation. A 3D face shape model is constructed by training a set of 3D faces (represented by 3D feature points) that are reconstructed from manually picked 2D feature points in face images. A 2D shape-and-pose-free texture model is constructed afterwards. During the matching process, a loss function is minimized using a direct exhaustive searching strategy. J. Xiao etc [25] proposed a combined 2D+3D AAM. A set of trained 3D shape modes are adopted to constrain the behavior of the 2D AAM search. As a result, the underlying 3D structure and projection parameters are simultaneously recovered once the 2D search converges.

Perhaps the most popular 3D morphable model nowadays is the 3D Morphable Model (3DMM) developed by V. Blanz and T. Vetter [5]. Unlike all previous 3D models, this one needs 3D laser data for the training purpose. Each 3D face is sampled densely so that after triangulation, enough details of the face structure are preserved. The same triangulation topology is used for all training faces. After all training faces (about 200 3D scans) are in full correspondence, each 3D face is described with a 3D shape vector formed by concatenating the 3D coordinates of all vertices, and a texture vector by concatenating the RGB values of the vertices as follows:

$$\mathbf{S}_i = \{x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n\}$$

$$\mathbf{T}_i = \{R_1, G_1, B_1, R_2, G_2, B_2, \dots, R_n, G_n, B_n\}$$

Notice how similar they are to the 2D shape vector and texture vector defined in the 2D AAM.

Training the 3DMM is straightforward. A model face can be represented in the subspaces as:

$$\mathbf{S}_{\text{model}} = \bar{\mathbf{S}} + \sum \boldsymbol{\alpha}_i \mathbf{s}_i \quad (4-2)$$

$$\mathbf{T}_{\text{model}} = \bar{\mathbf{T}} + \sum \boldsymbol{\beta}_i \mathbf{t}_i \quad (4-3)$$

where $\{s_i\}$ and $\{t_i\}$ are respectively the eigenvectors for the shape subspace and the texture subspace. With enough training faces, the distributions of $\{\boldsymbol{\alpha}_i\}$ and $\{\boldsymbol{\beta}_i\}$ can be approximated using multivariate normal distributions. These distributions are used to regulate synthesized 3D faces to avoid non-face synthesis instances.

Matching the 3DMM to an image also follows the analysis-by-synthesis strategy. The goal is to minimize a cost function measuring the difference between the synthesized image and the unknown face image. Recovering any 3D face structure from a single image is an ill-posed problem. Therefore the a priori distributions of the morphing parameters are incorporated to the cost function as constraints. The 3DMM utilizes the stochastic gradient descent algorithm to find the optimal model parameters.

The 3DMM is very powerful as the training 3D faces have dense shape and texture data. It could synthesize realistic face images of large pose variation and under different illumination conditions. For traditional sparse 3D models, surface interpolation is an indispensable operation in order to describe a continuous surface. However, it is no longer necessary as the 3DMM deals with dense face data. Though the 3DMM has the ability to model any face of arbitrary poses and illumination conditions, it is not an almighty algorithm. First, the 3D training faces are generated from laser scans and they lie in a high dimensional space. Building the training set and constructing the model is not a trivial job. Second, the optimization procedure is very slow and takes about 50 minutes [5]. So it is desirable to have a less complicated model that could work much faster while maintaining the same or similar

performance. In [61], the inverse compositional image alignment algorithm (refer to Section 3.2 for details) is extended from 2D to 3DMM. The computation time is greatly reduced.

The face recognition algorithm based on the 3DMM has proven to be very successful in the face recognition vendor test (FRVT) in 2002, which was an assessment conducted by the U.S. government. Its performance in the presence of large pose and illumination variation is superb and impressive.

Some hybrid models take advantage of both generic models and statistical models. For example, the CANDICE generic model is used in [60] to track faces from videos. The model parameters are updated using a regression model learned from a set of training faces. The algorithm preserves the simplicity of a generic face model. At the same time, it uses statistical updating rule learned from training faces.

4.2 Our Approaches

Up till now, one of the most successful ways to recover a 3D face structure is to analyze the face with a 3D face model. Different face modeling algorithms adopt either sparse or dense mesh models according to their specific scenarios. Matching an unknown face is realized through minimizing the point difference or texture difference. Generally speaking, dense face models are more complicated, yet are capable of modeling face details. Optimization schemes driven by texture difference are usually time consuming.

In our work, a sparse face model with over 70 feature points is adopted and the face matching task is based solely on the feature points. So the main challenge to us is to extract facial feature points automatically and establish the correspondence across different views. In [27],

an edge map is extracted by locally tracing edges given some starting points. The Canny edge detector is used to generate the edge map in [59]. Though the feature extraction step is very crucial for the whole modeling, edge-detection based algorithms often fail to locate features accurately and robustly. This is natural since human face images in nature are intensity patterns than other simple line drawings. In order to have robust and accurate alignment results in 2D images, we exploit a view-based AAM to automatically align 2D faces of different viewpoints.

Another challenge is that both the 3D face structure and the pose parameters are unknown. Algorithms such as [54] try to decouple the pose estimation from the 3D structure estimation, and the pose parameters are estimated first by adapting the model size and orientation globally. It assumes that the morphed face structure is within a reasonable error displacement after the pose is fixed. In our approach, the pose parameters and the face structure are solved simultaneously in an iterative minimization procedure.

In a previous work [27], a generic face model is morphed with a cubic polynomial function and the distance map technique is adopted to bypass the feature correspondence problem. We rephrase the polynomial function as a non-orthogonal linear transform. Now the iterative procedure to seek optimal parameters is more straightforward. Two different optimization schemes are utilized and compared. Furthermore, face contours are dynamically generated and incorporated to the cost function. Without this constraint, the morphed face model would look very unrealistic.

4.2.1 3D Generic Model

The generic model we use is revised from the model at University of Washington as shown in Fig. 4.2 previously. Though the original model has 1335 triangles, only a small portion of them are related with the face structure. We remove those related with the neck and hair part since they are usually considered not to encode any face identity information. The original model is tailored to a new compact generic model as shown in Fig. 4.3.

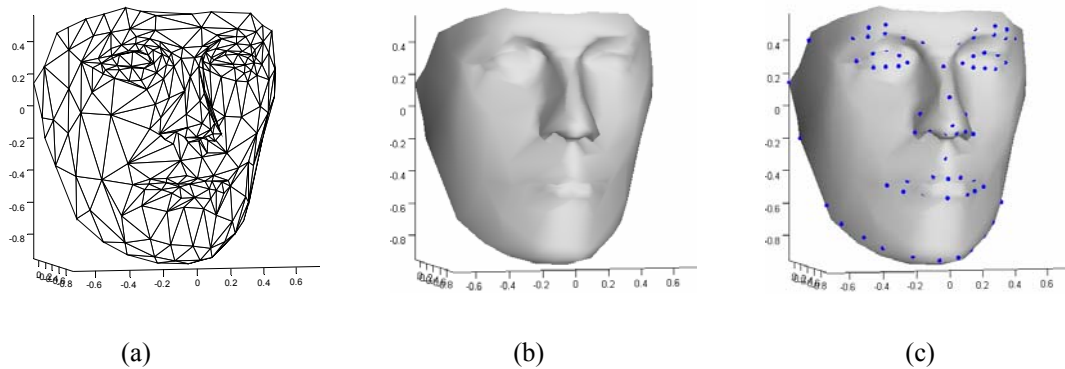


Figure 4.3 Revised generic model: (a) Mesh model. (b) Solid model. (c) Selected features for structure estimation

The number of feature points in one image used to morph the generic face varies from application to application. Originally we intended to take all of the Face Definition Parameters (FDP) points defined in the MPEG-4 protocol as the features to be extracted from face images. However, some of them are difficult to identify and hard to locate in face images. Points that are used to mark the teeth and the tongue are often invisible. Fig. 4.3(c) shows the 73 feature points we have chosen. Most of them are located on the distinctive edges of facial parts such as the eyes, the mouth and the face outline etc. The more points on one facial part, the more

important role that part plays in the face structure estimation. These points are carefully selected to balance the requirement for a compact yet complete underlying face description.

4.2.2 Weak Perspective Projection

A camera model is needed to project the 3D face model to the 2D image plane. General camera parameters include the focal length f and the relative position of the camera to the object in the world coordinate system. Let the camera coordinate system be spanned by the unit vector triple $(\mathbf{X}', \mathbf{Y}', \mathbf{Z}')$, and the face coordinate system be spanned by $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. The offset from the origin of the world coordinate system to that of the camera coordinate system is represented by the translation vector \mathbf{T} . We use capital letters (like \mathbf{P} or \mathbf{P}') to denote points in the 3D coordinate system, whereas their projections in the image plane are denoted by lower case letters. The camera model is shown in Fig. 4.4.

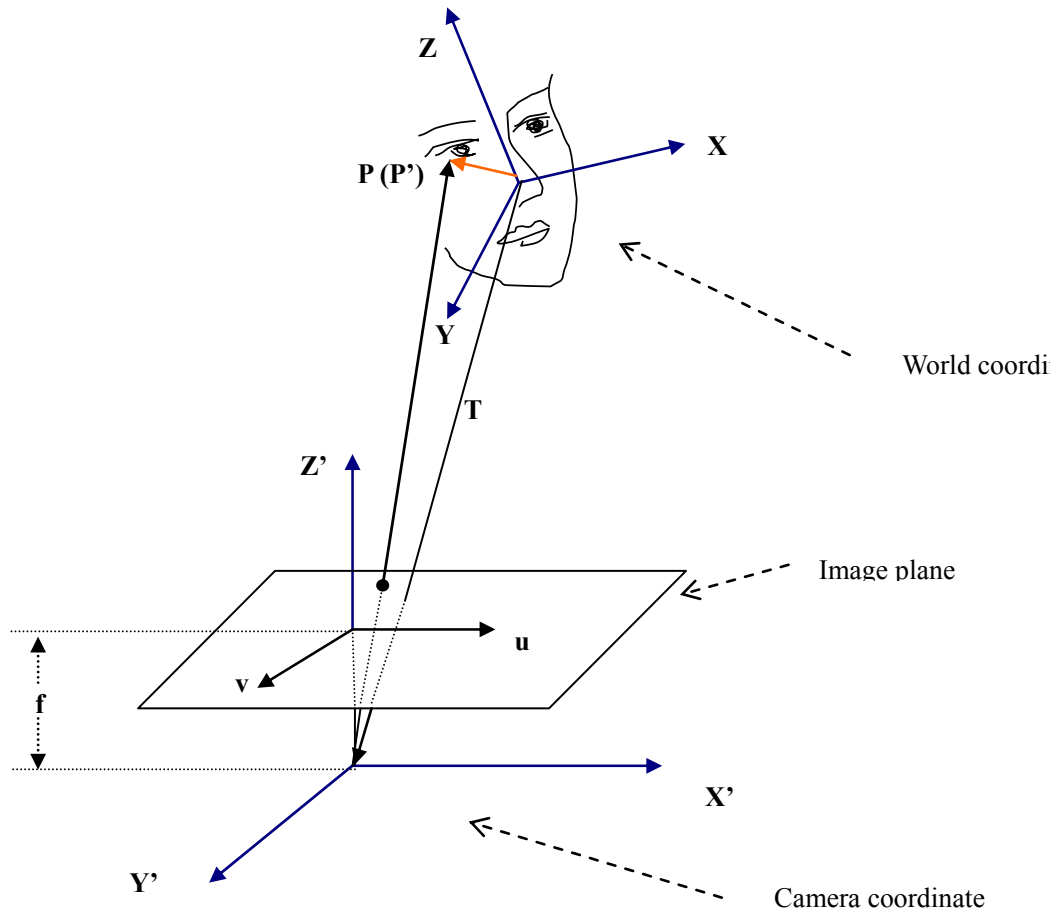


Figure 4.4 Camera model

The 3D point \mathbf{P} in the world coordinate system is transformed to the camera coordinate system as \mathbf{P}' :

$$\mathbf{P}' = \begin{bmatrix} (\mathbf{P} - \mathbf{T}) \cdot \mathbf{X}' \\ (\mathbf{P} - \mathbf{T}) \cdot \mathbf{Y}' \\ (\mathbf{P} - \mathbf{T}) \cdot \mathbf{Z}' \end{bmatrix} \quad (4-4)$$

The perspective projection of \mathbf{P} in the image plane is $\mathbf{p} = \begin{pmatrix} u \\ v \end{pmatrix}$, where

$$u = f \frac{(\mathbf{P} - \mathbf{T}) \cdot \mathbf{X}'}{(\mathbf{P} - \mathbf{T}) \cdot \mathbf{Z}'}; \quad v = f \frac{(\mathbf{P} - \mathbf{T}) \cdot \mathbf{Y}'}{(\mathbf{P} - \mathbf{T}) \cdot \mathbf{Z}'} \quad (4-5)$$

Rearrange the camera coordinate basis vectors in a matrix as $\mathbf{R} = [\mathbf{X}'^T; \mathbf{Y}'^T; \mathbf{Z}'^T]$. \mathbf{R} could be regarded as a composite rotation matrix by combining three rotation operations

respectively around three basis vectors with angles $(\theta_x, \theta_y, \theta_z)$. In detail,

$\mathbf{R} = \mathbf{R}_x \cdot \mathbf{R}_y \cdot \mathbf{R}_z$ with

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix}; \mathbf{R}_y = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix}; \mathbf{R}_z = \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(4-6)

Let $\mathbf{T}_p = -\mathbf{R} \cdot \mathbf{T}$, equation (4-4) could be rewritten as:

$$\mathbf{P}' = \mathbf{R} \cdot \mathbf{P} + \mathbf{T}_p \quad (4-7)$$

The orientation parameters $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ and the translation parameters \mathbf{T}_p constitute the pose parameters. With the focal length f of the camera in equation (4-5), a complete perspective projection has 7 free parameters.

The weak perspective projection model is an adequate approximation of the perspective projection model as long as the distance between the object and the camera is much bigger compared to the size of the object. Under this assumption, it is reasonable to treat

$\frac{f}{(\mathbf{P} - \mathbf{T}) \cdot \mathbf{Z}'}$ as one scaling factor s_w which is constant for all points on the 3D object.. Let

$\mathbf{R}_w = [\mathbf{X}'^T; \mathbf{Y}'^T]$ be the 2 by 3 partial rotation matrix. Since the 3rd coordinate basis \mathbf{Z}' is the cross product of the first two, $\mathbf{Z}' = \mathbf{X}' \times \mathbf{Y}'$, the full rotation matrix \mathbf{R} could be fully reconstructed from \mathbf{R}_w . Equation (4-5) is reformulated as:

$$s_w * \mathbf{R}_w \cdot \mathbf{P} + \mathbf{t}_w = \mathbf{p} \quad (4-8)$$

where $\mathbf{t}_w = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$ is the translation vector in the image plane, s_w is the world-to-image

scaling factor. Together with the three rotation angles $\theta_x, \theta_y, \theta_z$ implied by \mathbf{R}_w , there are 6

parameters in total for the weak perspective projection model.

4.2.3 2D Feature Extraction with the View-based AAMs

In order to reconstruct the underlying 3D face structure, the same set of feature points projected onto different image planes have to be located and aligned. This is not an easy task.

The AAM is known to be a 2D statistical appearance-based model that has robust alignment results. However, it tends to break down when the unknown face is subjected to a large angle rotation. One solution is to use a complete 3D face model to track and align all feature points.

This is more like an egg-and-chicken problem since reconstructing the 3D face structure is our goal instead. Another approach is to use 2D models based on nonlinear analysis [22]. The easiest way, however, is to build a set of models for different viewpoints [19][20][21].

Chapter 2 and 3 focus on the in-depth study of several improved AAM algorithms. Naturally, it is desirable and straightforward to extend them to view-based AAM algorithms.

Let the orientation of a face object in the space be represented by the pitch/yaw/roll. Naturally the pitch (rotation around the side-to-side axis to raise or lower the head) and the roll (rotation around the front-to-back axis) have a small range of variation compared to the yaw (rotation around the vertical axis). For the faces in our face database, 4 pose categories are defined according to their yaw (i.e. azimuthal angles). Table 4.1 shows the range of angles for each pose category we define. 4 models are developed for the 4 pose categories respectively. Due to the symmetric nature of human faces, these 4 models are enough to cover the azimuthal rotation of 180 degrees.

Table 4.1 Azimuthal range for 4 different view-based models

Pose category	Category 1	Category 2	Category 3	Category 4
Azimuth angle	$[-20^{\circ}, 20^{\circ}]$	$[10^{\circ}, 50^{\circ}]$	$[40^{\circ}, 80^{\circ}]$	$[70^{\circ}, 100^{\circ}]$

At the training stage, a set of 2D feature points are manually picked to form a Point Distribute Model (PDM) for each training image. The selected feature points for different post categories are not necessarily related [between](#) different models. However our goal is to build the 3D face model from 2D projected points on different image planes. Therefore, we deliberately choose the 2D feature points for each pose category so that they approximately correspond to the predefined feature points on the generic 3D model.

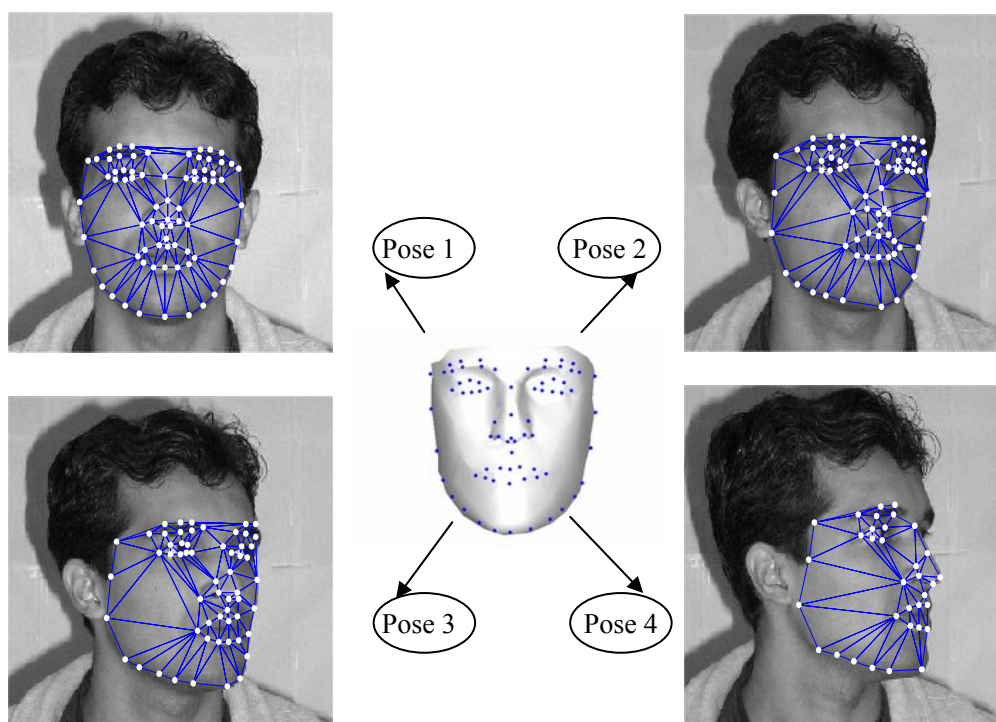
**Figure 4.5 The 3D face model and 4 view-based AAM models**

Fig. 4.5 shows a 3D face model and its face images viewed from 4 different angles. These 4 poses approximately fall into the 4 pose categories defined in Table 4.1. The feature points defined for the 2D view-based AAMs are assumed to be projected from the same set of feature points as marked on the 3D face model. When projected from the 3D space to a 2D plane, some feature points will be inevitably occluded. Apparently only for the pose category 1, all feature points might be possibly visible. For pose category 4, nearly half of the selected feature points are occluded. The set of occluded feature points vary for different face structures and viewpoints. To simplify our model and make it compatible with canonical AAM models, we assume that for the same pose category, the same set of feature points will always be occluded. At a first sight, this assumption violates the projection model for sure. In reality, this assumption not only accelerates the modeling process, but can still meet the accuracy requirement for face alignment and modeling task.

For the view-based AAM defined for images of pose category 2 (or 3), the face contour is represented with the piece-wise lines determined by four extra feature points, which do not explicitly correspond to any predefined 3D feature points as shown in Fig. 4.5. These points are as important in order to completely describe the 3D face shape. It is necessary to generate the face contour that is in accordance with the 3D face model. For this purpose, we define a dense set of 3D candidate points by interpolating some vertices on the 3D generic model. When the generic 3D model is morphed and projected, the projected face contour formed by the convex hull of the projected candidate points are pulled towards the piece-wise face contour in the image. In this way, the 3D face model generates face contours that are as close as possible to those detected from face images. Details about this part will be elaborated later

in this chapter.

4.2.4 Cubic Morphing: Revisited

4.2.4.1 Basic Cubic Polynomial Function

2D morphing technology has been widely used to morph photographic images of one person to another one, or one creature to another kind. The most famous example should be Michael Jackson's music video "Black or White" produced in 1980s, in which different actors are seemingly transformed to one another as they dance. 2D image morphing assumes a set of corresponding point pairs on two images and one image is deformed so that the selected points will be transformed to the points in the other image. Point correspondences between different images need to be established. Some intermediate shapes are necessary with the help of interpolation so that the viewers could see a smooth transformation. 3D morphing techniques adopt a similar idea for different objects in the 3D space. Among a variety of morphing functions, a cubic explicit polynomial function is elected as our morphing function for mainly two reasons. First, the shape difference among different human faces can be adequately captured using a cubic function. Secondly, resorting to higher order functions results in drastic increase in the number of morphing variables. Also more feature points have to be detected and aligned.

Let the i th feature point on the generic model be \mathbf{P}_i^m . It is morphed to point \mathbf{P}_i on a specific face model according to

$$\mathbf{P}_i = \mathbf{M} \cdot \mathbf{G}(\mathbf{P}_i^m) \quad (4-9)$$

Where \mathbf{M} is the morphing matrix, $\mathbf{G}(\mathbf{P}_i^m)$ is a monomial vector that is made up of the

polynomials of the coordinates of $\mathbf{P}_i^m = [x_i, y_i, z_i]^t$. For cubic morphing, $\mathbf{G}(\mathbf{P}_i^m) = [x_i^3 \ y_i^3 \ z_i^3 \ x_i^2 y_i \ x_i^2 z_i \ y_i^2 x_i \ z_i^2 x_i \ y_i^2 z_i \ z_i^2 y_i \ x_i y_i z_i \ x_i^2 y_i^2 \ z_i^2 \ x_i y_i \ x_i z_i \ y_i z_i \ x_i \ y_i \ z_i \ 1]$. Now the projection equation (4-8) can be rewritten as:

$$s_w \cdot \mathbf{R}_w \cdot \mathbf{M} \cdot \mathbf{G}(\mathbf{P}_i^m) + \mathbf{t}_w = \mathbf{p}_i \quad (4-10)$$

For the 3 by 20 morphing matrix \mathbf{M} , it has 60 parameters. Since human face is symmetric in nature, the number of unknown parameters could be reduced with this constraint. The generic face model is symmetric to the y axis of the object coordinate system as shown in Fig. 4.3. For two points (x, y, z) and $(-x, y, z)$ on it, the symmetric constraint assumes they should remain symmetric after being morphed to an individual 3D face. After imposing this symmetry constraint, the number of nonzero elements in the matrix \mathbf{M} is reduced to 33. Considering that the weak perspective projection model has 6 degrees of freedom, with K available face images of a person, the total number of unknown parameters is $(33 + 6K)$.

4.2.4.2 Cubic Morphing Reformulated as a Linear Operation

Equation (4-9) shows how an individual point is morphed. Any morphed point is generated from the cubic polynomial of its corresponding model point on the generic face model. On the other hand, for a fixed set of model points, their morphed points are just linear combinations of the morphing parameters. Denote the number of all selected feature points on the generic model with n_0 . Let Φ^m be the set of all n_0 participating feature points, the set of all morphed points are:

$$\Phi = \mathbf{M} \cdot \mathbf{G}(\Phi^m). \quad (4-11)$$

Bear in mind that $\mathbf{G}(\Phi^m)$ is just a fixed 20 by n_0 matrix. To emphasize the linear relationship between the morphed points and the unknown morphing parameters,

let $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_{33}]^T$ be the 33 parameters, equation (4-11) is reformulated as:

$$\boldsymbol{\Phi} = \mathbf{M}_e \cdot \boldsymbol{\alpha}, \quad (4-12)$$

where \mathbf{M}_e is a $(3n_0)$ by 33 constant matrix formed by rearranging elements in $\mathbf{G}(\boldsymbol{\Phi}^m)$. For the complete feature point set $\boldsymbol{\Phi}^m$, the matrix form of the projection equation (4-10) is then:

$$s_w \cdot \mathbf{R}_e \cdot \mathbf{M}_e \cdot \boldsymbol{\alpha} + \mathbf{T}_e = \boldsymbol{\Omega} \quad (4-13)$$

where \mathbf{R}_e is a $(2n_0)$ by $(3n_0)$ enhanced rotation matrix created by repeating the partial rotation matrix \mathbf{R}_w along its main diagonal. Similarly, the enhanced translation vector \mathbf{T}_e contains n_0 pairs of t_x and t_y . $\boldsymbol{\Omega} = [u_0 \ v_0 \ u_1 \ v_1 \ \dots \ u_{n_0} \ v_{n_0}]^T$ represents the set of all projected feature points. Clearly, the projected points are linear functions of the morphing parameters $\boldsymbol{\alpha}$, the 2D translation parameters t_x , t_y and the scaling factor s_w .

4.2.4.3 Regulate Cubic Morphing Parameters

For a statistical face model like the AAM or the 3DMM, its face shape subspace [models](#) the statistical shape variations of human faces assuming that the shapes of human faces conform to the multivariate Gaussian distribution. Pose parameters and shape parameters are decoupled and well defined to account for pose and shape variations respectively. Instantiating a novel face by editing the shape parameters does not affect the pose of the face. However, the pose parameters and shape parameters are coupled for the cubic morphing operation without proper constraints. Previously we mentioned that the symmetric property of human face is incorporated and there are now 33 degrees of freedom for the cubic morphing operation. However, the morphing operation not only varies the face shape, but also causes it [to be](#) scaled and rotated. Equation (4-13) clearly shows that the pose parameters and morphing parameters are mutually dependent. Therefore, it is necessary to regulate the morphing

operation to avoid a biased estimation of the pose parameters.

We solve this problem by aligning the morphed face to the generic face with a 3D similarity transform \mathbf{Q} . As the morphed model is self-symmetric, its center of gravity always lies in the y-z plane (with the x component being zero, see Fig. 4.4). Then the number of unknown alignment parameters in \mathbf{Q} is reduced from 7 to 4, including the scaling factor s_a , the rotation angle θ_a and the translation pair (o_y, o_z) in the y-z plane. Optimal parameters are obtained by minimizing the sum of squared distance:

$$\arg \min_{s_a, \theta_a, o_y, o_z} \sum_i \left\| \mathbf{Q}(\mathbf{G}(\mathbf{P}_i^m)) - \mathbf{P}_i^m \right\|^2 \quad (4-14)$$

The minimization procedure is given in Appendix C.

The regulation of the morphing operation is conducted after the morphing and pose parameters have been solved using the algorithms introduced in the following sections. Both the morphing parameters and the pose parameters are adjusted accordingly based on the computed regulation parameters. Without the regulation, the shape parameters and the pose parameters would be confused and no longer meaningful.

4.2.5 Distance Map: Revisited

Distance transform was proposed by Rosenfeld and Pfalz in 1968. Over several decades, it has evolved and developed into various algorithms that can handle alignment tasks of multidimensional data sets based on two-dimensional and three-dimensional Euclidean distance functions. Take two-dimensional alignment as an example. Assume that a set of feature points have been extracted from an image. For any point in the image, a distance vector map finds its nearest feature point and measures their distance. Distance transform is

very helpful when the correspondence between two data sets is unknown or partially corrupted. Sometimes two data sets might be [in good correspondence](#) at first, yet after some transform, the original correspondence is no longer valid. Distance transform provides possible solution to this kind of problems.

In [27], distance transform plays an important role for the 3D modeling task. In order to recover the 3D point coordinates from the extracted feature points in the images, point correspondence has to be established across the images. With a distance map serving as a lookup table, explicit point-to-point correspondence is avoided. Instead the distance map automatically maps the projected model points to their nearest feature points extracted from the images. The modeling parameters are updated iteratively to minimize the average distance map. In order for the evaluation of the distance function measured from the distances map to be reliable and practical, the projected model points have to be within a small range of the real feature points. Distance maps in [27] are generated based on an edge detection algorithm. Several crucial feature points from the eye corners, the nose tip and the mouth are manually picked to initialize the model parameters. In fact, without this step, the modeling process in [27] would be much slower if it ever could converge at all.

The view-based AAMs are adopted to align 2D face images of different viewpoints. The 2D model points are assumed to be the projection of the selected 3D feature points. Since the 2D model points are aligned to the face images, there is no manual feature marking involved. With the pre-defined point correspondence, it is straightforward to reconstruct the 3D model points and further compute the optimal model.

However, the pre-defined correspondence between the 3D model points in the 3D generic

model and the 2D view-based AAM model points is only an approximation of the real 3D-to-2D projection. Such a strict point-to-point correspondence is not only over-constrained, but also incorrect. To relax the correspondence problem, distance maps are generated from the 2D aligning results. The modeling process is then refined and evaluated based on the new distance function. Fig. 4.6 shows some examples of distance maps generated from the view-based AAM alignment results for a person in our database.

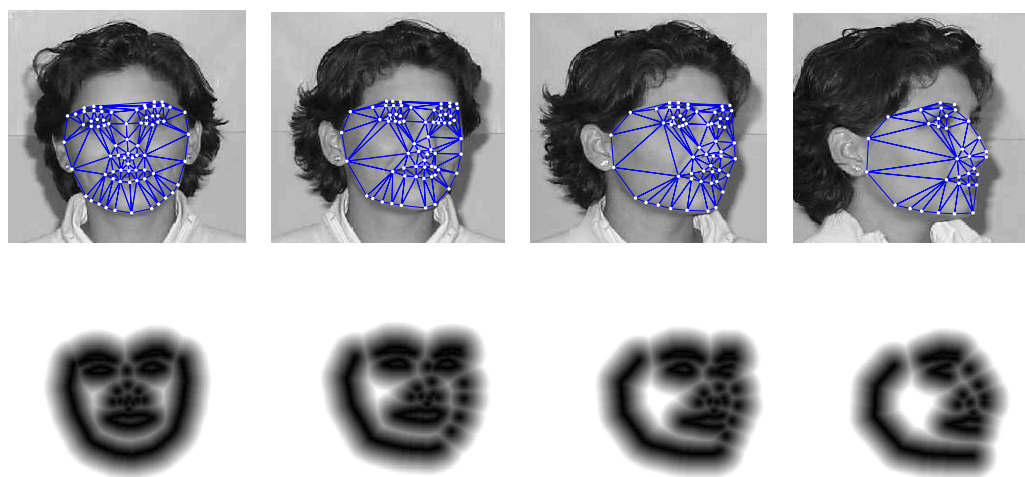


Figure 4.6 Four face images of different viewpoints on the first row. Their corresponding distance maps are shown on the second row.

In Fig. 4.6, the 2D aligned model points are shown as white dots. The triangle meshes show the connectivity of all triangles in the 2D view-based AAMs. Note that for different pose categories, the triangulation schemes are totally different as they are automatically generated by the Delaunay triangulation algorithm based on the pose-specific mean face shapes. In order to generate the distance maps on the second row, model points are first connected to form piece-piece edge maps.

During the modeling process, the model parameters are first estimated with the assumption of explicit correspondence between the 3D model points and the 2D models points. Clearly, it is crucial to successfully extract the 2D feature points with the view-based AAM algorithm. For a person with k_0 face images, failure to accurately align any one of those images would lead to a biased estimate of the 3D face structure. Distance transform is adopted only to refine and improve the 3D modeling result, while in [27], the whole 3D modeling algorithm will not function at all without the extracted distance maps.

4.3 Morphing and Pose Parameter Estimation

4.3.1 Partial Linear Optimization Algorithm

We have made one thing clear that the definition of the 2D model points is in accordance with the selected 3D feature points in the generic model. Equation (4-10) and (4-13) show explicitly how they are related. The view-based AAM algorithms make it possible to extract facial feature points robustly even when the face is partially occluded. Assume for the person to be modeled, k_0 face images of different viewpoints are available. Let $p_{i,k}$ be the extracted i^{th} feature point in the k^{th} face image. The goal is to find the morphing and pose parameters so that the projected feature points are optimally mapped to the extracted feature points in the face images. Based on the projection equation (4-10), the following cost function is derived:

$$CC = \sum_{k=1}^{k_0} \sum_{i=1}^{n_0} \left\| S_k \cdot \mathbf{R}_k \cdot \mathbf{G}(\mathbf{p}_i^m) + \mathbf{t}_k - \mathbf{p}_{i,k} \right\|^2 = \sum_{k=1}^{k_0} \sum_{i=1}^{n_0} \left\| \mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi}) \right\|^2 \quad (4-15)$$

where $\mathbf{\Pi} = (\boldsymbol{\alpha}, \mathbf{\Pi}^* = \{s_k, \theta_{x,k}, \theta_{y,k}, \theta_{z,k}, t_{x,k}, t_{y,k}\}, k = 1, 2, \dots, k_0)$ represents the set of all unknown parameters, including 33 morphing parameters in vector $\boldsymbol{\alpha}$ and $6 \cdot k_0$ pose parameters for k_0 face images. $\mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi})$ is the Euclidean distance vector describing the

difference between the i^{th} projected model point and the corresponding feature point in the k^{th} image. The cost function is simply the sum of squared distances.

The modeling process aims to minimize the cost function in (4-15) with respect to $\mathbf{\Pi}$.

According to (4-10) and (4-13), the projected model points are linear functions of the scaling factor, the translations and the morphing parameters. Apparently, the cost function in (4-15)

can be minimized by setting the partial derivatives with respect to parameters in $\mathbf{\Pi}$ to zeroes.

Let $\mathbf{q}_{i,k} = \mathbf{R}_k \cdot \mathbf{G}(\mathbf{p}_i^m)$ denote the i^{th} morphed and rotated model point. Given the current estimation of the model parameters in $\mathbf{\Pi}$, the partial linear solution to s_k is:

$$\mathbf{s}_k = \frac{\sum_{i=1}^{n_0} \mathbf{q}_{i,k}^T (\mathbf{p}_{i,k} - \mathbf{t}_k)}{\sum_{i=1}^{n_0} (\mathbf{q}_{i,k}^T \cdot \mathbf{q}_{i,k})} \quad (4-16)$$

Similarly, the translation vector \mathbf{t}_k should be updated as:

$$\mathbf{t}_k = \frac{1}{n_0} \sum_{i=1}^{n_0} (\mathbf{p}_{i,k} - s_k \cdot \mathbf{q}_{i,k}) \quad (4-17)$$

The solution to \mathbf{R}_k is a little bit trickier since its two rows must be orthogonal unit vectors for

\mathbf{R}_k to be a valid rotation matrix. Instead of updating \mathbf{R}_k directly, or the three underlying

rotation angles $\theta_{x,k}, \theta_{y,k}, \theta_{z,k}$, we replace \mathbf{R}_k in (4-15) with $\Delta\mathbf{R} \cdot \mathbf{R}_k$, where $\Delta\mathbf{R}$ is a 2 by 2

matrix. For $\Delta\mathbf{R} \cdot \mathbf{R}_k$ to be a valid rotation matrix, it is not difficult to find out that $\Delta\mathbf{R}$ has to

be a rotation matrix in 2D, which indicates that we only need to estimate the rotation angle to

minimize the average distance [between](#) two set of 2D points. The problem is simplified to the

solution of $\Delta\mathbf{R} \cdot [\mathbf{U}_0 \quad \mathbf{V}_0]^T = [\mathbf{U}_1 \quad \mathbf{V}_1]^T$, where $\mathbf{U}_0, \mathbf{V}_0, \mathbf{U}_1, \mathbf{V}_1$ are respectively the x

components and y components of the source and destination set of points (in the form of

column vectors). The rotation angle θ in $\Delta\mathbf{R}$ has the optimal solution as:

$$\theta = \arctan\left(\frac{\mathbf{U}_0^T \cdot \mathbf{V}_1 - \mathbf{V}_0^T \cdot \mathbf{U}_1}{\mathbf{U}_0^T \cdot \mathbf{U}_1 + \mathbf{V}_0^T \cdot \mathbf{V}_1}\right) \quad (4-18)$$

Though in the above equations, all model points are considered, in reality, some of them are not visible due to occlusion. It is our assumption that the same pre-defined points will be occluded for a specific pose category. Therefore those points are excluded from the [estimation](#) the model points.

After the pose parameters are updated in turns according to equation (4-16) to (4-18), the morphing parameters are updated based on the linear equation (4-13). First, the cost function in (4-15) is rewritten as an explicit function of morphing parameters \mathbf{a} .

$$e(\mathbf{\Pi}) = \sum_{k=1}^{k_0} \left\| s_k \cdot \mathbf{R}_{e,k} \cdot \mathbf{M}_e \cdot \mathbf{a} + \mathbf{T}_{e,k} - \mathbf{\Omega}_k \right\|^2 \quad (4-19)$$

The morphing parameters \mathbf{a} that minimize equation (4-19) is:

$$\mathbf{a} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b} \quad (4-20)$$

where

$$\mathbf{A} = \sum_{k=1}^{k_0} (s_k \cdot \mathbf{R}_{e,k} \cdot \mathbf{M}_e) \quad \text{and} \quad \mathbf{b} = \sum_{k=1}^{k_0} (\mathbf{\Omega}_k - \mathbf{T}_{e,k}) \quad (4-21)$$

Basically, the partial linear optimization procedure finds the pose and morphing parameters by iteratively updating them using (4-16), (4-17), (4-18) and (4-20). The order of updating is carefully arranged so that the estimation is stable and fast.

4.3.2 Optimization with Marquardt-Levenberg algorithm

In the previous section, the cost function in (4-19) (or equivalently (4-15)) is minimized using the partial linear optimization algorithm. The algorithm is stable. However, since the parameters are updated one by one, it is of low efficiency compared to gradient descent based

algorithms. Gradient descent based algorithms make use of the explicit partial derivatives of the cost function. For a multivariable cost function like (4-19), the parameter updating rule is dictated by the slope of the cost function at current estimate in the parameter space. For example, the simplest steepest descent algorithm updates along the negative direction of its slope. The partial derivatives of the cost function (4-15) with regard to the model parameters can be expressed using the gradients of the Euclidean distance vectors. The cost function (4-15) is repeated here for better understanding.

$$e(\mathbf{\Pi}) = \sum_{k=1}^{k_0} \sum_{n=1}^{n_0} \|\mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi})\|^2$$

Assume the initial estimate of model parameters are $\mathbf{\Pi}_0$. At the j^{th} iteration, the cost function is approximated by its first order Taylor series over $\mathbf{\Pi}$ around the point $\mathbf{\Pi}_0$ as follows:

$$e_j(\mathbf{\Pi}) = e(\mathbf{\Pi}_j) + \Delta^T \cdot \mathbf{D} + \frac{1}{2} \Delta^T \cdot \mathbf{H} \cdot \Delta \quad (4-22)$$

where \mathbf{D} is the exact gradient of $e(\mathbf{\Pi})$ and \mathbf{H} is the approximation to the Hessian matrix.

$$\mathbf{D}_l = 2 \sum_{k=1}^{k_0} \sum_{i=1}^{n_0} (\mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi}_j) \cdot \frac{\partial \mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi}_j)}{\pi_l}) \quad (4-23)$$

$$\mathbf{H}_{l,m} = 2 \sum_{k=1}^{k_0} \sum_{i=1}^{n_0} \left(\frac{\partial \mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi}_j)}{\pi_l} \cdot \frac{\partial \mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi}_j)}{\pi_m} \right) \quad (4-24)$$

In the Marquardt-Levenberg (LM) optimization algorithm [62], the incremental update Δ for the model parameters is obtained by solving

$$(\mathbf{H} + \lambda \cdot \mathbf{I}) \cdot \Delta + \mathbf{D} = 0 \quad (4-25)$$

where λ is a small positive scalar. When λ is zero, the direction of Δ is identical to that of the Gaussian-Newton method. When λ tends to infinity, Δ will be a very small vector pointing the steepest descent direction. Therefore, the LM method uses a search direction that balances the Gauss-Newton direction and the steepest descent direction. Usually in the

iterative procedure, λ starts with a pre-defined small number and it decreases as the modeling error becomes smaller.

For both the partial linear optimization method and the LM optimization method, it is assumed that the projected model points will always be visible regardless of the real viewing angle. Since for a fixed pose category, the same occlusion is assumed. Those occluded feature points are excluded from the equations in (4-15) to (4-25).

4.3.3 Incorporate Contour Constraints to the Optimization

The goal of this chapter is to reconstruct 3D face structures from face images. The reconstruction is approached by a 3D face modeling algorithm that relies on the extracted feature points from images.

However, as mentioned in the view-based AAM section, the 2D feature points describing the face contour do not explicitly correspond to a set of 3D feature points in the generic 3D face model. Fig. 4.7 shows the mean face meshes in pose category 2 and 3 for the view-based AAM algorithm, where the extra 4 points are circled out with dashed ellipses.

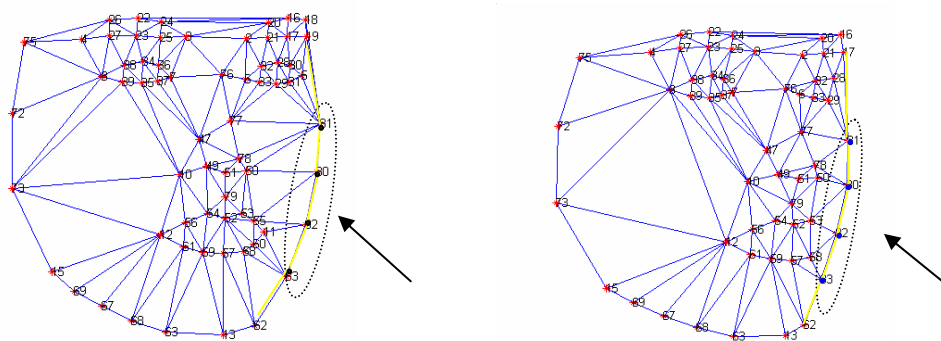


Figure 4.7 Face meshes used in the view-based AAM: pose category 2 and 3 from left to right.

Naturally, it is desirable that the reconstructed 3D face generates the same face contour as the extracted face one from the face image. For a generic face mesh, it is not difficult to determine the projected face contour given the morphing and pose parameters. Usually, the contour is generated as the convex hull of all projected points. However, it is not trivial to measure the distance of two arbitrary curves and adjust the model parameters to minimize the distance accordingly. Since the extracted face contour is spanned by 4 points, it is enough to analyze the piece-wise curve defined by 6 points (including two extra neighbors). Here, we aim to add extra constraints from the face contours in yellow color in Fig. 4.7 to the modeling process.

To incorporate the contour constraint to the modeling process with a minimum extra computation, we simplify the curve-to-curve matching problem to an average point-to-curve distance problem. The identities of those 3D model points that might contribute to a projected face contour are unknown. On the generic face model, we manually pick 48 points as a pool of candidates and denote the set with Ψ . They are believed to be all possible points that might constitute the face contour after the projection. Fig. 9 shows those candidate points on the generic face model as red stars.

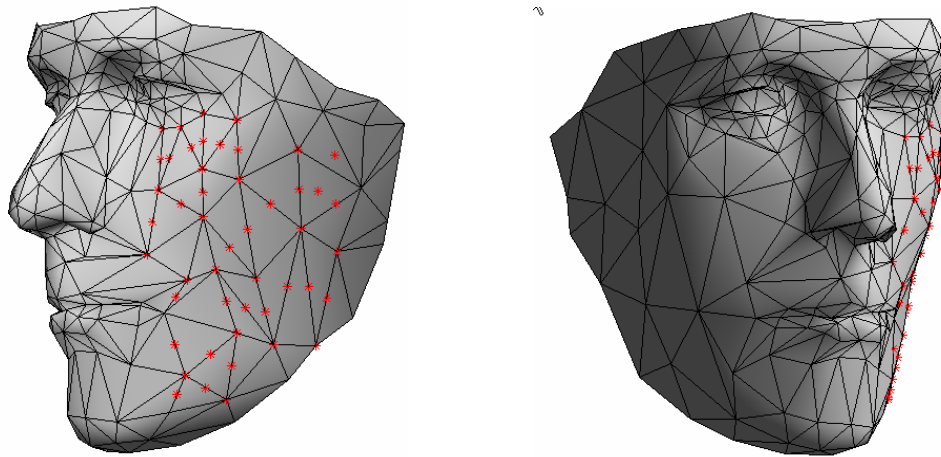


Figure 4.8 Candidates for contour points on the generic model viewed from two different angles

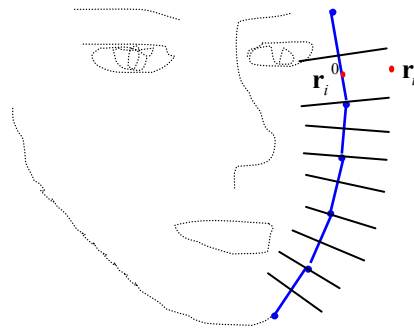


Figure 4.9 Nine normal lines on the face contour

Fig. 4.9 shows an extracted face contour from a face image in pose category 2 (or 3). 9 lines normal to the face contour are plotted. They pass through 4 vertices plus 5 in-between points.

Therefore the face contour area is divided into 10 neighborhoods. Neglect the one on the top,

the model points in Ψ will fall into one of the 9 neighborhoods. $\Psi = \{\Psi_i, i = 1, 2, \dots, 9\}$. For

the projected points inside the Ψ_i neighborhood, the one that is furthest to the right of the

contour line is detected. Let this point be \mathbf{r}_i and its projection to the contour line be \mathbf{r}_i^0 . In



order to minimize the distance between the projected face contour and the extracted face

contour, it is equivalent to minimize the sum of squared distances for all night pairs of points.

The cost function is:

$$e_1(\mathbf{\Pi}) = \sum_{i=1}^9 \|\mathbf{r}_i - \mathbf{r}_i^0\|^2 \quad (4-26)$$

Since the outline of the 9 rightmost points reflects the projected face contour, minimization of (4-26) is equivalent to the matching of the projected face contour and the extracted one from the face image.

The overall cost function is then revised as the addition of $e_1(\mathbf{\Pi})$ and the one defined in (4-15).

$$e(\mathbf{\Pi}) = \sum_{k=1}^{k_0} \sum_{n=1}^{n_0} \|\mathbf{d}(\mathbf{p}_{i,k}, \mathbf{\Pi})\|^2 + e_1(\mathbf{\Pi}) \quad (4-27)$$

The same algorithm described in section 4.3.1 and 4.3.2 is used to minimize this new cost function as it is still a sum of squared distance. However, the identities of the 9 rightmost projected points have to be dynamically updated in the iterative optimization procedure.

4.3.4 Refine the Parameter Estimation with Distance Mapping

In section 4.2.5, we have reviewed the basic idea of using distance maps to improve the modeling accuracy. For an object that is as complicated as a human face, a distance map can not correctly reflect the distance of the source and destination point sets when the model parameters are far from optimal. Therefore, it is of low efficiency to make use of a distance map for the entire optimization procedure.

Fortunately, with the methods in section 4.3.1 and 4.3.2, we are able to have a good estimation of the model parameters. Distance transform is used to further refine the modeling result and relax the strict correspondence between the 3D and 2D points.

It is very easy to incorporate the distance map to either the partial linear optimization algorithm, or the LM algorithm. The only difference is the corresponding 2D points are not fixed in the iterative procedure. Instead, they are looked up from distance maps dynamically.

4.4 Experiments

4.4.1 View-based AAM

We use a face database of 278 face images for our experiments. It is based on a small database of 32 people previously collected in Drexel [27] and expanded with some available face images from public databases on the Internet. Table 4.2 shows some information of the database.

Table 4.2 Our face database

Pose	Category 1	Category 2	Category 3	Category 4
	$[-20^{\circ}, 20^{\circ}]$	$[10^{\circ}, 50^{\circ}]$	$[40^{\circ}, 80^{\circ}]$	$[70^{\circ}, 100^{\circ}]$
Number of img	83	85	47	63

Sine this database is quite small, we would like to use all face images for the training of face subspaces for the 4 view-based AAM methods designing for the 4 pose categories. Practically, a leave-one-out strategy is adopted. That is, for N images, N-1 images are used for training the AAM, while the left-out is subject to testing. The 2D AAM-based alignment algorithm has been discussed a lot in previous chapters. Fig. 4.10 shows the average faces for the 4 view-based AAMs respectively.

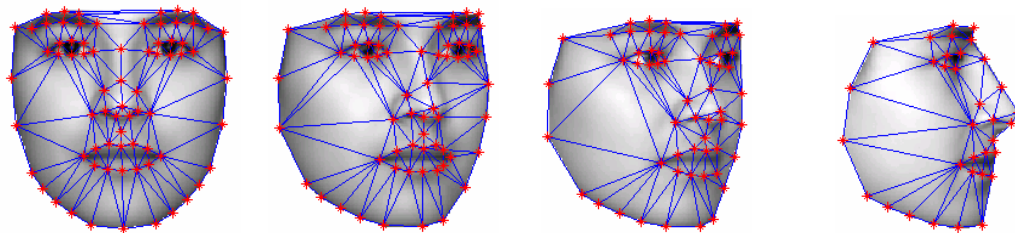


Figure 4.10 Average faces for pose category 1 to 4 (from left to right).

Note the face meshes are generated using the Delaunay triangulation algorithm. The triangulations are different for different pose categories as shown in Fig. 4.10. In Appendix D, the shape variations caused by editing the principal shape mode are demonstrated. Also Appendix E shows a table of face images that are generated by varying only the principal texture mode of the face texture subspace.

The alignment of a 2D face image using the view-based AAM is similar to the canonical AAM algorithm. First, the pose of the face is roughly determined in order to choose the right AAM. The average model parameters serve as the initial parameters as being plotted in the first row in Fig. 4.11. The second row in Fig. 4.11 shows the converged faces for the test images that are from 4 different pose categories.

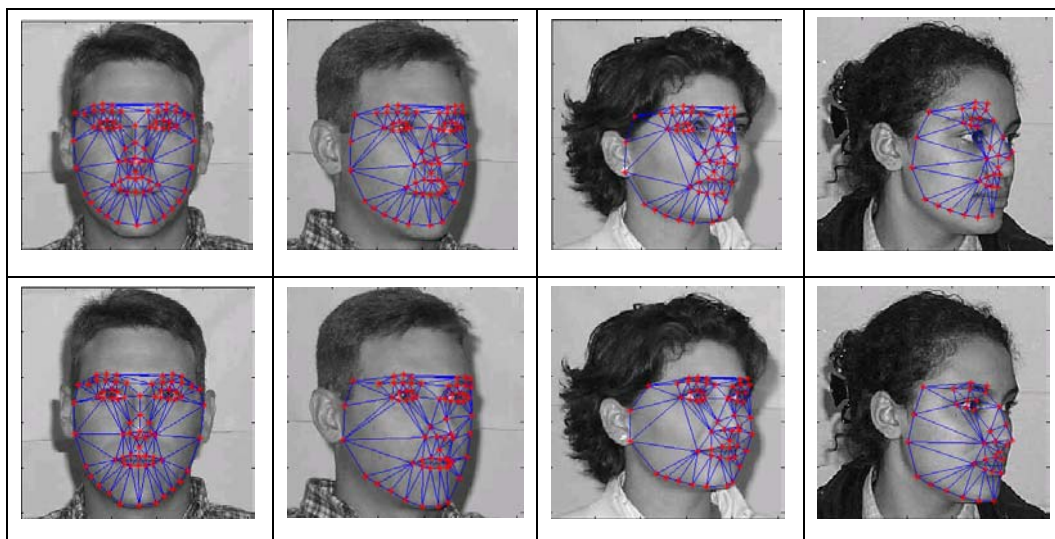


Figure 4.11 Face alignment results of different poses

4.4.2 3D Modeling Experiments

4.4.2.1 Partial Linear Optimization versus LM Optimization Algorithms

For both the partial linear optimization algorithm and the LM optimization algorithm, the same initial settings and stopping criterion are used. Fig. 4.12 compares the reconstructed 3D faces for the person in Fig. 4.6 resulting from the partial linear optimization algorithm (on the first column), the LM algorithm (on the second column) and the regularized LM algorithm (on the 3rd column). The top row shows the frontal views. On the second row, the 3D faces are rotated by -20 degree around the vertical axis. The profile views are shown on the last row.

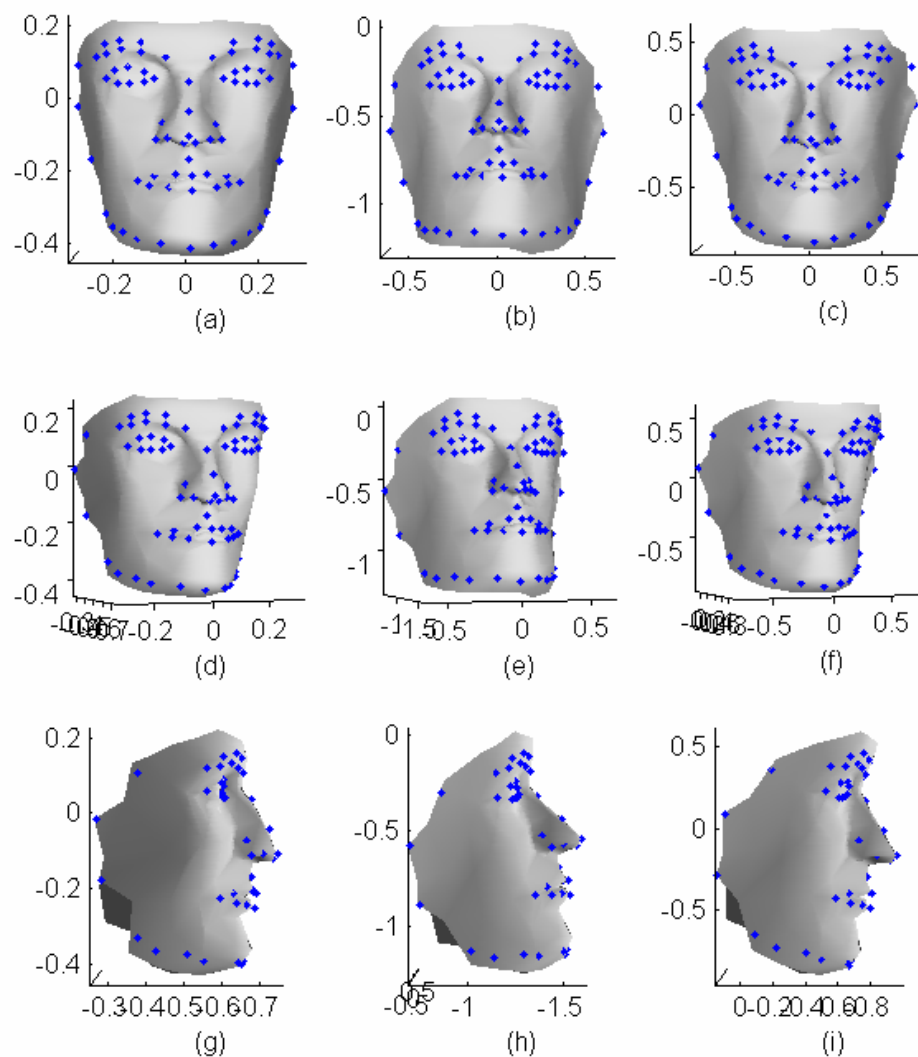


Figure 4.12 Modeling results: (a)(d)(g): the partial linear method shown as frontal, half-profile and full profile; (b)(e)(h) the LM method shown from three angles; (c)(f)(i): the regularized LM method from three angles.

As expected, all recovered 3D faces have steep and unconstrained cheeks, which is reasonable since the morphing of the generic model is only dictated by the feature points in blue. For the LM algorithm, the morphing operation also causes the 3D face shifted, scaled and rotated. Without any regulation, the morphing parameters are coupled with the pose parameters. On the other hand, the partial linear method seems less affected. After regularization of the

morphing parameters, the reconstructed 3D face is normalized with the generic 3D face.

Table 4.3 compares the average modeling errors of these two algorithms by evaluating the cost function (4-15) for the first 5 steps in the iterative optimization procedure.

Table 4.3 Average modeling errors for different methods

Step	1	2	3	4	5
Partial	80.8078	19.6619	13.7256	12.7508	12.5186
Linear					
LM Method	1059.6	21.7412	4.3867	4.1611	4.1278

Clearly, both algorithms converge quickly and reach their stable modeling errors within 4 steps. After the first iteration, the partial linear optimization algorithm has a better performance (compared to the number 1059.6 for the LM algorithm). However, the overall estimation using the LM algorithm is more accurate. In fact, even after 30 iterations, the partial linear algorithm still has an average modeling error around 12.1. Apparently the LM algorithm is a better choice in terms of the modeling accuracy. In our MATLAB implementation, an iteration of the LM algorithm takes about 5 ms, which is twice the time needed for one step in the partial linear algorithm. For further analysis of the incorporation of the contour constraints and the distance maps, only the LM algorithm is considered.

4.4.2.2 Incorporation of Face Contour Constraints

The 3D face looks very different without the constraint of face contours, though only 9 extra contour points are considered. Fig. 4.13 compares the modeling results with or without the

face contour constraint. The morphed face model together with the selected feature points are plotted on the first column. Their 3D projections, as well as the generated face contour, are overlapped on the face image on the second column. If we compare the results on the first row (without contour constraint) with the results on the second row (with the face contour constraint), the difference is drastic. Clearly the contour constraint is very crucial in order to interpret and model the face accurately.

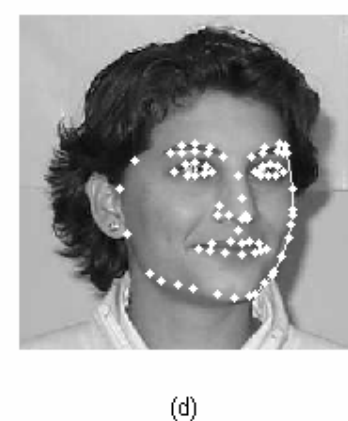
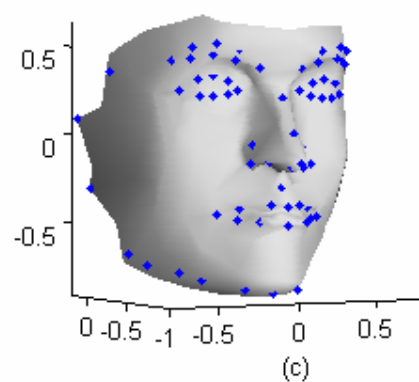
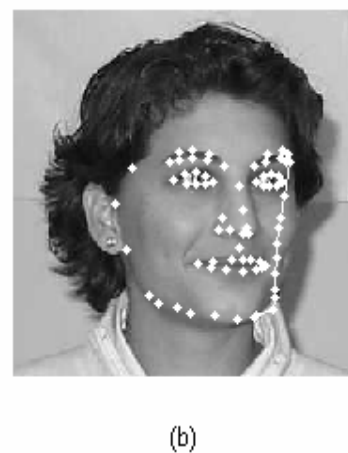
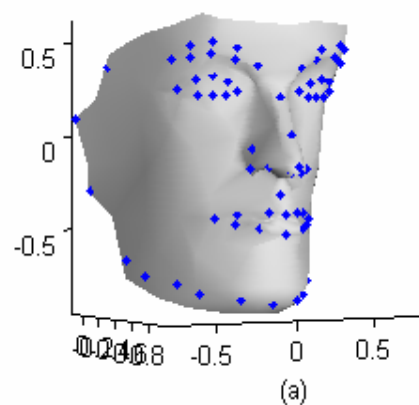


Figure 4.13 Modeling results with and without the contour constraint for pose category 2

Fig. 4.14 shows similar results except the pose of the face is classified to the pose category 3.

Both Fig. 4.13 and Fig. 4.14 explicitly demonstrate the importance of the face contour in the reconstruction process of a realistic 3D face.

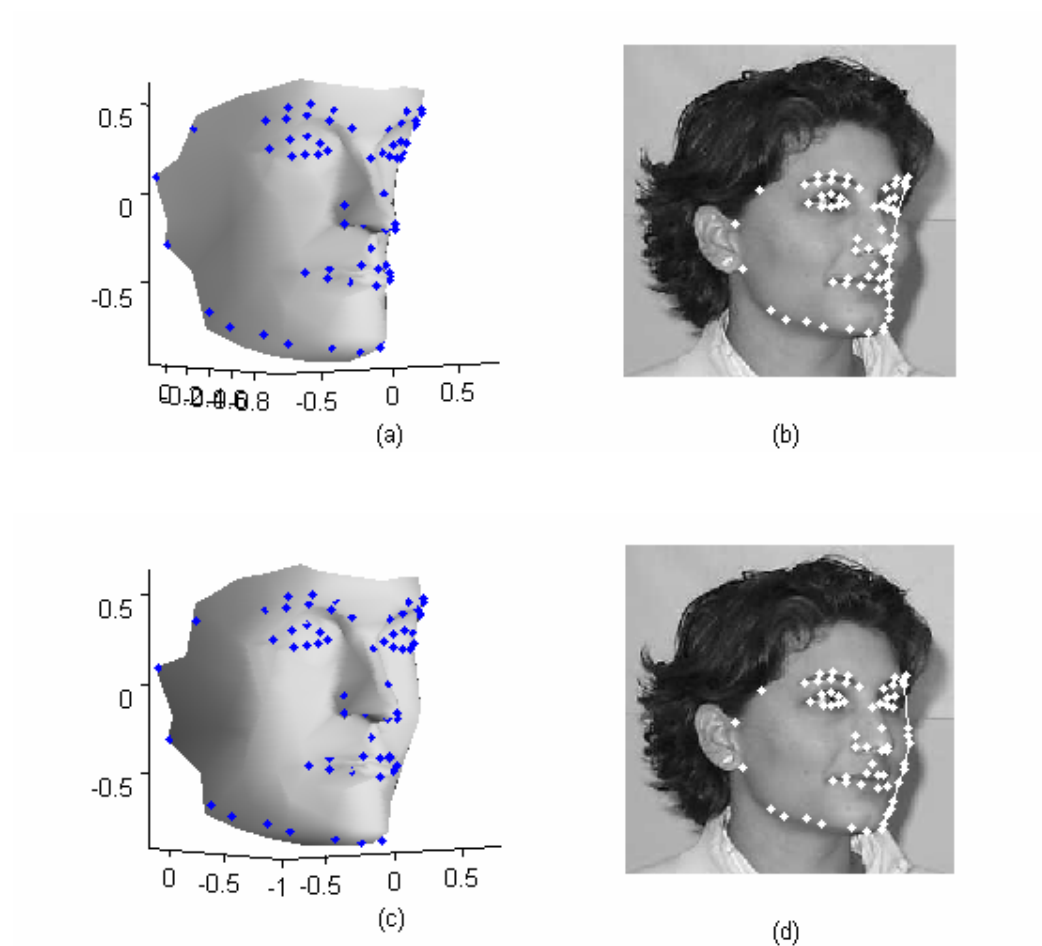


Figure 4.14 Modeling results with and without the contour constraint for pose category 3

4.5 Conclusions

In this chapter, we present an algorithm to reconstruct the 3D face structure from 2D face images. The reconstruction is conducted by morphing a 3D generic face model with a cubic

polynomial, which in essence is a linear function of the model parameters. The view-based AAMs are employed to extract 2D feature points. Face contours are extracted from images and combined into the modeling procedure as constraints. As a result, the reconstructed 3D face looks more realistic and natural. While the extraction of the 2D feature points with the view-based AAM is robust and fast, the 3D modeling task is straightforward and takes less than a second to finish in a Pentium IV computer.

Reconstructed 3D face structures can be applied for a variety of tasks like face recognition, pose determination and synthesis of novel views. There is only one thing left. For a specific face, a texture descriptor is needed to characterize its unique intensity pattern. Usually, this is approached by using texture mapping techniques. However, a more powerful approach is to explicitly model the formation of image irradiances and the environmental illumination condition. This is the topic of our next chapter.

CHAPTER 5 : FROM 2D TO 3D: ILLUMINATION-FREE TEXTURE

EXTRACTION WITH THE SPHERICAL HARMONIC

ILLUMINATION MODEL

Any human face, as a 3D object, has a unique surface geometry that is determined by the underlying bone structure and the associated facial muscles. Just like the surface shape, the surface appearance also serves to distinguish different individuals. A face is perceived by human eyes or cameras as an image of unique intensity pattern due to different surface albedo at each surface point. Surface albedo is an inherent property of the specific material. However, the appearance of a face not only depends on the face surface albedo, face shape and the viewing angle, but also varies significantly under different illumination conditions. In fact, the intra-personal appearance difference caused by solely varying the illumination condition could be much larger than the inter-personal difference. This remains one of the main challenges for automatic face recognition algorithms. It is of great interest to model the inverse imaging process and extract illumination invariant texture (or surface albedo) information. In this chapter, a latest analytical illumination model is adopted which is developed based upon the Spherical Harmonic representations of the illumination function and the Lambertian reflectance function. The illumination in a face image is analyzed after the 3D face shape is extracted using the algorithm introduced in last chapter. A unique illumination-free texture map is generated afterwards. This global texture map, together with the extracted face structure, completely describes the face. Several simple applications are

demonstrated.

5.1 Literature

The analysis and synthesis of the images of an illuminated 3D object is an important topic in computer graphics community and computer vision community. For computer graphics people, it is of great interest to mimic the image forming process, that is, to generate images given known surface shape, albedo and illumination conditions. The rendered images from 3D parametric objects are desired to be as realistic as possible. Computer graphics techniques are nowadays widely applied to make animated films in Hollywood. Researchers in the computer vision community, on the other hand, are more interested in the inverse procedure, which is to recover 3D objects and their surrounding environmental illumination conditions in the physical world by analyzing face images or videos. The two communities respectively emphasize different aspects of the same phenomenon. Research progresses in the two areas are closely related and mutually boosted.

5.1.1 When Illumination is not considered: Face Texture Mapping Techniques

Assume for an ideal scenario, the irradiance at an image point is proportional to the surface albedo of the corresponding 3D surface point. The appearance of the face is said to be illumination free. Novel images of that face can be synthesized using existing example face images. This is a typical texture mapping (or texture rendering) problem. By definition, texture mapping is a technique to map 2D images onto 3D surfaces so that the transformed color data conforms to the surface plot. Specifically, a texture color is computed for each point on a face model from a collection of photographs, with the knowledge of the face

structure and the pose parameters of all photographs. The main task of texture mapping is to establish correspondence across photographs of different viewing angles and blend different values to yield best estimation for a novel image.

There are two different texture mapping approaches. The first one is view-independent texture mapping, which results in a texture map that can be used to render the target face from any viewpoint. View-independent texture mapping generates one and only one texture map for each individual. A texture map is usually defined on a 2D texture space, and the most popular projection model to map the 3D coordinates of an object to the 2D texture space is the cylindrical projection model [63]. In the 2D texture space, any texture unit is computed as a weighted sum of the texture values at the corresponding image points across all photographs. When synthesizing a novel view of the face object, image intensities (or textures) are easy to generate given the explicit mapping between the image coordinate system, the 3D world coordinate system and the texture space.

Another approach is view-dependent texture mapping. Depending on different viewpoints, different blending weights are applied to the available photographs. It starts with choosing a subset (or all) of the available photographs, followed by determining a blending weight for each of these photographs. Pulli et al [64] selected three photographs based on a Delaunay triangulation of a sphere surrounding the object. When azimuthal angle is the dominating factor among different viewpoints, it is enough to choose two photographs whose viewing angles are the closest to the desired new image.

Since view-dependent texture mapping synthesizes a new image directly from existing images of closest viewpoints, usually the resulting images look more realistic compared to those

using view-independent texture mapping techniques. For view-independent texture mapping, a 2D cylindrical texture map might not be able to cover the whole surface of the object in the 3D space due to possible self-occlusion in the mapping. However, view-independent texture mapping has its own advantages. As a global texture map is a weighted sum of images of different viewpoints, view-independent mapping is less sensitive to any variations in exposure or lighting conditions in the original photographs. It also requires less memory since only one global texture map is generated and saved.

5.1.2 Basic Illumination Models for Photorealistic Rendering in Computer Graphics

This world is full of all kinds of colorful objects of different materials. Modeling the colors and lighting effects is a complex process. Internally, due to the interaction of electromagnetic energy with the object surface, the amount of reflection and absorption of the incident light varies. When the reflected light reaches human eyes, it triggers perception processes that yield the image of the scene as we perceive. Besides different color effects, objects show some other illumination effects. They might look opaque or transparent to some extent. Their surfaces might be shiny or dull etc. An illumination model [65], in computer graphics, is a term for the process to model and calculate the intensity radiating from a particular surface point to the image plane, given the surrounding lighting environment (a number of light sources of varying shapes, colors and positions), the geometry in the scene (relative positions of light sources, surfaces and the camera), as well as the surface properties of the presented objects.

5.1.2.1 Light Sources

Light sources, refer to those light-emitting sources like light bulbs or the sun, or the natural sky light etc. Some objects, like a mirror or a shiny wall, might serve as indirect sources for lighting nearby objects. They are not considered as light sources. Instead they are only light-reflecting sources. The simplest light source is a point source which evenly emits energy in all directions from one location. As radiant energy from a point light source travels through space, its amplitude is attenuated by the reciprocal of the squared distance it has traveled. Therefore when a point light source is close enough to an object, both its direction and amplitude have to be considered for a specific surface point. Another common light source is a distributed light source, or a directional source. A directional source usually is featured with constant amplitude and one lighting direction. A point source can be approximated as a distributed light source when it is far away enough from the object surface. A typical example is the sun. There are some special light sources like the X-ray, structured light, laser light etc. Images formed in those ways have special applications and are beyond the topic here.

5.1.2.2 Basic Illumination Models

Illumination models are often derived from physical laws. Most models are empirical models that are based on simplified photometric calculations. Here several basic illumination models are reviewed. For most objects in the world, overall surface intensities are the results of joint contribution from different types of reflections.

The first contribution is from ambient light, or background light. It is a uniform illumination without spatial or directional characteristics. It casts same amount of light on all objects. This

happens when no object in the scene is exposed directly to a light source. Ambient light is usually related with the general level of brightness of an image.

The second contribution comes from the diffuse reflection. For rough or grainy surfaces, they tend to scatter the reflected light in all directions so that the surface appears equally bright from all viewing angles. An ideal surface of this type is referred as Lambertian reflector since the reflection function is governed by Lambert's cosine law, which states that the reflected energy is proportional to the incident light intensity, surface albedo, as well as the cosine of the angle between the negative incident direction and the surface normal. The cosine function implies that for the same amount of incident energy, only the perpendicular equivalent area should be considered. The color of the surface is actually the color of the diffuse reflection of the incident light.

Besides ambient light and diffuse reflection, sometimes we see highlights or bright spots on a surface that vary under different viewing directions. That is caused by another type of reflection called specular reflection. It often happens on shiny surfaces. The incident energy is reflected in a concentrated region around the specular-reflection angle. When the viewing direction coincides with that reflection angle, the specular phenomenon is most prominent.

Given one single point light source, the intensity on a surface point is a combination of different reflections [according to the Phong illumination model](#):

$$I = k_a I_a + k_d I_l (\mathbf{n} \cdot \mathbf{l}) + k_s I_l (\mathbf{n} \cdot \mathbf{h})^{n_s} \quad (5-1)$$

where the first item is the ambient component with k_a being the ambient reflection coefficient and I_a being the intensity of the ambient light. The second and third items are respectively the diffuse and specular components. The diffuse intensity is proportional to the

inner product of the surface normal \mathbf{n} and the incident lighting direction \mathbf{l} . \mathbf{h} is the halfway vector between \mathbf{l} and the viewing vector \mathbf{v} . When it coincides with the surface normal, the specular intensity is maximized. n_s is the specular-reflection parameter that varies between 0 and 1. k_d and k_s are diffuse and specular reflection coefficients respectively.

When rendering an image of wire-modeled 3D objects in a scene, consideration for the above illumination models is a must, yet it is far from complete. First of all, contributions from individual light sources have to be summed. Secondly, if transparent objects are present, transparency effects have to be modeled by considering the light refraction. When an object surface is not directly lit by a light source, there will be shadows in the scene. There are mainly two different types of shadows. Attached shadow is produced when a surface normal is facing away from the lighting direction. In equation (5-1), the dot product for the second item would be negative. Practically it is just set to zero for negative values. On the other hand, even when the dot product is positive, that surface point might be occluded by another object. A shadow generated in this way is called cast shadow. A commonly adopted method to determine the shadow areas is the hidden surface algorithm. Finally, a ray of light might be transmitted and reflected several times before it reaches the image plane. A realistic image rendering algorithm should consider global reflection and transmission effects. One of the successful methods is ray tracing, where a ray is sent out from each pixel position and allowed to bounce around the scene to obtain global lighting effects.

5.1.3 Illumination Modeling for Face Recognition

Modeling illumination effects in computer graphics and computer vision are two different

concepts with different goals. Basic illumination models are reviewed in the previous section. Those models are fundamental for one to understand the basic illumination process. Modeling illumination in face recognition deals with illuminated human faces in images or video sequences. Usually the ultimate goal for illumination analysis is to extract illumination free features for recognition purpose, or synthesize novel face images under arbitrary illumination conditions. Illumination models, from this point on, refer to methods to describe and characterize a low-dimensional illumination subspace for each face, given a number of its example images. In another word, what's the set of all images of a face object under all possible illumination conditions?

Though generally, both diffuse and specular reflections co-exist. Simple models are adequate for face analysis and recognition. For most existing studies of illumination analysis, it is assumed a convex Lambertian object is illuminated by distant light sources. By distant light sources, it is justified to assume that a light shines on each point in the scene from the same angle and with the same intensity. Secondly, the object is Lambertian so that the specular effect is not considered. In equation (5-1), the second item is the Lambertian reflection contribution. For a single [light source pointing at direction \$\mathbf{u}_l\$](#) with light intensity $l(\mathbf{u}_l)$, according to Lambert's law, image irradiance at the i th surface point is:

$$\mathbf{I}_i = l(\mathbf{u}_l) \rho_i \max(\mathbf{u}_l \cdot \mathbf{v}_i, 0), \quad (5-2)$$

where vectors \mathbf{v}_i is the surface normal vector and ρ_i is the surface albedo that describes the fraction of the light reflected at this point. Clearly, for Lambertian objects, all we need to know are surface normal and surface albedo at each surface point. Finally, the object is assumed to be convex so there is no cast shadows. Attached shadows are considered in

equation (5-2). All above assumptions lead to much simpler and efficient algorithms.

5.1.3.1 PCA-based Low-dimensional Linear Subspace Representation

If raw image data of a face image is scanned line by line to form a vector, it can be considered as one point in a high-dimensional image space. Given enough sample points (example images), their distribution and other properties can be studied. Based on their observation and experiments, researchers believe that the set of all images of a specific face under all possible illumination conditions lie in a low-dimensional linear subspace. Hallinan [65] studied the set of images of a face viewed from a fixed viewing angle while illuminated by a floodlight placed in various positions. With Principle Component Analysis (PCA), he found that five or six principal components are enough to characterize the illumination subspace. In [66], Epstein et al. conducted experiments on various Lambertian objects and concluded that images of a Lambertian object can be approximately with a linear illumination space of dimension that is between three and seven.

Since PCA is a statistical tool, it usually requires a training database that is general enough to include most of the variations for the population in the real world. Without enough samples, the resulted models are somehow biased and not representative. Another drawback for PCA based illumination models is related with different viewpoints. There is no effective way to handle the pose problem. The pose space is usually sampled and a linear subspace is generated for each pose category as being done in [67].

5.1.3.2 Illumination Modeling based on Theoretical Analysis of Lambertian Reflectance

The PCA-based analysis shows empirically that the illumination space has very low

dimensions. This conclusion could also find its theoretical foundation from the analysis of the reflectance function, assuming we are dealing with Lambertian objects. Without any consideration for cast and attached shadows, let $\tilde{\mathbf{v}}_i = \mathbf{v}_i \rho_i$ be the albedo-weighted normal vector at the i th surface point. Its image irradiance is then the inner product of the weighted normal and the lighting vector, $\mathbf{I}_i = \tilde{\mathbf{v}}_i \cdot \mathbf{u}$. Rearrange it in matrix form as $\mathbf{I}_i = \tilde{\mathbf{v}}_i^T \mathbf{u}$. Let \mathbf{V} be a matrix whose rows are the weighted normal vectors for all surface points. $\mathbf{V} = [\tilde{\mathbf{v}}_1^T; \tilde{\mathbf{v}}_2^T; \dots; \tilde{\mathbf{v}}_{n_0}^T]$. The image is represented as $\mathbf{I} = \mathbf{V}\mathbf{u}$. When multiple light sources are present, just sum them up and the resulted image is $\mathbf{I} = \mathbf{V} \sum_m \mathbf{u}_m$. Clearly, the image has 3 degrees of freedom since \mathbf{V} is a matrix of n_0 by 3. It indicates that without any consideration for shadows, all images of a Lambertian object lie in a three-dimensional space. When the ambient component in equation (5-1) is also considered, all set of images (no shadows) lie in a four-dimensional subspace.

In reality, a face might be presented in an image with heavy shadows. Linear subspace methods based on the above analysis (or PCA analysis) have poor performance as they can not extrapolate novel images at unseen poses and under new illumination conditions. Illumination cone is an extrapolative illumination model proposed by Belhumeur and Kriegman [50]. First of all, they pointed out that the set of images of an object under arbitrary illumination form a convex cone in image space. It is convex since any positive linear combination of two images is still a valid image. It is also a cone as a scaled image is still within the same illumination space. Secondly, for a specific face, the convex cone can be learned from as less as three images. To construct the convex cone, Singular Value Decomposition (SVD) analysis is applied to the training images to yield the best orthogonal

basis. As a result, surface albedo at each surface point is computed and the 3D surface of the given face is reconstructed up to a GBR transformation. With the geometry of the surface, [extreme](#) image rays (vertices in the illumination cone) can easily be calculated. Any image in the cone is a convex combination of those image rays (or extreme rays). The dimension of the illumination cone is $O(n^2)$, where n is the number of distinct surface normals of the object. Clearly, all images of an illuminated face do not lie in a low-dimensional space as widely believed before. However, it was also observed that the illumination cone is “thin”. Face recognition experiments were carried out by measuring the distance of a given image to the constructed illumination cone of each face in the library to find the best match. It is a convex optimization problem with well developed solutions [\[50\]](#).

Compared to PCA-based methods, the illumination cone is extrapolative, and theoretically derived. However, the construction of the illumination cone requires calculation of all extreme rays in order to cover all possible settings of different illumination conditions. What’s more, the only efficient way to handle the pose problem is to sample the pose space to create different illumination cones. Apparently, this illumination model is quite time consuming. The fact that the illumination cone is “thin” implies that even the illumination subspace is of high dimensions, it might still be well approximated by a low-dimensional space.

It is not until recently that the secret of illumination [was](#) further revealed with the help of spherical harmonic representations. Basri and Jacobs [\[68\]](#), and in parallel Ramamoorthi and Hanrahan [\[69\]](#), represented the lighting function and the reflectance function in the form of spherical harmonics and concluded that all images of a convex Lambertian object at a fixed pose could be approximated to high accuracy using nine or less basis images. With explicit

expressions for all basis images, it is straightforward to construct basis images from known 3-D surface geometry and surface albedo. This method is a breakthrough for the study of illumination effects since it is for the first time ever that the illumination subspace is analytically formulated, independent of any sample images. As it is the most promising and advanced illumination model, it is adopted in our work in order to analyze and extract illumination invariant texture information from images. In the next section, this model is explained in details.

5.2 Preliminary Background: Spherical Harmonics and Their Applications in Illumination Modeling

For the purpose of illumination analysis, a convex Lambertian object is assumed to be illuminated by distant light sources. Equation (5-2) shows the intensity at i th surface point for a single light source. If we neglect the surface albedo for the time being, then it simply maps surface normal to image irradiance. A reflectance map is a lookup table mapping surface normals directly to image radiances and is the foundation for most Shape from Shading (SFS) algorithms.

In reality, an object might be illuminated by numerous different types of light sources like point source, distributed light source etc. As long as they are distant sources, it is legitimate to assume they shine on the object evenly. Therefore, the intensity of the light is independent of the position in the scene. Equation (5-2) could be generalized to an arbitrary lighting environment by integrating the light over all directions as:

$$r(\mathbf{v}_r) = \int_{S^2} l(\mathbf{u}_l) \max(\mathbf{u}_l \cdot \mathbf{v}_r, 0) d\mathbf{u}_l = \int_{S^2} l(\mathbf{u}_l) k(\mathbf{u}_l \cdot \mathbf{v}_r) d\mathbf{u}_l \quad (5-3)$$

The maximum operation is denoted as a kernel function $k(\mathbf{u}_l \cdot \mathbf{v}_r)$. Later equation (5-3) is

reinterpreted as convolution on the sphere. The key to that reinterpretation is the spherical harmonic representation.

5.2.1 The Spherical Harmonic Analysis

The spherical harmonic analysis is the analogue of the Fourier series analysis for signals in time domain. The Fourier series analysis states that any periodic signal can be decomposed as a weighted sum of a set of orthonormal basis functions. The set of orthonormal functions are also defined in time domain in the form of complex sinusoid functions that are of harmonic frequencies. Spherical harmonics, literally, should be a set of harmonic functions that are defined on the surface of the sphere. The spherical harmonics analysis is a transform that can work on any spherical function in a similar way as the Fourier series analysis on any periodic time-domain function. Any spherical function (function that is defined on the surface of the unit sphere) can be written as a sum of weighted orthonormal basis. In other words, any spherical function can be transformed into another domain, which is spanned by a set of orthonormal functions. These functions are denoted as Y_{nm} , for n being positive integers and $-n \leq m \leq n$. Y_{nm} is the n 'th order harmonic defined on the unit sphere. It could be parameterized as a function in the spherical coordinate system as $Y_{nm}(\theta, \phi)$, where θ is the azimuthal angle in the xy -plane from the x -axis with $0 \leq \theta < 2\pi$, and ϕ is the polar angle from the z -axis with $0 \leq \phi \leq \pi$. Sometimes it is more convenient to express $Y_{nm}(\theta, \phi)$ as a function of the surface normal $\mathbf{v} = (x, y, z)$. $Y_{nm}(x, y, z)$ then becomes a polynomial of degree n . The first nine polynomials are

$$\mathbf{u} = (x, y, z) = (\sin(\theta) \cos(\phi), \sin(\theta) \sin(\phi), \cos(\theta))$$

$$\begin{aligned}
Y_{nm} &= Y_{n|m}^e \pm Y_{n|m}^o \\
Y_{00} &= \frac{1}{\sqrt{4\pi}} \\
(Y_{11}^e, Y_{11}^o, Y_{10}) &= \sqrt{\frac{3}{4\pi}}(x, y, z) \\
(Y_{21}^e, Y_{21}^o, Y_{22}^o) &= 3\sqrt{\frac{5}{12\pi}}(xz, yz, xy) \\
Y_{20} &= \frac{1}{2}\sqrt{\frac{5}{4\pi}}(3z^2 - 1), \quad Y_{22}^e = \frac{3}{2}\sqrt{\frac{5}{12\pi}}(x^2 - y^2)
\end{aligned} \tag{5-4}$$

The superscripts e and o denote the even and odd components of the harmonics. Since in practice, we only deal with real functions. The even and odd versions are more convenient to use.

For any piecewise continuous function $f(\mathbf{u})$ defined on the surface of the sphere, the transform pair for the spherical harmonic analysis is

$$\begin{aligned}
f(\mathbf{u}) &= \sum_{n=0}^{\infty} \sum_{m=-n}^n f_{nm} Y_{nm}(\mathbf{u}) \\
f_{nm} &= \int_{s^2} f(\mathbf{u}) Y_{nm}^*(\mathbf{u}) d\mathbf{u}
\end{aligned} \tag{5-5}$$

Anybody with a little bit knowledge of the Fourier series analysis can see the high resemblance between these two transforms. Their differences are also apparent. Not only the targeted functions in spherical harmonic analysis are defined on the surface of the unit sphere, but for the n'th order, there exist $2n+1$ harmonic functions Y_{nm} , for $-n \leq m \leq n$.

For the Fourier analysis, the convolution theorem states that convolution of two functions in time domain is equivalent to the multiplication of their transformed functions in the frequency domain. There is a similar theory called the Funk-Hecke theorem [68] for the spherical harmonic analysis. Basically, the Funk-Hecke theorem says that convolution of two spherical

functions equals multiplication of the coefficients of the spherical harmonic expansions of the two functions.

5.2.2 Illumination modeling by Spherical Harmonic Analysis: from the Lighting function to the Reflectance Function and Basis Images

Illumination on a Lambertian surface is characterized by the reflectance function in equation (5-3). $r(\mathbf{v}_r) = \int_{S^2} l(\mathbf{u}_l) k(\mathbf{u}_l \cdot \mathbf{v}_r) d\mathbf{u}_l$. The kernel $k(\mathbf{u}_l \cdot \mathbf{v}_r)$ is the maximum of the inner product $\mathbf{u}_l \cdot \mathbf{v}_r$ and zero. For different \mathbf{v}_r , it is a rotated version of the same function. The reflectance function is a convolution of the lighting function $l(\mathbf{u}_l)$ and the unrotated kernel function $k(\mathbf{u}_l)$ (fix \mathbf{v}_r on the northern pole) on the surface of the unit sphere. The lighting function $l(\mathbf{u}_l)$, in the form of weighted sum of spherical harmonics, is:

$$l = \sum_{n=0}^{\infty} \sum_{m=-n}^n l_{nm} Y_{nm} \quad (5-6)$$

The kernel function $k(\mathbf{u}_l)$ has only the zonal harmonics since it is circularly symmetrical about the northern pole.

$$k = \sum_{n=0}^{\infty} k_n Y_{n0} \quad (5-7)$$

The reflectance function r is the spherical convolution of function l and k . Then r , in terms of the expansion coefficients of l and k , is:

$$r = \sum_{n=0}^{\infty} \sum_{m=-n}^n \left(\sqrt{\frac{4\pi}{2n+1}} k_n l_{nm} \right) Y_{nm} \quad (5-8)$$

Equation (5-8) associates the reflectance function with the lighting coefficients. However this expansion has infinite number of basis functions. How many basis functions are needed in order to have a satisfactory approximation of the reflectance function? The answer to this

relies on the analysis of the kernel function.

The unrotated kernel function is just a half cosine function. $k(\mathbf{u}_l) = \max(\cos(\theta), 0)$. As an explicit function, it is not difficult to compute all coefficients of its spherical harmonic expansion. It is shown that this kernel could be approximated with only a few basis functions. Its first three coefficients basically dominate the kernel, with 99.22% of the energy being preserved. In other words, the kernel acts as a low-pass filter. When convolving with the lighting function, the high frequency components of the lighting function will be suppressed. As a result, the reflectance function could also be approximated with only a few low-order spherical harmonics, no matter how complicated the lighting function might be. The accuracy of the approximation depends on the energy distribution of the specific lighting function. A lower bound on the accuracy of the approximation for any lighting function is given in [68], assuming all the higher order components are saturated. So consider both the kernel and the lighting function, even in the worst case when using a second order approximation, “the accuracy of the approximation for any lighting function exceeds 97.96% [68]”.

$$r \approx \sum_{n=0}^2 \sum_{m=-n}^n \left(\sqrt{\frac{4\pi}{2n+1}} k_n l_{nm} \right) Y_{nm} = \sum_{n=0}^2 \sum_{m=-n}^n l_{nm} r_{nm} \quad (5-9)$$

Image intensity function (5-2) differs from the reflectance function in (5-3) with the surface albedo. $\mathbf{I}_i = \rho_i r(\mathbf{n}_i)$, let $b_{nm}(\mathbf{p}_i) = \rho_i r_{nm}(\mathbf{n}_i)$. Denote \mathbf{b}_{nm} as the n'th harmonic image, any image is just a linear combination of harmonic images.

$$\mathbf{I}_i \approx \sum_{n=0}^2 \sum_{m=-n}^n l_{nm} \mathbf{b}_{nm}(\mathbf{p}_i) \quad (5-10)$$

Arrange all nine harmonic images column-wisely to form a big matrix.

$B = [\mathbf{b}_{00}, \mathbf{b}_{10}, \mathbf{b}_{11}, \dots, \mathbf{b}_{22}]$. Let \mathbf{L} be a vector of all lighting coefficients. Equation (5-10)

could be reinterpreted as simple matrix multiplication as

$$\mathbf{I} \approx \mathbf{BL} \quad (5-11)$$

From the lighting function and the reflectance function to the harmonic basis images, the accuracy of the low-order approximation and the orthogonality of these basis functions have to be reconsidered. First of all, generally these basis images are not orthogonal, even though the original spherical harmonics are orthogonal functions on the unit sphere. This is easy to see since the basis images are generated by scaling surface reflectance values with surface albedos. Not only the distribution of the surface normals for a human face is by no means similar to that of the unit sphere, but all surface normals are somehow emphasized or deemphasized due to different surface albedo on the surface. Therefore practically, it is possible to render an image with a lighting configuration so that the low-order approximation is very poor. However on average, the low-order representation provides a good approximation for the illumination subspace of human faces. First nine spherical harmonics are said to have an average accuracy of approximation about 99.22%.

Equation (5-11) should be the simplest way to understand illumination phenomenon with the help of the spherical harmonic representations. If the 3D shape and surface albedo property of a face is already known, its illumination space (the set of images under all possible lighting configurations when viewed from a fixed angle) is spanned by some basis images that can be analytically computed. What's more, the illumination space could be approximated well by a low-order subspace of nine dimensions.

This analytically derived illumination subspace can be applied to synthesize novel face images under arbitrary illumination conditions by simply editing the nine lighting coefficients

as explicitly shown in equation (5-11). On the other hand, given an image of the person under novel illumination, the lighting coefficients could be estimated by solving a simple least squared problem:

$$\hat{\mathbf{L}} = \arg \min_{\mathbf{L}} \|\mathbf{I} - \mathbf{B}\mathbf{L}\| \quad (5-12)$$

Often the surface albedo information is unknown. Since human face has approximately the same skin color, it is reasonable to assume constant albedo when estimating lighting coefficients with (5-12).

If for a big database, each person is associated with a 9D illumination subspace. Given a face image of unknown identity, its distance to the illumination space of a known person measures the similarity of these two faces. Specifically, it is the distance between the input image and the nearest image that the illumination space can possibly generate under arbitrary lighting condition. Arbitrary linear combination of the harmonic basis images might generate physically impossible images since arbitrary lighting coefficients might correspond to a lighting function with negative values. Since a negative light is impossible, it is necessary to enforce nonnegative light. The nonnegativity is usually obtained by approximating the low-order illumination subspace with linear combination of a fixed set of directional light sources [50] [70].

5.3 Extraction of Illumination Invariant Texture Map from Images

The analytic description of the low-order illumination subspace makes it possible to analyze illumination effects without gathering statistical information from a large number of images.

This is probably the most distinctive advantage of adopting the spherical harmonics based

illumination model. All it requires are the 3D surface shape description and the surface albedo property. Since the analytical illumination space represents the set of all possible images of a face under arbitrary illumination condition, the distance between an input image and the illumination space serves to measure the similarity of the unknown face and the given face. It is of our great interest to recover surface albedo information, or loosely speaking, illumination-free appearance for the purpose of synthesis and recognition. In our work, for any specific face, an illumination invariant texture map is generated given the 3D face structure and one or more face images.

In previous chapter, 3D face surface is modeled by warping a generic face model to fit feature points and face contours extracted from input images. The morphing function is a cubic polynomial and the optimal morphing parameters are called shape parameters. The morphed 3D face represents the fundamental face structure of that person. Surface details could be estimated by interpolating nearest face mesh vertices. Estimating illumination coefficients and illumination free texture map could be carried out in two different frameworks.

5.3.1 View-dependent Illumination Editing and Normalization

Illumination editing could also be referred as face relighting. Started with face image(s) and the underlying 3D face structure of a person, we lack accurate surface albedo information in order to build a low order illumination space. If the pose is fixed, the albedo problem can be circumvented with a simple yet practical solution [71]. A face image could be relit without explicitly solving the surface albedo. Assume for the original face image, the image intensity at point $\mathbf{u} = (x, y)$ is decided by $\mathbf{I}(\mathbf{u}) = \rho(\mathbf{u}) \cdot r(\mathbf{n})$, where $\rho(\mathbf{u})$ is the surface albedo

of this point. $r(\mathbf{n})$ is the reflectance function. Under a different illumination environment, the intensity function would be $\mathbf{I}'(\mathbf{u}) = \rho(\mathbf{u}) \cdot r'(\mathbf{n})$. Apparently, the new image is related to the original image as

$$\begin{aligned} \mathbf{I}'(\mathbf{u}) &= \rho(\mathbf{u}) \cdot r'(\mathbf{n}) = \frac{\mathbf{I}(\mathbf{u})}{r(\mathbf{n})} \cdot r'(\mathbf{n}) \\ &= \mathbf{I}(\mathbf{u}) \cdot \frac{\sum_{n=0}^2 \sum_{m=-n}^n l'_{nm} r_{nm}(\mathbf{n})}{\sum_{n=0}^2 \sum_{m=-n}^n l_{nm} r_{nm}(\mathbf{n})} \end{aligned} \quad (5-13)$$

Since the lighting coefficients $\{l_{nm}\}$ could be estimated using equation (5-12). The harmonic reflectance $r_{nm}(\mathbf{n})$ is only a function of the surface normal (see equation (5-8) for details). Equation (5-13) clearly shows that a novel image could be generated by directly editing the values of the lighting coefficients $\{l'_{nm}\}$.

Editing the lighting coefficients is the easiest way to see how illumination change affects the appearance of a person. However, controlling the lighting environment by editing the lighting coefficients does not make any sense practically. Typically, a face is illuminated by arranging several point lights in the scene. For example, a single directional light is a delta function on the unit sphere. The spherical harmonic coefficients could be derived from the fundamental transform pair in equation (5-5). Then we can say the new face image is illuminated under that single directional light.

If the camera-face geometry is fixed and rotated in the same lighting environment, the resulting image can also be predicted easily. Assume for the same image point, surface normal changes from \mathbf{n}_a to \mathbf{n}_b . The new image is then

$$\mathbf{I}'(\mathbf{u}) = \rho(\mathbf{u}) \cdot r(\mathbf{n}_b) = \frac{\mathbf{I}(\mathbf{u})}{r(\mathbf{n}_a)} \cdot r(\mathbf{n}_b) = \mathbf{I}(\mathbf{u}) \cdot \frac{r(\mathbf{n}_b)}{r(\mathbf{n}_a)} \quad (5-14)$$

To summarize, the new image can be generated by multiplying the original image with the ratio of the reflectance function of the surface normal after and before the rotation.

For comparison of different face images of a fixed pose, the illumination problem could be handled by either extracting illumination free features or normalizing all illumination conditions to one standard. This could be realized by modifying every face image so that it matches the lighting condition of a canonical face image. In other words, the coefficients $\{l_{nm}'\}$ are set to be a standard lighting condition.

Normalized face images could also serve as the source of texture map for the purpose of image synthesis. Just like a typical view-dependent texture mapping operation, the result is comparatively noise sensitive.

5.3.2 Extraction of View-independent Illumination-free Texture Map

Given several images of a person viewed from different angles and maybe under various different lighting conditions, an illumination-free texture map is extracted by fusing different face images. A view-independent illumination-free texture map is very useful for the purpose of face recognition and face synthesis.

In order to fuse intensity information from different images and represent 3D surface albedos, a texture map is necessary. It is usually defined on a 2D texture space. The cylindrical projection model is chosen to map 3D coordinates of a face model to the 2D texture space.

The cylindrical coordinates (r, θ, h) are defined in terms of the Cartesian coordinates (x, y, z)

as

$$\begin{aligned}
 r &= \sqrt{x^2 + y^2} \\
 \theta &= \text{atan}\left(\frac{y}{x}\right) \\
 h &= z
 \end{aligned}
 \tag{5-15}$$

Global texture maps are defined on the 2D texture space of variables (θ, h) . Figure 5.1 demonstrates the mapping between the generic face model and the texture reference coordinate system.

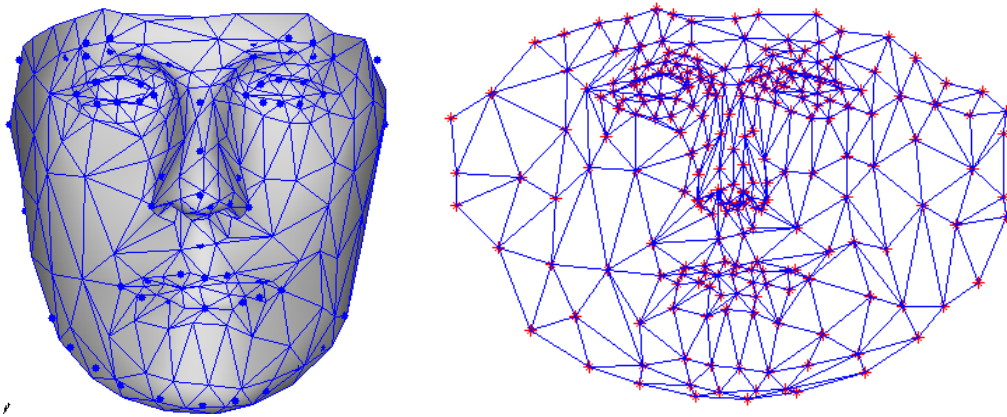


Figure 5.1 The generic face model (left) and the texture space (right)

The generic face model is first rotated a little bit so that no overlap occurs after the mapping. Note that the 2D mesh structure in the texture space is projected from the generic face model, therefore it is shapeless in a sense it doesn't reflect the 3D face structure of any specific person. Given a texture map, 3D shape parameters and pose parameters, a 2D face image could be rendered. It is desirable that the rendered image is as close as possible to the illumination space where the available face image of that person lies. In order to incorporate texture information from face images of different viewing angles, the cost function on the texture reference coordinate system s_0 is:

$$f(\rho, l_{nm}) = \sum_{i=1}^{n_0} \sum_{\mathbf{u} \in S_0} w_i(\mathbf{u}) \cdot [\rho(\mathbf{u}) \cdot r_i(\mathbf{n}_u) - I_i(\mathbf{u})]^2 \quad (5-16)$$

where

$$r_i(\mathbf{n}_u) = \sum_{n=0}^2 \sum_{m=-n}^n l_{nm}^i r_{nm}(\mathbf{n}_u)$$

is the reflectance function. The summation is over all available face images of the person. For each pose, a weight function is enforced so that the conversion from the image frame to the texture space is compensated. What's more, the invisible pixels are excluded by setting their weights to be zero. The surface albedos and illumination coefficients for each pose are solved iteratively. First, fix the illumination coefficients for i th image, $\{l_{nm}^i\}$ are estimated. Then $\rho(\mathbf{u})$ can be updated as:

$$\rho(\mathbf{u}) = \frac{\sum_{i=1}^{n_0} w_i(\mathbf{u}) \cdot r_i(\mathbf{n}_u) \cdot I_i(\mathbf{u})}{\sum_{i=1}^{n_0} w_i(\mathbf{u}) \cdot r_i(\mathbf{n}_u) \cdot r_i(\mathbf{n}_u)} \quad (5-17)$$

After the minimization procedure converges, not only the illumination coefficients for each face image are estimated, but a global illumination-free texture map is extracted. Synthesis of novel images with the extracted 3D face structure and the global texture map is straightforward.

5.4 Experiments

Images in our database are obtained from different sources. Most of them were taken without strict control of the lighting environment. Attached shadows and cast shadows are present in quite some images. This makes it necessary to analyze illumination phenomenon and extract illumination-free texture map.

5.4.1 View-based Illumination Analysis

In this section, illumination analysis is conducted on the image frame and face images of a person are handled independently according to different viewing angles. In order to apply equation (5-12) to estimate the illumination coefficients, the harmonic basis images $\{\mathbf{b}_{nm}\}$ have to be constructed first, which requires the knowledge of both the 3D surface structure and the surface albedo information. Since we only have the approximate 3D surface shape reconstructed from morphing a generic model, usually it is assumed that the surface albedo is constant when estimating the illumination coefficients in equation (5-12). In Fig. 5.2, the bottom row shows a textureless face of the reconstructed 3D face with the same poses as in the original input images (shown in the top row). A single point light source along the optical axis is assumed.

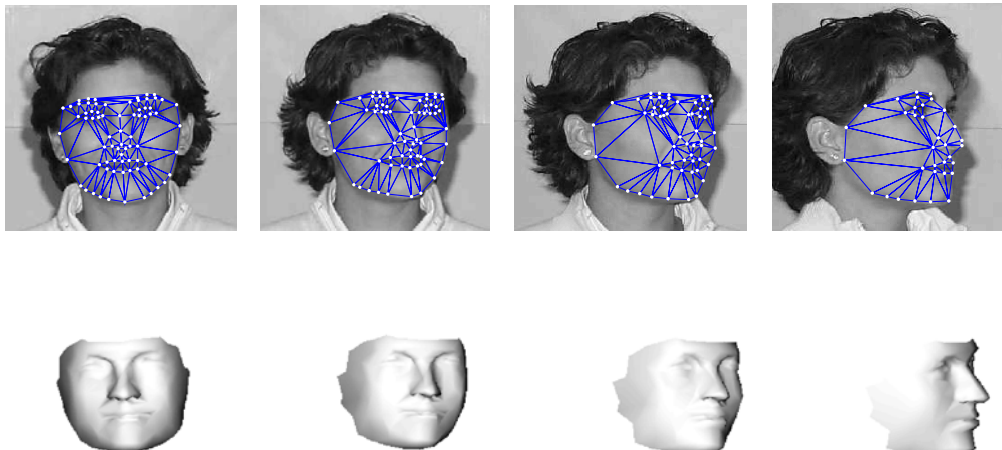


Figure 5.2 Four views of a texture face based on the reconstructed 3D face

5.4.1.1 Illumination Editing

Once the illumination coefficients are estimated, it is very easy to manipulate the coefficients to show different illumination effects based on equation (5-13). Fig. 5.3 shows the original images on the top left corner. In the second column, lights mainly come from left and right. The third column shows the simulation results of illuminating from top and bottom. The last column shows two more results with more complicated illumination conditions. Artificial effects can be seen due to the approximation of the face surface by triangular patches.

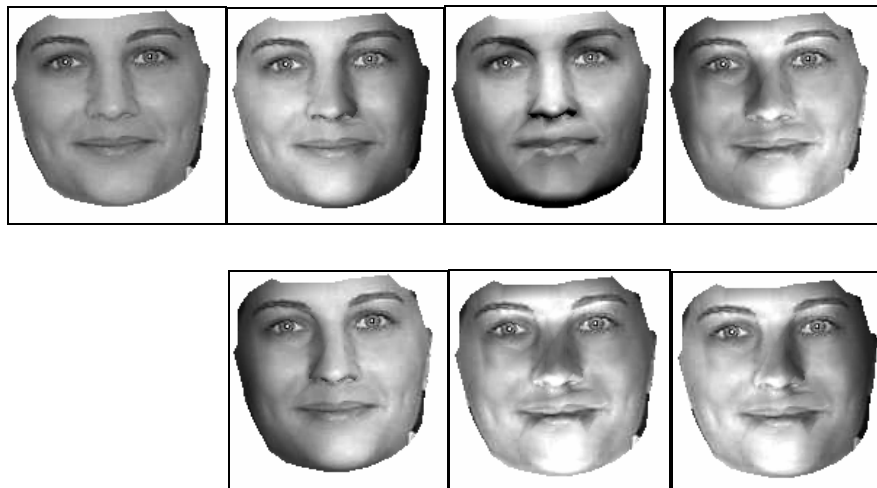


Figure 5.3 Simulation of different illumination effects by editing coefficients directly

The same experiment is repeated for this person for a different viewpoint. The simulation results are shown in Fig. 5.4.

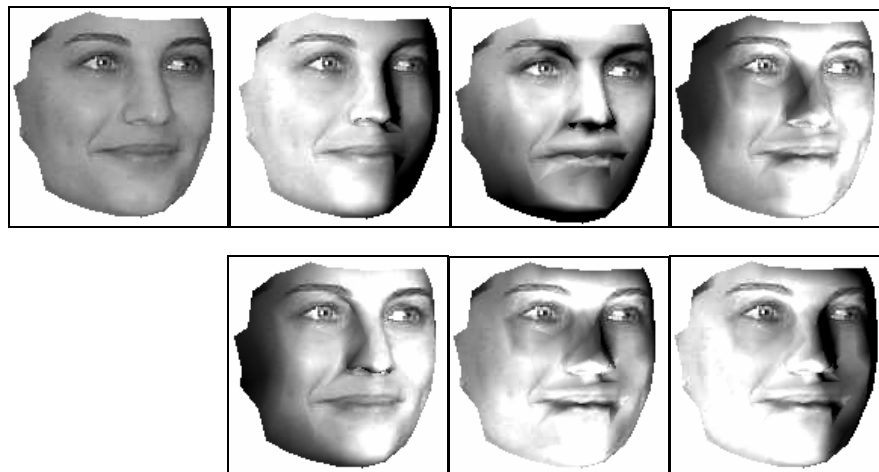


Figure 5.4 Simulation of different illumination effects another viewing angle

5.4.1.2 Rotation in the same Illumination Environment

Another way of simulating the illumination effect is to let the face rotate in the same lighting environment. Fig. 5.5 shows the experiment results by assuming the face is rotated around y-axis from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ with a step size $\frac{\pi}{10}$. The simulation shows very realistic and convincing illumination results. The partial distortion and triangular shadows are due to the lack of surface details and they could be avoided by improving the underlying 3D surface description.

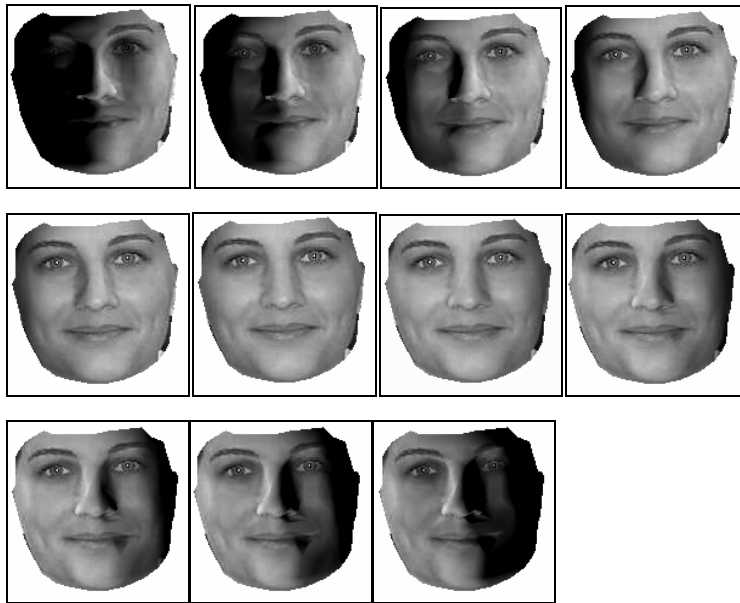


Figure 5.5 Simulation of rotating the face in the same illumination environment

5.4.1.3 Illumination Regulation by “Copying” Illumination Effect

When recognizing or verifying different faces of the same viewpoint, the variance caused by different illumination environment can be minimized by either extracting illumination-free surface albedo map or regulating all face images as if they are illuminated in the same environment. The regulation could be done by modifying the illumination coefficients of each face image to match the illumination coefficients of an example image. In another word, the illumination environment of the example image is copied and duplicated. Figure 5.6 shows in the first row two images in our database. The left image is regulated according to the illumination in the right image. The regulated image is shown as the left one in the second row. Apparently, after the regulation, the heavy shadow along the right nose bridge and cheek in the original image has disappeared. Normally this is the desired illumination effect. If, on the contrary, the right face image is regulated according to the illumination condition of the left

image, the result is displayed as the right face image in the second row.



Figure 5.6 Illumination "copying" example: top row: two original images; bottom row: two regulated images that copied illumination effect from each other.

5.4.1.4 Synthesis of Novel Faces

For view-based synthesis of novel face images, the procedure is very similar to the traditional view-based texture mapping. That is, in order to synthesize a novel image, we select one or two images with the nearest pose from all available images of that person and it serves as the texture map. The difference here is the illumination regulation. Illumination of the available image is first regulated before mapping to the novel image and the resulted illumination is also under control. To better manifest and compare the synthesis result, the experiment is simplified as follows: we try to synthesize the images that are already in the database from different images of the same identity with different viewpoints. Since those images are already in the database as ground truth, it is very straightforward to evaluate the synthesis results visually or by measuring their differences. Fig. 5.7 shows four original images in the

first row. On the second row, the novel image that falls in the pose category 2 is synthesized based on the frontal image. The synthesized image of pose category 3 is synthesized based on the original image categorized as pose 2. Similarly, the last image is synthesized based on the original image of pose 3. Novel images with larger azimuth angles are synthesized from existing images with smaller azimuth angles. In the way, the occlusion problem is very much relieved.

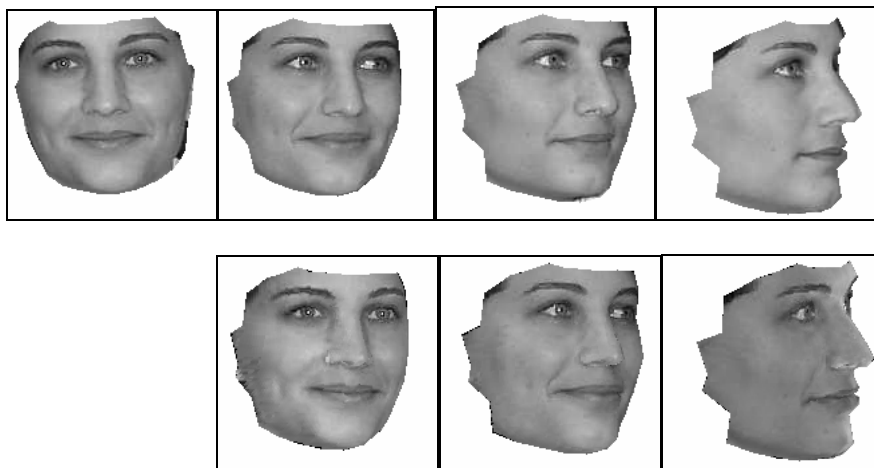


Figure 5.7 Synthesized images (bottom row) versus original images (top row)

5.4.2 View-Independent Illumination Analysis

5.4.2.1 Extraction of Texture Map

View-dependent illumination analysis analyzes different illumination effects for a fixed viewpoint. Different viewpoints are handled separately. View-independent illumination analysis, as its name indicates, coordinates face images of different viewpoints and simultaneously analyzes the presented illumination. Since they are images of the same object,

they share the same underlying 3D surface structure and surface reflection properties. With the spherical harmonic analysis, a global illumination-free texture map can be generated, which in return, refines the illumination analysis result for each viewpoint. The iterative optimization procedure is carried out on the 2D texture space. Figure 5.8 shows four original face images mapped on the texture space. Areas in black indicate occlusions on the face for a specific viewpoint. The warping operations from original face images to the universal texture space are uniquely determined by the extracted 3D surface structure and pose parameters. Inaccurate estimation of these parameters leads to bad correspondence between different viewpoints. Inaccurate face structure estimation is usually due to the inaccurate 2D feature extraction. It also happens when a generic model is morphed to fit all available 2D extracted features in different images. To minimize the error propagation caused by the 3D modeling approximation, these face images on the texture space are warped directly according to the texture space triangulation and the triangulation outputs from the view-based AAMs.

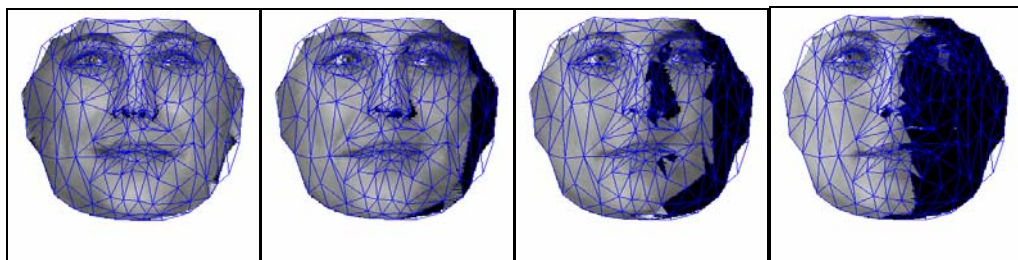


Figure 5.8 Original face images on the texture space

For the same face point in different images, different weights are assigned. It is necessary for the estimation of a global illumination-free texture map. On the one hand, for a surface point whose surface normal is pointing away from the camera-object axis, it has lower resolution

than when it is along the optical axis. The deformations from the original image planes to the texture space are taken into consideration to reflect the reliability of each image pixel. Fig. 5.9 shows weight functions for the above four images on the texture space.

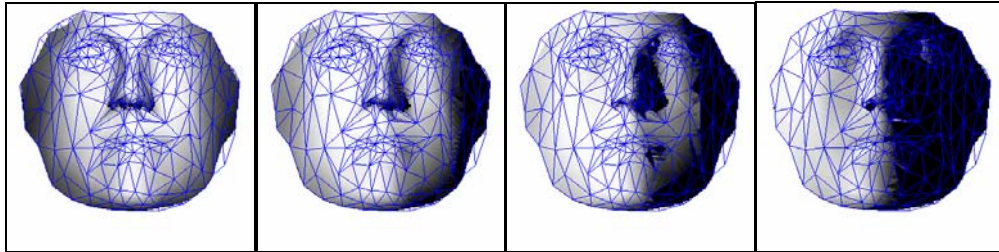


Figure 5.9 Weight functions for the images in Fig.5.8

While it is not easy to utilize the symmetric properties of both the human face structure and the surface reflection properties for view-dependent algorithms, it is very straightforward to add both as constraints for the view-independent analysis since it is carried out on the symmetrical texture space. Based on the surface normal field for the morphed 3D face surface, a symmetric normal field is generated. During the iterative optimization, the extracted surface albedo map, as calculated in equation 5.17, is also made symmetric before the next iteration.

Fig. 5.10 is the resulting illumination-free texture map after 20 iterations.

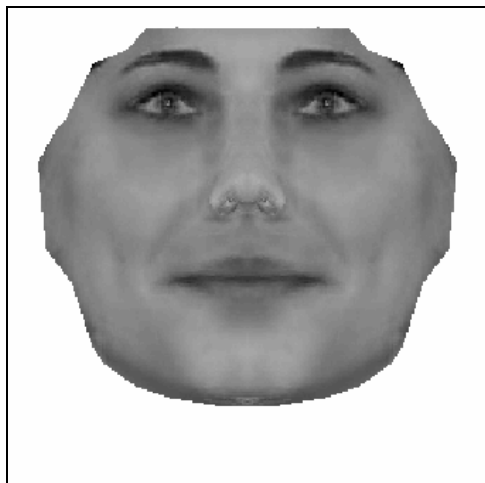


Figure 5.10 An illumination-free texture map after 20 iterations

5.4.2.2 Synthesis of Novel Faces

For view-dependent face synthesis, a good result could be achieved only when a close pose is found in the available face images of the person. View-independent face synthesis has the merit of synthesizing a fair face image regardless of the desired pose parameters. Besides, only one global texture map is required to generate the surface texture information. In order to make a comparison to view-dependent face synthesis, the global texture map is utilized to generate novel images at exactly the same poses as available images in our database. Figure 5.11 shows the synthesized results on the bottom row.

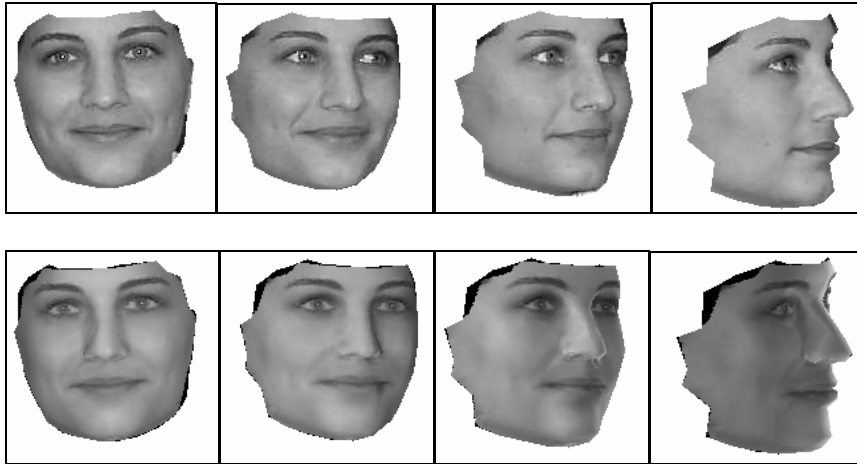


Figure 5.11 Synthesized images (bottom row) versus original images

Compared to Fig. 5.7, these synthesized images are a little blurry. This is the effect of global fusion. More synthesized images are shown in Fig. 5.12.

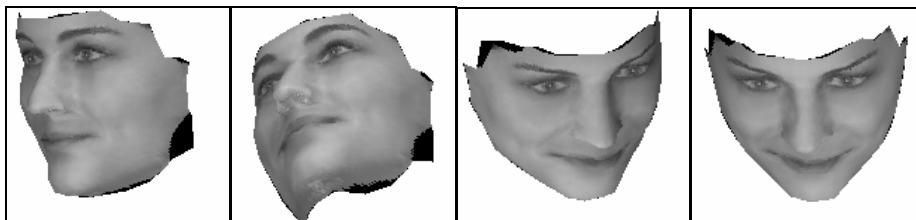


Figure 5.12 Synthesized images of arbitrary poses

5.5 Conclusion

In this chapter, the latest analytic illumination model is integrated in our work. Under the spherical harmonic representation, the interaction between the light and object surface, which is a convolution operation on the unit sphere, becomes simply the multiplication of their spherical harmonic coefficients. The illumination space of any convex object can be

analytically expressed given known surface structure and surface reflection properties. We have shown how easy it is to manipulate the illumination effect. Our view-dependent illumination analysis focuses on the analysis from a fixed viewpoint, while view-independent illumination analysis tries to fuse information from different viewpoints to make more robust estimation of the illumination coefficients and a global texture map. Novel images of arbitrary pose and arbitrary illumination are synthesized and compared to existing images in the database. The illumination analysis, together with the 3D morphable model, provides a powerful generative tool for image synthesis. By applying illumination-regulated or illumination-free face images for face recognition or verification, better performance is expected.

The morphed 3D face structure is used as input only. It is possible to imply the surface structure from the illumination model (that is how SFS algorithms work). However, it is a big challenge. There are several reasons we didn't go further in that direction. First, the illumination model captures the majority of the illumination variations, which is basically the low frequency part. So by its nature, it is not a good technique to recover high frequency surface details. The estimation from the cubic morphing and view-based AAM is good enough to represent the main shape of the face surface. At last, those shape-from-shading algorithms usually recover a noisy surface normal field first, followed by reconstructing an integrable surface. This procedure is often accompanied by a smoothing operation that in fact loses surface details, which is absolutely not desired. Nevertheless, it is still a very intriguing topic to imply or refine the estimated face structure from illumination models.

CHAPTER 6 : 3D FACE RECOGNITION FOR IMAGES AT ARBITRARY POSES AND UNDER ARBITRARY ILLUMINATION CONDITIONS

Face recognition and identification is the ultimate goal of most face applications. When an unknown face is presented in an image, usually it is first detected with face detection (face alignment) techniques, followed by analysis of its viewing angle, illumination condition and possible facial expressions. It is desirable to extract identity-encoded features that are invariant of these factors for the purpose of recognition. In previous chapters, we have shown how to extract key feature points of an input image by fitting a morphable view-based AAM. With several images available for a person, it is straightforward to construct its 3D face structure. This reconstructed 3D face, on the other hand, helps to analyze illumination conditions in the scene and as a result, an illumination-free texture map could be generated. The extracted 3D face structure, together with the illumination-free texture map, completely encodes the identity of this specific person. In previous chapter, it has been manifested that a novel face image at a specific pose and under a desired illumination condition could be synthesized by regulating, rotating or copying an existing illumination setting in another face image. The results look convincing and realistic. Yet we need to study how good it is at identifying and recognizing images unseen in the training library. In this chapter, recognition experiments are carried out based on purely shape information, texture information, illumination-free texture information and a combination of them. Experiment results are

analyzed and compared afterwards.

6.1 Literature

In biometrics area, a variety of characteristics like fingerprint, eye iris, retinal, voice and face could be used to uniquely identify different persons. Though face recognition has received as much attention as other methods, in reality it doesn't have much commercial applications as others. This is because face appearance could be affected by a lot of factors like facial expression, different pose, illumination condition, as well as disguises and natural aging. For practical use, a lot of constraints have to be applied. Face recognition has been used in personal identification [72] and access controls [73]. Face recognition techniques could be divided into two main categories: feature-based and appearance-based. The latter has prevailed ever since the 90s. Typical appearance-based algorithms include eigenface algorithms [1], elastic graph matching algorithms [2], Hidden Markov Models [3] and Neural Network algorithms etc. In a survey paper [4] in 1993, appearance-based methods are compared with geometrical feature-based methods and the author concluded that appearance-based methods have better performance. Model matching algorithm is one of the earliest appearance-based methods. With years, it has evolved from simple rigid model matching algorithms to more complicated flexible models. Another good survey paper can be found in [74].

The three main challenges for face recognition task are facial expressions, poses and illumination conditions. Facial expression problem is more of an independent topic since recognition of different expressions is itself an interesting task. In terms of recognition of

facial expressions, it is more important to recognize the same expression irrespective of different identities. In the framework of face recognition, facial expressions are as undesirable as different poses and illumination conditions. Facial expressions are normally classified by tracking the motion of facial features, or by modeling how the underlying muscles stretch or contract from the anatomic point of view. We will focus on the pose and illumination problems since both are addressed in our work.

Before introducing our illumination model in last chapter, a thorough and detailed review of illumination as to its formation, elimination and modeling has given. Following is a brief introduction where pose problem is being emphasized.

When an unknown face image is presented, different approaches could be adopted. They could be roughly classified into four categories: 1) View-based approaches. 2) Class-based approaches. 3) Single image approaches. 4) 3D approaches.

1) View-based Approaches

A simple idea to deal with the pose problem is to include as many photos of different viewpoints as possible for one person. During the recognition, the pose of the input image is roughly estimated. Then it is compared to the gallery images that have similar poses. Beymer [75] proposed such an algorithm that measures template-based correlation between images. Pose estimation is based on a 2D affine transformation of three key feature points. The test image is aligned to each gallery image of similar pose and their correlation score is used for recognition. The pose space has to be sampled densely for this algorithm to work well. Therefore a lot of images of different viewpoints are needed for each person.

To overcome this drawback, an alternative approach [76] is to use synthesized images from

one single example view and include them as gallery images for this specific person. In order to synthesize images of arbitrary viewpoints from a single face image, prior knowledge of some prototype faces under different rotations is exploited and utilized given plenty of their example images of different viewpoints. Though this algorithm is not competitive compared to a real multi-view algorithm, pose problem is fairly handled and relieved. Using linear combination of example images to interpolate or extrapolate a novel image has been well studied and extended [77].

2) Class-based Approaches

In a class-based approach, images of the same identity are treated as sample points of a class in the high dimensional space. Face recognition problem is therefore reinterpreted as a clustering and classification problem where a lot of traditional pattern recognition methods could be utilized. Prior class information of human faces indicates how pose changing information might be encoded. Face recognition algorithms like eigenfaces, fisherfaces [78] etc belongs to this category. The drawback of most class-based approaches is that many example images are needed in order to extract the face class information and all poses have to be included.

3) Single Image Approaches

Face pose is not directly modeled or estimated for single image approaches. Nor any virtual view is synthesized. Usually pose-invariant features are extracted and used for the purpose of recognition. After perspective projection, there are some inter-feature point distances that are invariant for a 3D object. A rotation invariant is computed in [79] based on the extracted fiducial points like eye corners, nose tip and mouth corners. The rotation invariant is used to

recognize face images of different viewpoints in the database. The recognition accuracy is 66% for 84 test images. Wavelet based features are extracted in the elastic graph matching algorithms [2]. Wavelet features are shown to be robust to small rotation of faces, however the performance deteriorates when large rotation angle is present.

4) 3D Face Recognition

Treating face image as a projection of a 3D object and reconstructing the underlying 3D structure from one or more images is the ultimate solution to pose problem. In another word, if the algorithm is complete enough to separate and model different poses, illumination conditions, facial expressions or other factors, it is theoretically more powerful than other face recognition algorithms that only try to eliminate or suppress these factors. We have said enough in Chapter 4 about 3D face surface reconstruction. 3D face models, especially those that are based on the statistical properties and real face laser data, are up till now the most successful models in terms of reconstruction and recognition. The drawback is its computational efficiency due to enormous laser face data needed for training purpose. It also makes the optimization procedure slow and error prone.

This thesis covers the aspects of face feature extraction via alignment, 3D face structure modeling and illumination modeling. Altogether we would like to develop a face recognition system that can handle unknown face images at arbitrary poses and under arbitrary illumination conditions.

6.2 Face Recognition based on the Extracted 3D Structure and the Illumination-free Texture Map

6.2.1 Face Modeling Phase

Previous chapters have detailed every step towards building a 3D face model and extracting an illumination-free texture map for an individual with several images available. Assume a single face is presented in one face image. This assumption simplifies the otherwise complicated face detection procedure. Predefined face features are extracted by the improved view-based AAM algorithm, which first selects a suitable AAM based on the approximate pose category this face image belongs to. The hybrid constrained optimization algorithm is adopted and the algorithm is first applied on the down-sampled face image to avoid being trapped in local minima. The component-based AAM is then applied for better feature localization. After all the available face images are successfully aligned, the extracted 2D mesh structures are fed to the 3D modeling module, which morphs the generic face model to fit the distance maps generated based on the 2D meshes. After the morphing parameters are estimated, the extracted 3D face structure makes it possible to analyze the illumination conditions for the available images. For view-based applications, their illumination conditions could be normalized to standard illumination settings. Fusion of the extracted albedo maps from different viewpoints leads to a global illumination-free texture map. The cubic morphing parameters, pose parameters, together with the texture map, constitute a generative 3D face model that encodes both surface shape and surface albedo information of this individual. Fig. 6.1 illustrates the complete face modeling process.

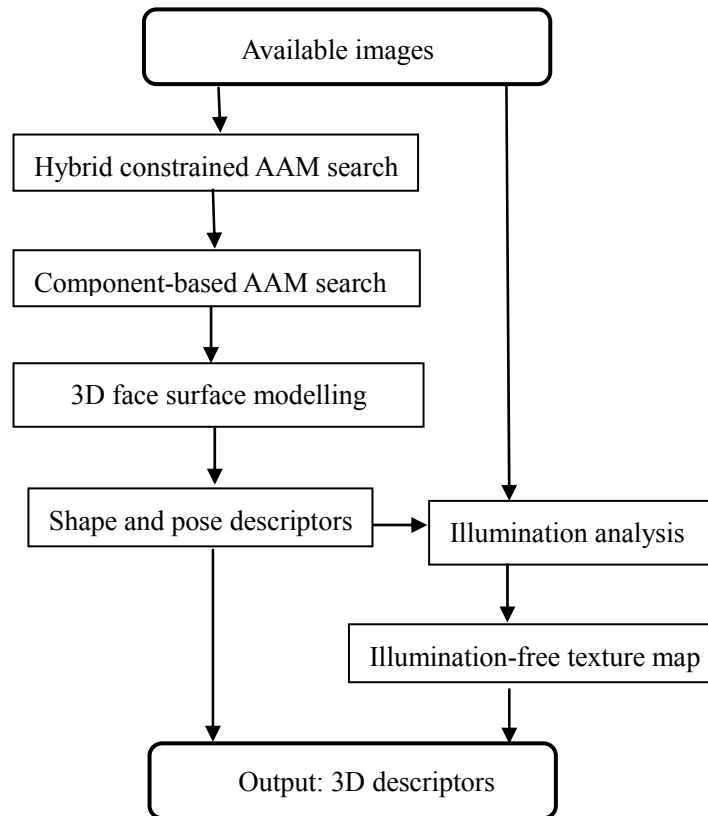


Figure 6.1 Face modeling flowchart

6.2.2 Testing Phase

For each person in the database, its 3D face structure is reconstructed and an illumination-free texture map is extracted. In previous chapter, it has been shown that any view of this person at arbitrary pose and under illumination condition could be synthesized. If there exist such a synthesized image which most resembles the test face image with unknown identity, it is reasonable to think this test image has the same origin as the synthesized one from the database (it is assumed that this test person does exist in the database). To be more precise, the difference of the synthesized image and the test image will be examined in terms of their shape, texture and regulated texture respectively. The comparison is conducted at the 2D

image level. Since the time-consuming face rendering step has to be repeated for each person in the database, the recognition process is very slow.

The face recognition task could also be conducted at the 3D level if the 3D face structure and texture information could be extracted from merely one test image. In fact, the 3DMM [5] algorithm adopts this approach. Apparently, face recognition at this level could be implemented much faster. However, the scenario is quite different here. First, we don't have a 3D face laser database that well defines and constrains the behavior of the 3D face model. Without strong constraints, one single image is just not enough to yield a good estimation of the underlying 3D face structure and a complete texture map (even when the symmetry property of human face is exploited). Therefore, our recognition task is conducted at the 2D image level despite the heavy computation involved.

For the k th individual in the database with its reconstructed 3D face structure and associated illumination-free texture map, in order to synthesize a novel view of this face that best resembles the test image, it is assumed that the test image is originated from this individual. Pose parameters for the test image could be estimated based on this assumption. The same optimization procedure as in Chapter 4 is used, except that the morphing parameters are fixed as known parameters. The cost function is similar to equation (4-15). After the optimal pose parameters $\mathbf{\Pi}_k^*$ are estimated, the average Euclidean distance of the projected feature points from the 3D face structure to the extracted 2D feature points in the test image serves as a good measurement of the matching error. If this test image is originated from this individual, the matching error should be mainly caused by estimation error instead of different identities. Therefore, the matching error would be very small. The overall decision is made based on the

smallest matching error for all individuals in the database as:

$$\min_k \left(\frac{1}{n_0} \sum_{i=1}^{n_0} \left\| \mathbf{d}(\mathbf{p}_{i,k}, \boldsymbol{\alpha}_k, \mathbf{\Pi}_k^*) \right\|^2, k = 1, 2, \dots, K \right) \quad (6-1)$$

where $\boldsymbol{\alpha}_k$ stands for the morphing parameters of the k th person. The projected 3D feature point set $\{\mathbf{p}_{i,k}\}$ is determined by the morphing parameters $\boldsymbol{\alpha}_k$ and the pose parameters $\mathbf{\Pi}_k^*$. $\mathbf{d}(\mathbf{p}_{i,k}, \boldsymbol{\alpha}_k, \mathbf{\Pi}_k^*)$ is the distance function of the i th projected feature point and its corresponding 2D extracted feature point. The correspondence problem is relaxed by using the distance map technique. Since the recognition computes only the average Euclidean distance between two set of feature points, recognition based on this measurement is purely shape-based.

As the face intensity pattern (or texture, as the two terms are abused in our work) is also unique for each person, texture-based recognition could be implemented by measuring the texture difference of the synthesized image and the test image. Assume the synthesized image and the test image has been aligned to have the same pose. Note that the shape of the synthesized image and the test image are normally different unless they originate from the same person. If the rigid pixel-wise image difference is measured, it is analogous to the traditional rigid template matching problem. The problem is, even though the computation is carried out over the region of interest for one of the two images, some unwanted background or other irrelevant information might still be included for the other image due to the shape difference. To overcome this and exclude the shape influence in the texture-based recognition, the test image is further warped to the synthesized image shape according to the fiducial points on both images. The recognition is based on the decision rule as follows:

$$\min_k \left\{ \sum_{\mathbf{u} \in S} \left| \mathbf{I}_{syn}^k(\mathbf{u}) - \mathbf{M}(\mathbf{I}_{test}(\mathbf{u})) \right|^2, k = 1, 2, \dots, K \right\} \quad (6-2)$$

where $\mathbf{M}(\cdot)$ is the 2D morphing operation to transform the test image to the synthesized image.

For the texture-based recognition, image intensity difference is measured. Up till now, the illumination in the test image is not modeled. This would cause a problem when illumination in the test image is a dominant factor. It is necessary to apply suitable illumination effect on the synthesized image so that it mimics the illuminating environment for the test face image.

$$\min_k \left\{ \min_{L_{nm}} \left\{ \sum_{\mathbf{u} \in S} \left| \mathbf{I}_{syn}^k(\mathbf{u}) \cdot \mathbf{E}_{nm}(\mathbf{n}_u) \cdot \mathbf{L}_{nm} - \mathbf{M}(\mathbf{I}_{test}(\mathbf{u})) \right|^2 \right\}, k = 1, 2, \dots, K \right\} \quad (6-3)$$

The only difference of equation (6-2) and (6-3) is that the nine optimal illumination coefficients $\{\mathbf{L}_{nm}\}$ are estimated first and the illuminated synthesized image is compared to the morphed test image. This normalized texture-based recognition is expected to have a better performance compared to the texture-based recognition where illumination is not considered.

6.3 Experiments and Discussions

Our face database is shortly introduced in previous chapters. It is a mixture of face images from public online databases and the previous face database in our lab. Though in Chapter 3, as many as 218 frontal images are used in the face alignment experiment. Unfortunately not all of them can be used for the recognition experiment due to the fact that different people in the whole face database might have various numbers of images, ranging from one to four. It doesn't mean that recognition experiment can't be carried out on a face database with

different number of images per person. However, the 3D modeling quality varies due to different number of available training face images for different individuals. Recognition experiment based on that might be biased. Therefore, only those individuals with four poses are chosen for the recognition experiment. As a result, 38 people with four images per person are used. Fig 6.2 shows one view for the selected individuals in our database.

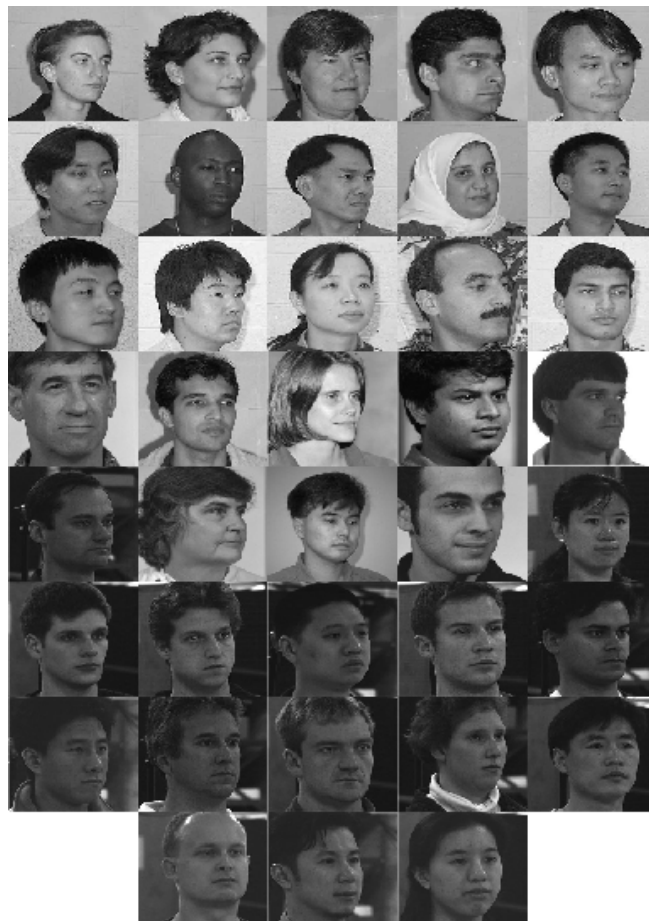


Figure 6.2 One example view of the 38 people

For face recognition, images of one viewpoint are selected as test images while three other images of different viewpoints are used to train 3D face models and generate illumination-free texture maps. The recognition experiments are repeated for all four pose categories.

6.3.1 A Typical Face Recognition Example

Assume current test pose is the front view and the rest views are used to train the 3D face models. At the training phase, the 3D face structure and an illumination-free texture map is extracted for each person in the database based on the analysis of the other three viewpoints. At the test phase, given a frontal test image, key feature points are first aligned with the view-based AAM algorithm. Assume it is the k th person in the database. The optimal pose parameters for this test image are estimated. The average point to edge distance measures the estimated shape error. A novel image with the same pose parameters is synthesized based on the 3D face structure and the texture map of the k th person. The test image is then aligned to this synthesized image to measure the texture difference. Furthermore, the synthesized image is illuminated with similar lighting environment as the test image in order to minimize the difference caused by different illumination conditions.

Let's take the frontal view of the same person we have used a lot as an example. Fig. 6.3 shows the test image.



Figure 6.3 A typical test image

The 3D face structure and the illumination-free texture map of this person is extracted based on the other three views. Assume currently the test image is tested against the individual of the same identity in the database. Based on the estimated optimal pose parameters, a synthesized image is generated as shown in Fig. 6.4(a). Based on the illumination analysis of the test image, an illuminated image is synthesized as shown in Fig. 6.4(b). The test image is warped to the same shape as the synthesized one. The warped test image is shown in Fig. 6.4(c).

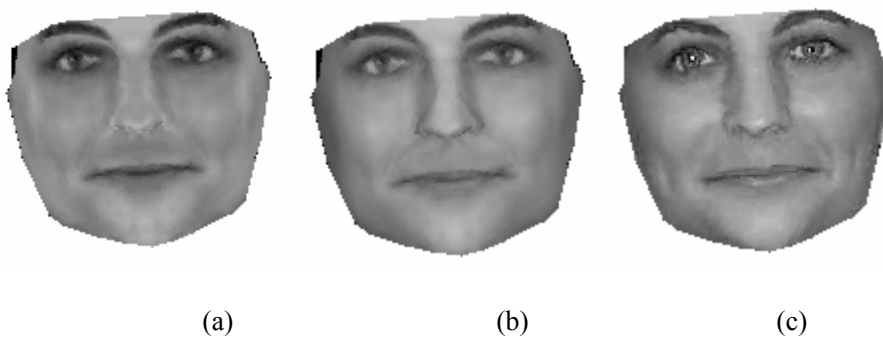


Figure 6.4 Comparison of the synthesized images and the test image: a) Synthesized illumination-free image. b) Illuminated image. c) Test image

The mean squared error of Fig. 6.4 (a) and (c) is the texture-based matching score without illumination analysis, while the difference of Fig. 6.4 (b) and (c) is the texture-based matching score after copying the illumination environment of the test image to the synthesized image.

The synthesized image is a little blurred as it is synthesized based on other three views.

Fig. 6.5 shows similar results as in Fig. 6.4 when the test image is assumed to be of other identities. Note that the test image is warped differently in order to match the shape of the person being verified.

Fig. 6.6 is a plot of matching errors for this test image. All three curves indicate that the best match is the no. 1 person in the database with minimal matching errors. The recognition is successful as this test image is the frontal view of the no.1 person indeed.

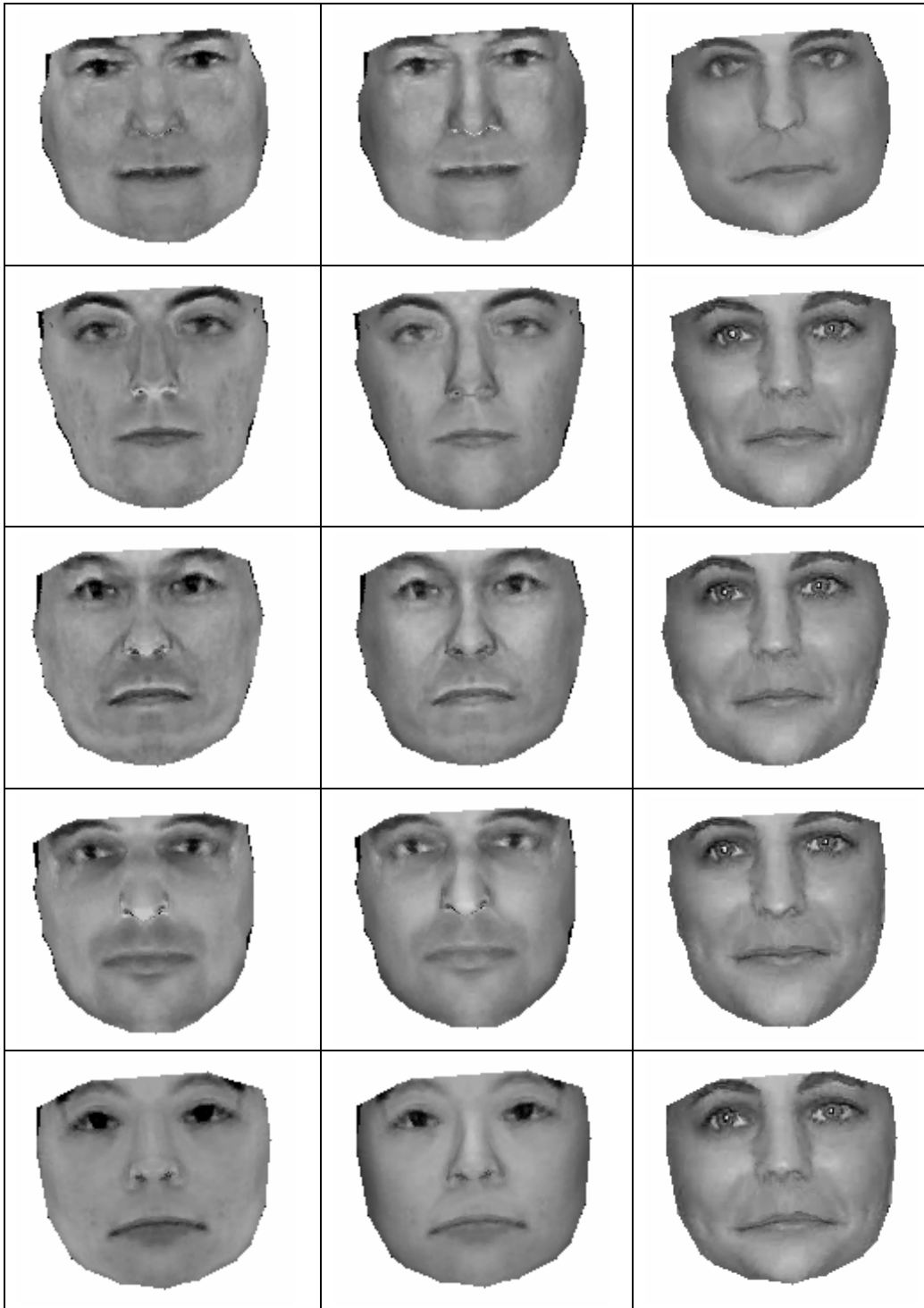


Figure 6.5 More examples of texture matching

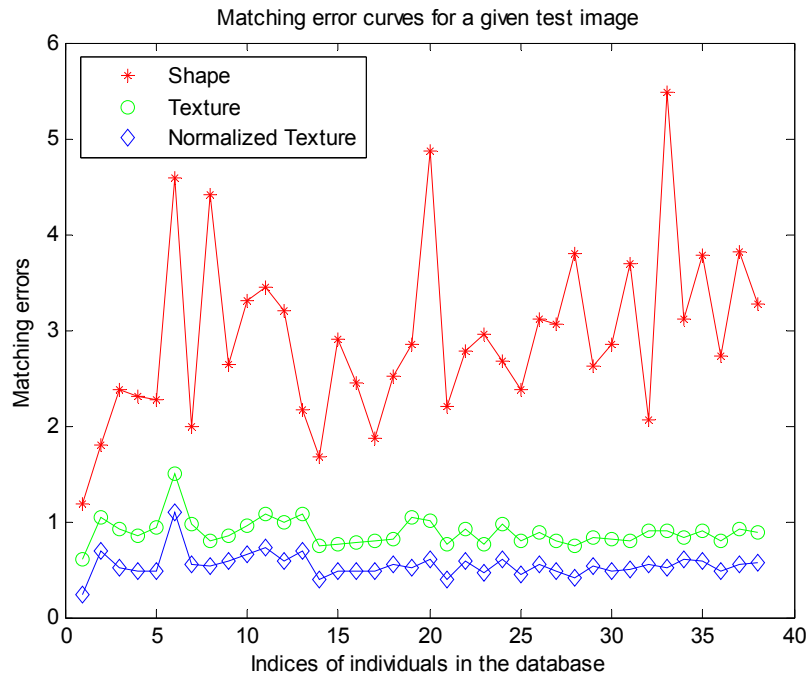
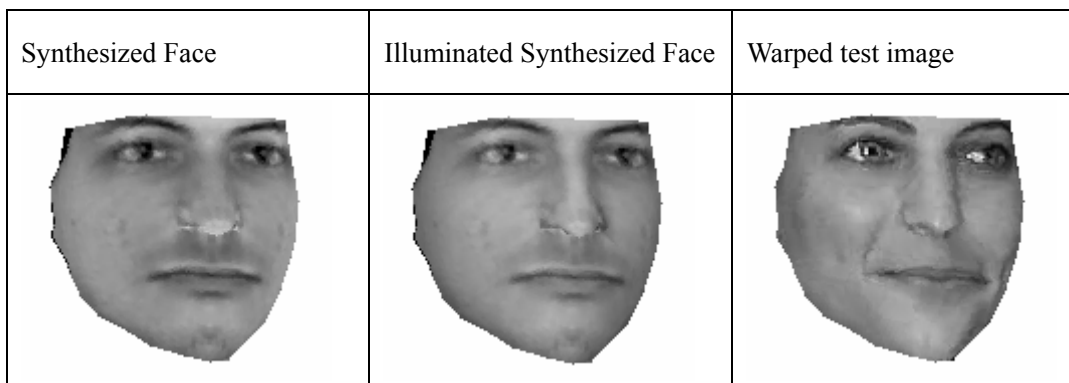


Figure 6.6 Three matching error curves

When facial images in other pose categories are set to be test set, the same recognition procedure is followed. Fig. 6.7 shows one example view per pose of the synthesized images and the test images when the same person is tested.



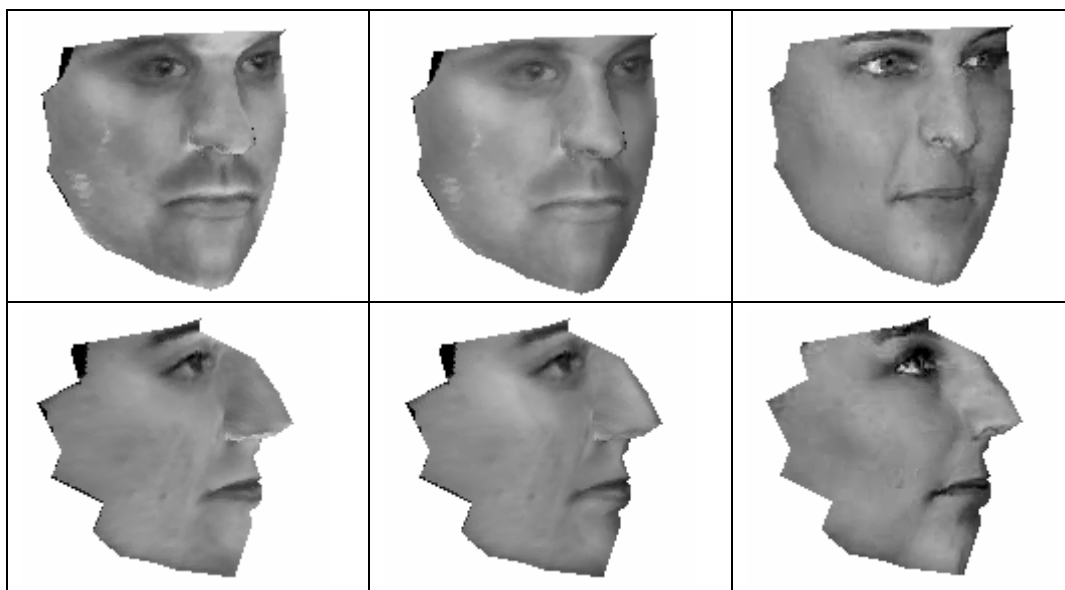


Figure 6.7 Example views of synthesized images and test images for pose 2 to pose 4 (from first row to last row)

In Fig. 6.7, the main artificial effect for the synthesized images is caused by the fusion of different views to the global illumination free texture map. The test images on the last column look unlike the original person due to the severe image warping operation.

6.3.2 Complete Recognition Results for Different Pose Categories

Given a test image, its 2D feature point set is extracted with the view-based AAM algorithm based on the approximate pose category it falls into. It is then verified against all people in our database in terms of the shape matching error, the texture matching error and the normalized texture matching error. The whole recognition procedure has been demonstrated in the previous section. The same recognition experiment is repeated on test images from all 38 people in our recognition database in sequence. In this section, experimental results based on different matching errors are compared when a specific pose is used as test pose. Later statistical performances across different poses are summarized.

Figure 6.8 shows the complete recognition results when pose category one (frontal view) is being tested. The x-axis indicates IDs of the individual being tested, and the y-axis shows the distance map residual errors by assuming the test image is originated from the total 38 3D face models respectively. The identity of the test image will be recognized to be the one in the database that yields the minimum error, marked with ‘o’ in the figure, while ‘*’ marks the residual error associated with the true 3D face model of that person. When the positions of the ‘o’ and the ‘*’ differs, the test image will be misclassified.

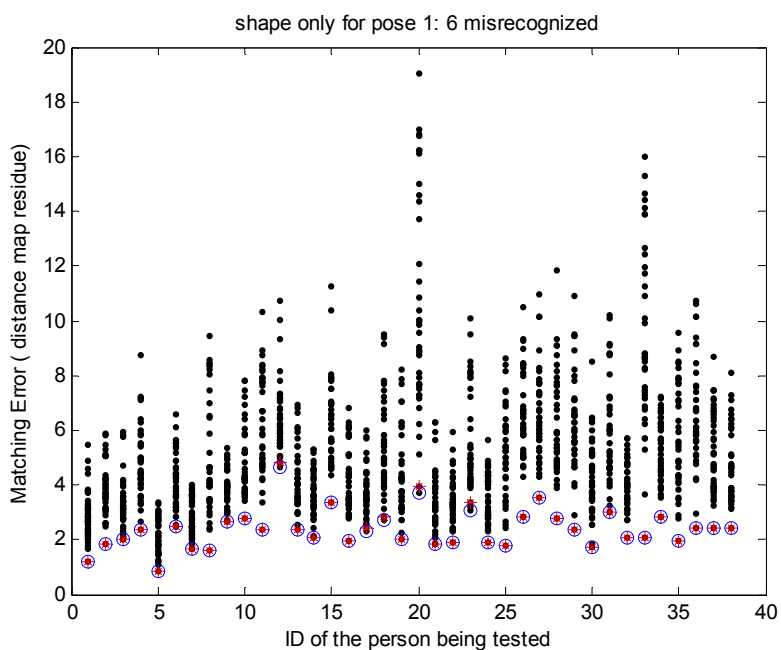


Figure 6.8 Face recognition based on distance map error

6 of the 38 people are misclassified with a recognition rate of 84%. Among the 6 misclassified cases, the true identities of 4 of them are the secondary choices. Figure 6.9 shows a misclassified case.



Figure 6.9 A misclassified case: Left image is the test image. The best match is the person in the right image.

At first it looks a little unacceptable that a male is misrecognized as a female. However they do share certain similarity in terms of the 2D shape defined as a set of feature points. When the front pose is being tested, 3D model is reconstructed from other viewpoints. From certain viewpoint, one face is possible to look like another face from a different viewpoint. This may cause misclassification. The introduction of distance map as a lookup table helps relax the correspondence problem, however it might also cause over-smoothing that results in diminished shape difference of two totally different people. Though in Fig. 6.9, their noses look very different, the difference is not so apparent in their distance maps due to the limited number of feature points depicting nose shape. In summary, recognition based merely on shape information has a sound yet limited performance.

Fig. 6.10 shows a similar plot as Fig. 6.8 for the recognition results based on texture information only. Fig. 6.11 shows complete recognition results based on the illumination-normalized texture information.

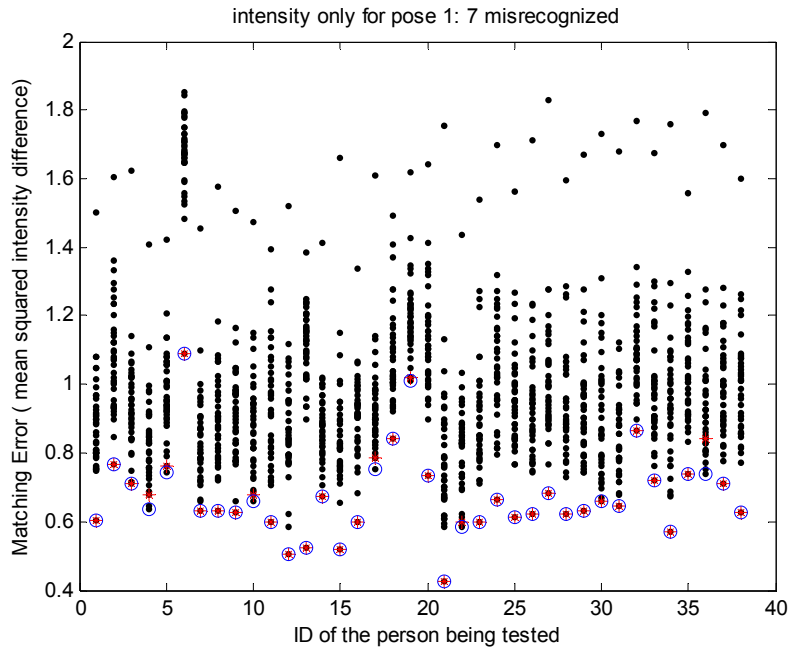


Figure 6.10 Face recognition based on texture error

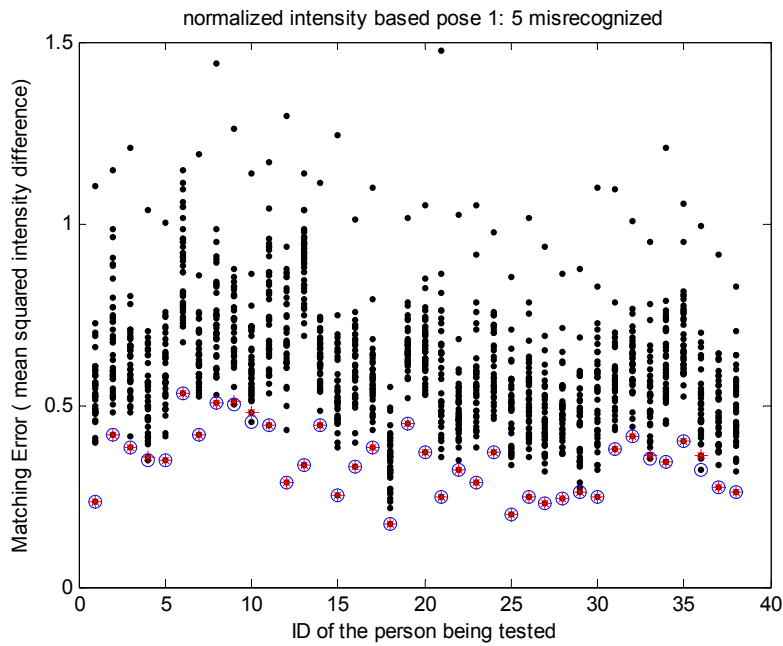


Figure 6.11 Face recognition based on illumination-normalized texture error

The misclassification rate for the texture-based recognition and the normalized texture-based recognition is respectively 7 out of 38 and 5 out of 38. Among the 7 people that are misclassified based on their texture information, one of them are also misclassified by the shape-based algorithm, while three of them are also misclassified by the normalized texture-based algorithm. Apparently the texture-based algorithm and the normalized texture-based algorithm are more correlated. Though some misclassified faces are successfully recognized after applying the illumination normalization, some others fail. It is hard to have any further conclusion except that the recognition based on the shape, texture and normalized texture have quite similar performance.

As the shape-based algorithm and texture-based algorithm are mutually complementary, we attempt to fuse them together for classification. The rationale behind the fusion is to improve the recognition rate. Here the normalized texture-based algorithm is adopted. Different weights are assigned to two errors (shape and normalized texture) and their weighted sum is used as the measurement for classification. After experimented with different weights, the best recognition rate is 100%. This recognition rate is obtainable when the shape and texture error is weighted among a wide range from 0.05:0.95 to 0.45:0.55. Figure 6.12 plots the combined errors with the shape and normalized texture weighted with 0.20:0.80. From the figure, it is also clear that matching error for the true identity is well separated from the rest.

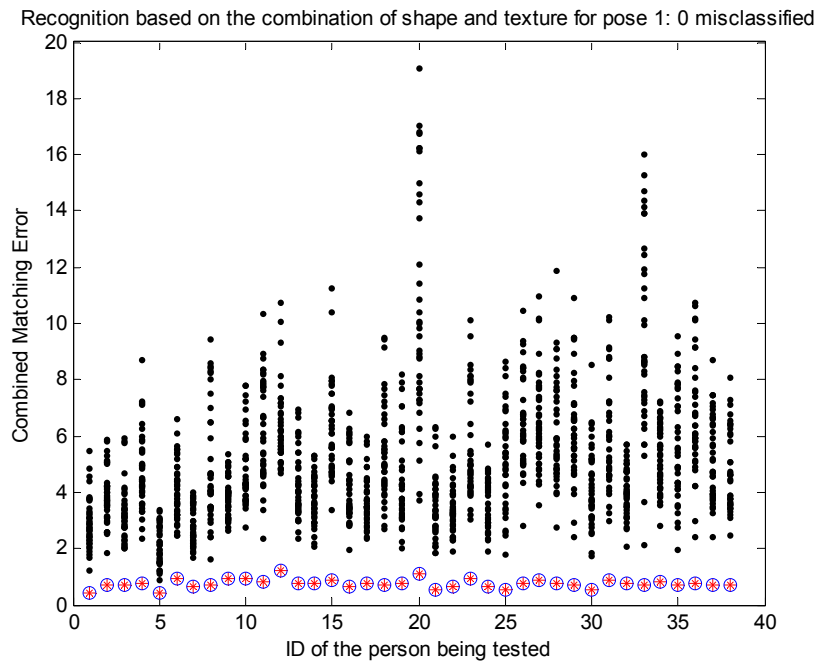


Figure 6.12 Face recognition based on the combined error

The same recognition experiment is repeated for other pose categories and their results are shown in Fig. 6.13. The first row shows the shape-based recognition results for pose 2, 3 and 4 respectively. The second row shows the texture-based recognition results and the normalized texture-based recognition results are displayed on the last row. The overall misclassification rates are summarized in Table 6.1.

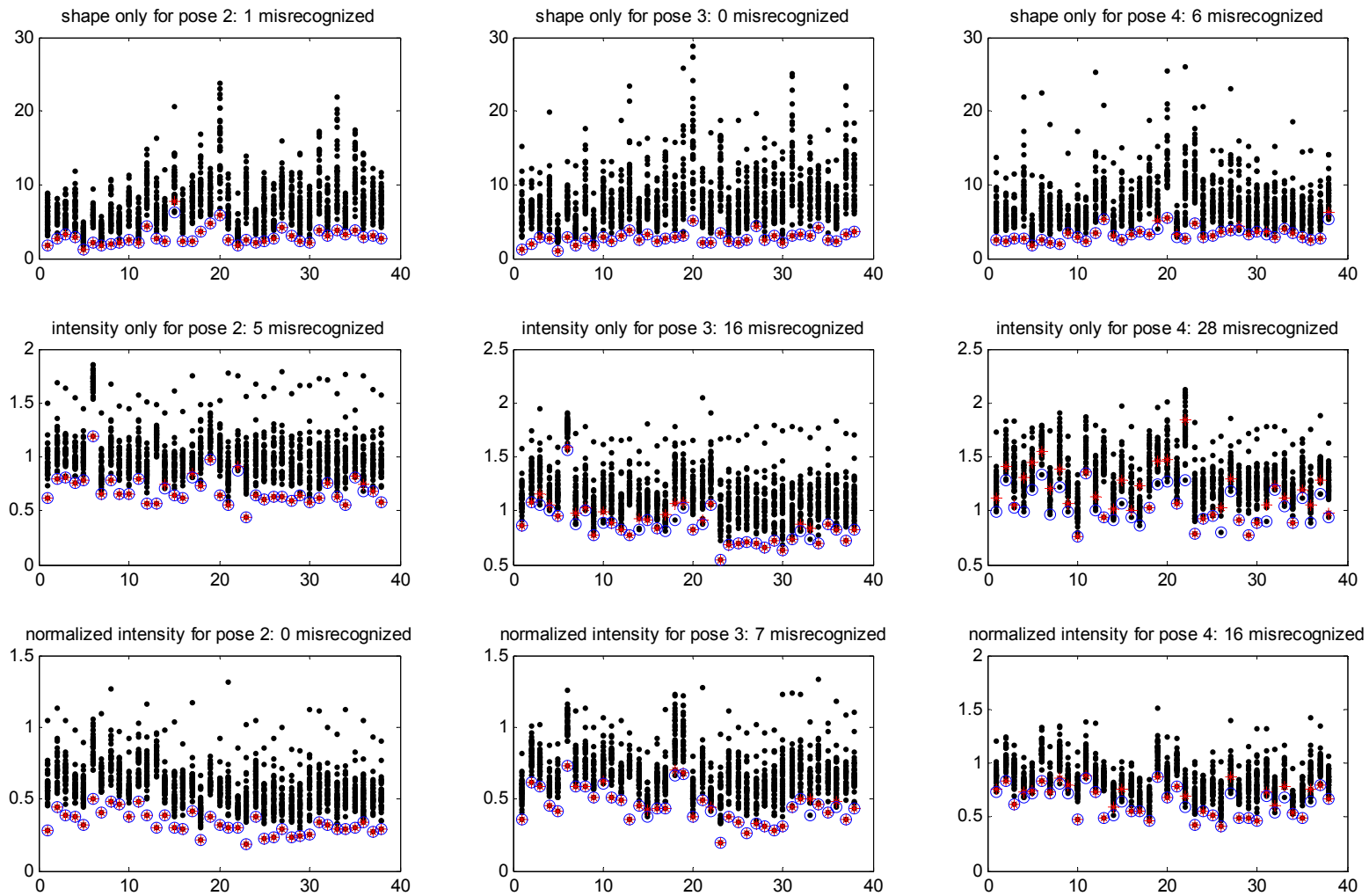


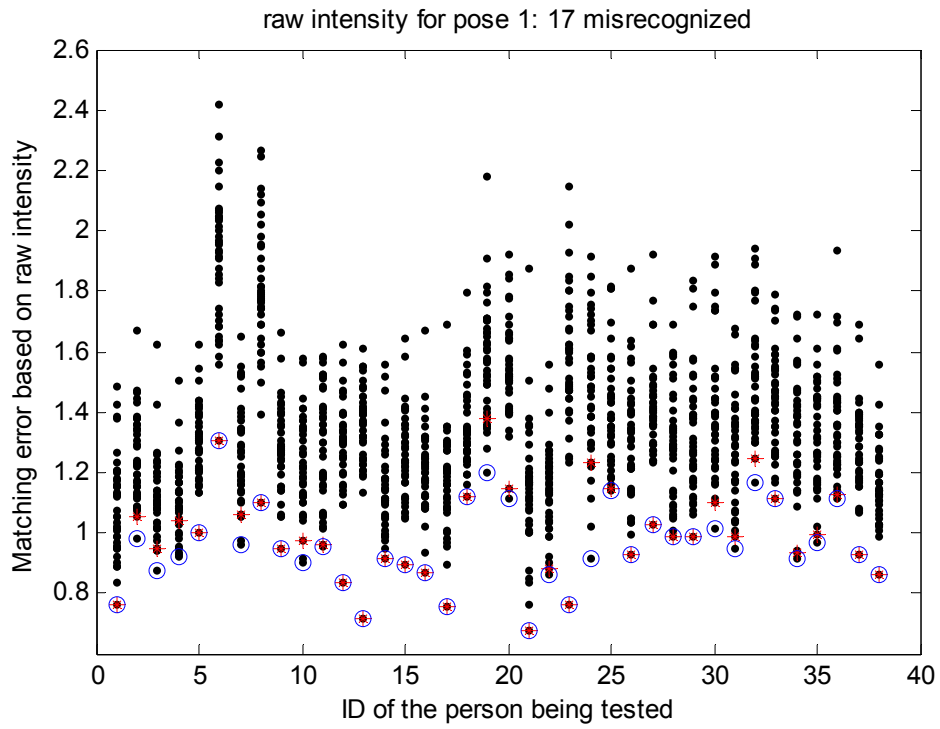
Figure 6.13 Recognition results for pose categories 2 to 4 (from the leftmost column to the last column)

Table 6.1 Misclassification rate for 38 people

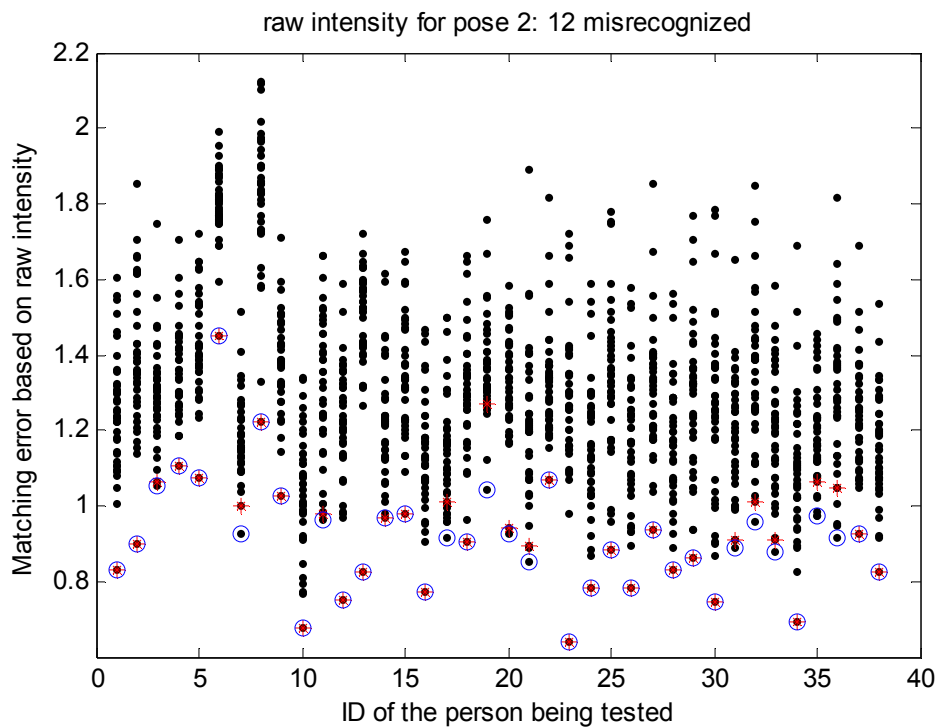
Pose\Method	Shape	Raw Texture	Texture	Normalized Texture	Combined
1	6	17	7	5	0
2	1	12	5	0	0
3	0	12	16	7	0
4	6	27	28	16	0

Table 6.1 clearly reveals better performance for the shape-based recognition than the texture-based recognition. As the face turns away from the camera (from frontal view to partial profile to complete profile), the texture-based recognition deteriorated quickly. For complete profile view, the number of misclassified people is as high as 28 for the texture-based algorithm and 16 for the normalized texture-based algorithm. It is obvious that after normalizing the illumination condition, the performance is significantly improved. Fusion of the shape error and texture error could improve the overall performance due to their complementary property.

Table 6.1 also lists the recognition rate for the raw texture-based algorithm. Different from the texture-based algorithm (and the normalized texture-based algorithm), the test face image is not morphed to the shape frame of the synthesized image for the raw texture-based algorithm. Instead, the test face image is clipped based on the convex hull of the synthesized face model. Figure 6.14 shows recognition results for four pose categories respectively.



(a)



(b)

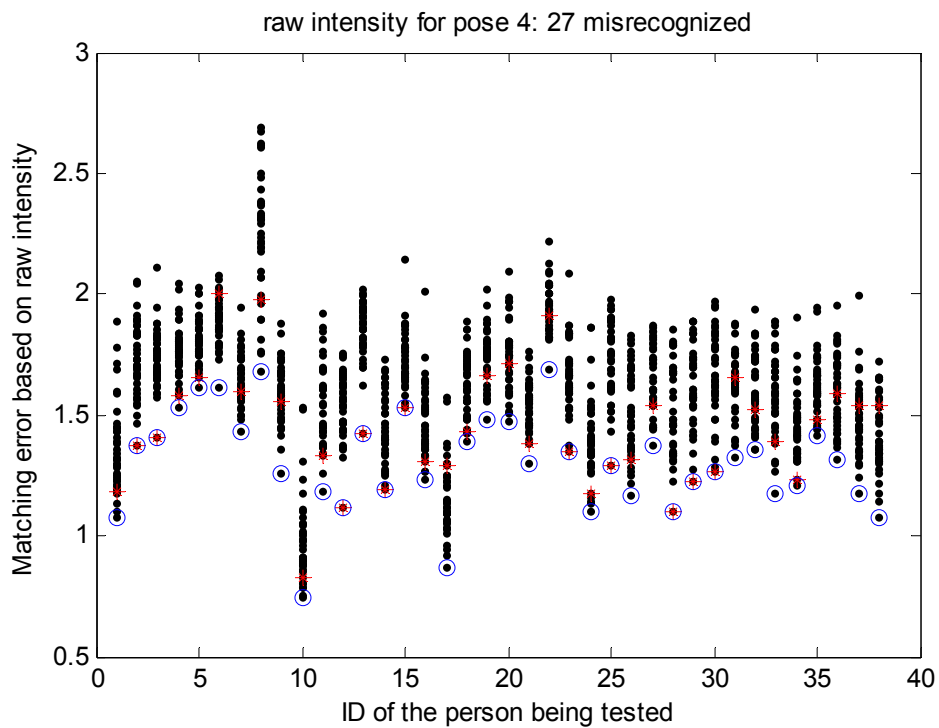
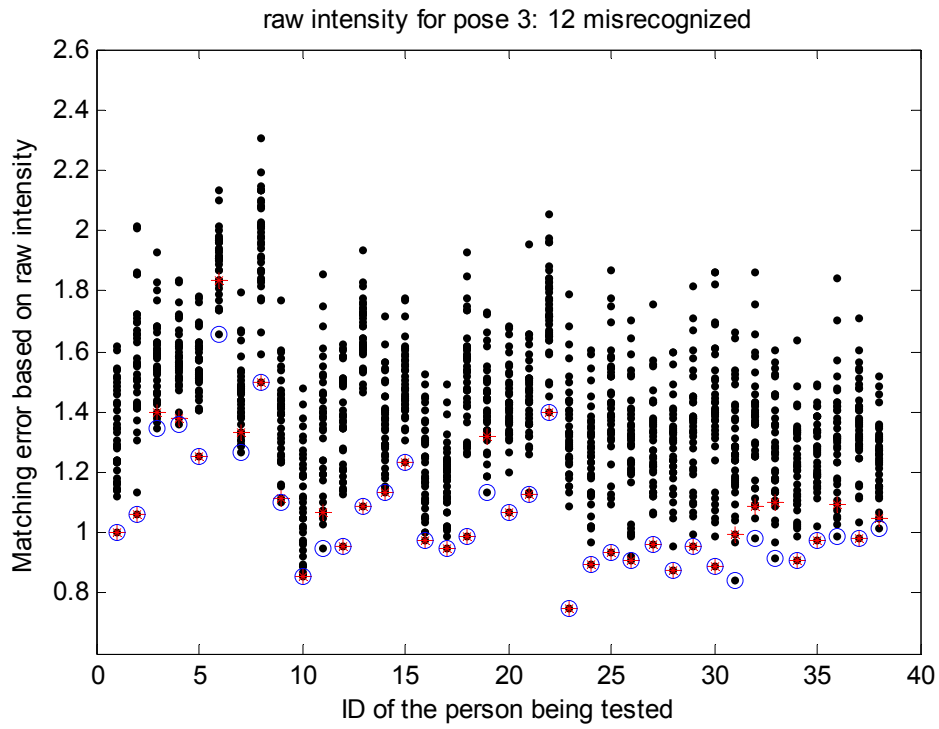


Figure 6.14 Recognition results based on raw texture error for pose categories 1 to 4

6.4 Conclusions

In this chapter, our 3D face model is explored for the purpose of face recognition. A small face database is carefully constructed to include four views that fall into the pose categories we defined. The recognition experiment is conducted for each pose category. When one pose is set to be the test pose, the 3D face model is constructed from the other three available poses during the training period. Given a test image, its feature point set is extracted and refined with the view-based AAM algorithm. When the face is assumed to be originated from a 3D face model, the distance map error, the texture error and the normalized texture error serves as the measurements as to how the test image fits to the 3D face model. Recognition experiments are conducted based on the shape, texture and normalized texture respectively. From our experiment, shape-based method has a better and stable performance throughout all poses. This is not contradictory to the claim [4] that appearance-based algorithms on average have better performance than geometry-based algorithms. Note that conclusion was made in 1993 and face shape descriptions have evolved from simple geometrical features like point and corner in 2D space to more complicated flexible shape model with the capability to model face from the perspective of 3D dimension. As expected, the illumination-normalized algorithm is better than the pure texture-based algorithm.

It is almost certain that the recognition performance could be greatly improved if more face images are available for each individual in the database. On the one hand, the reconstructed 3D model will be more reliable. On the other hand, when the pose space is densely sampled, it would be more likely to have a training view that has similar pose as the test image, which will make the recognition much easier.

CHAPTER 7 : CONCLUSIONS AND FUTURE WORK

This thesis addresses several important aspects of a face recognition system: face alignment and face feature extraction, 3D face surface reconstruction and facial illumination analysis.

The task of feature extraction is coupled with the face modeling task under the framework of the view-based AAM algorithm. Apparently the model-based analysis is more robust and accurate to extract feature points compared to traditional edge detection based methods. Sub-model analysis is adopted for better feature localization. A hybrid optimization scheme is presented to incorporate several shape constraints analytically to improve the face alignment accuracy. With explicit definition of feature points in the 3D face model, it is straightforward to reconstruct the positions of the 3D feature points from the corresponding extracted 2D feature points. The correspondence problem is relaxed with the distance mapping technique. Instead of recovering 3D feature point one by one, the 3D modeling is conducted by transforming a generic model with a cubic morphing function to match the projected feature points. The reconstructed 3D model is utilized to analyze the illumination condition from the images. The purpose is to synthesize novel images from arbitrary viewing angle and arbitrary illumination condition, as well as extract illumination-invariant texture map for face recognition task.

There are several possible directions to extend our work in the future.

7.1 Incorporating Texture Information to Extract Face Features

In our work, once the 2D feature points are extracted through 2D face modeling in the

view-based AAM algorithm, they are assumed to be accurate and fixed thereafter, which is not true in reality. The face alignment is only conducted within a specific face image. The correlation between different images of the same person is not investigated. The view-based models for different pose categories are mutually independent. In the introduction section of Chapter 3, some 3D AAM algorithms are briefly mentioned. A 3D AAM algorithm is a natural extension of 2D AAM algorithms. A 3D model is trained directly from available training images or video sequences. The update of the 3D model is driven by the modeling error with either a canonical optimization algorithm or a regression model. Inspired by existing 3D AAM algorithms, we tried a similar scheme to directly update our 3D face model. A 3D deformable face model is used to limit the searching space for 2D AAMs of different viewpoints. It is expected to have a better alignment performance than several independent view-based AAM models. However, not only it is very slow, but the convergence of the algorithm is very poor. One of the reasons might be that the constraint imposed by the 3D face model to several 2D models is relatively weak. As the 3D model still needs to be reconstructed from 2D models, extra reconstruction error is introduced, which results in deteriorated 2D modeling performance in return.

Nevertheless, it is still possible to make use of the correlation between different images of the same person to improve the accuracy of both 2D alignment and 3D structure estimation. One possible approach relies on the global texture map as introduced in Chapter 6. Let the global texture map be dynamically updated as the 2D modeling is carried out. [The global texture map fuses information from different viewpoints. When it is incorporated in the 2D face aligning process, a better alignment performance is expected.](#) One problem with this approach

is that an overwhelming amount of computation is involved.

7.2 Incorporating Illumination Information to Refine Surface Details

In the previous section, several possible approaches are discussed in order to improve the 2D 

feature extraction accuracy with the help of extra constraints from face images of different viewpoints. Feature points are defined along the boundary of facial parts where pixel intensity

varies dramatically. In the conclusion part of Chapter 5, it is briefly mentioned that we

initially aimed to refine our 3D face structure with some inference from facial illumination

conditions. A main initiative is the complementary nature of our model-based face alignment

algorithm and the traditional shape from shading algorithms. Typical SFS algorithms work

under the assumption that the object surface is uniform with one single albedo parameter, so

that the variation of image intensity is purely caused by the illumination setting. Chapter 5

also reviews some literatures about fusing SFS algorithm with stereo vision algorithm. It is


more challenging to combine our face structure estimation algorithm with shape from shading

algorithm since it is more difficult to establish correspondence between images from different

viewpoints than stereo images. We didn't go in that direction at last. Besides the reasons we

mentioned in the conclusion part of Chapter 5, we lack a face database where images are

taken under well-controlled illumination environment. Besides, current 3D model is sparse on

those textureless facial areas and it has little capability to describe those areas in detail. 

The modeling ability of the cubic morphing function we adopted is quite limited for a very

complicated surface. Currently, the symmetry property of face structure and texture is only

applied to average the normal field and texture map of the left and right half face. It remains

an open challenge to infer surface details from illumination or other useful information in the follow-up research. In the future, the CMU-PIE [80] face database should be used as our training and test purpose. The CMU-PIE database systematically samples a large number of pose and illumination conditions along with a variety of facial expressions. Besides, as a lot of algorithms report their performance on the CMU-PIE database, different algorithms could be compared and evaluated.

7.3 Tracking Faces

Our research work deals with still face images of different viewing angles. It is possible to extend our work to video sequences that contains one or several faces. Given an initial rough estimation of the face pose and structure in the video sequences, the 3D face morphing parameters and face pose parameters could be refined as more frames are considered. The tracking scheme is similar to the 3D AAM algorithm in [5], except that the cubic morphing method replaces the training of a linear shape subspace to model the variation of different 3D faces.

7.4 Facial Expression Modeling and Recognition

Facial expressions reflect one's internal emotion states and they are direct results of stretching and contracting of different facial muscles. Most facial recognition systems try to recognize the following six prototypic emotional expressions: disgust, fear, joy, surprise, sadness and anger. Usually facial expression analysis follows the step of facial data extraction and representation. In our scenario, face images are ready for expression analysis after the face is aligned and all feature points are assumed to be located. Early attempts in facial expression

recognition try to identify spots on an image sequence and analyze how their relative positions vary temporally. In recent years, the research trend is to employ more facial features to recognize more expressions and facial action units rather than emotion-specific expressions are recognized. There are several good review papers about facial expression recognition [81][82][83].

In Chapter 4, Candide-3 model is briefly introduced as one of the popular face models. Candide-3 has the ability to model internal-person variance as it has a set of animation units. Our model is not suitable for modeling different facial expressions. [However, it is possible to !\[\]\(d84e7ea36f695d92cb39ec32c307ac93_img.jpg\) model facial expressions by locally grouping facial feature points to several action units.](#) Cubic morphing should be more than enough to model facial parts separately. It would be interesting to extend our model to model and recognize different facial expressions.

List of References

1. M. Turk and A. Pentland, "Face Recognition Using Eigenfaces", Proceedings of IEEE Conf. On Computer Vision and Pattern Recognition, pp.586-591, 1991
2. L. Wiskott, J. M. Fellous, N. Kruger, "Face Recognition by Elastic Bunch Graph Matching", IEEE Trans. On PAMI, vol. 19, no. 7, July 1997
3. Samaria and S. Young, "HMM-based Architecture for Face Identification," Image and Vision Computing, vol.12, no.8, Oct. 1994
4. R. Berto, T. Poggio, "Face Recognition: Feature versus Templates", IEEE Trans. On PAMI, vol.15, no.10, pp. 1042-1052, Oct. 1993
5. V.Blanz and T. Vetter. "A Morphable Model for the Synthesis of 3D-faces." SIGGRAPH 99, 1999
6. R. Basri, D.W.Jacobs. "Lambertian Reflectances and Linear Subspaces." IEEE Trans. on PAMI. Vol. 25 no. 2, pp. 218-233, 2003
7. R.Ramamoorthi and P. Hanrahan. "On the Relationship between Radiance and Irradiance: determining the Illumination from Images of Convex Lambertian Object." Journal of the Optical Society of America, vol. 18, no. 10, pp 2448-2459, 2001
8. T. F. Cootes, G. J. Edwards, and C. J. Taylor. "Active Appearance Models." IEEE Trans. on PAMI, vol. 23, no. 6 pp. 681-685, June 2001
9. C. Huang, C. Chen, "Human Facial Feature Extraction for Face Interpretation and Recognition", Pattern Recognition, vol. 25, no. 12, p1435-1444, 1992
10. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes - Active Contour Models." International Journal of Computer Vision, vol. 1, no. 4, pp 321-331, 1987.
11. S. Osher and N. Paragios (eds), "Geometric Level Set Methods in Imaging, Vision and Graphics." Springer Verlag, New York, 2003
12. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. "Active Shape Models: Their Training and Application." CVGIP: Imaging Understanding, vol. 61, pp38-59, 1995

13. T. F. Cootes, G. Edwards and C. J. Taylor, "Comparing Active Shape Models with Active Appearance Models." The 10th British Machine Vision Conference, pp. 173-182, University of Nottingham, Sep. 1999
14. C. Hu, R. Feris, M. Turk, "Active Wavelet Networks for Face Alignment. British Machine Vision Conference, East Eaglia, Norwich, UK, 2003
15. S. Yan, C. Liu, S. Li, H. Zhang, H. Shum, Q. Cheng, "Texture-Constrained Active Shape Models." Proc. Of Int. W. on Generative-Model-Based Vision, Copenhagen, 2002
16. X. Hou, S. Li, H. Zhang, Q. Cheng, "Direct Appearance Models." CVPR'2001. pp. 828-933, 2001
17. I. Matthews, S. Baker, "Active Appearance Models Revisited." International Journal of Computer Vision vol. 60, 2004
18. M. Stegmann, "Active Appearance Models: Theory, Extensions and Cases." Master's thesis, Department of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark, 2000
19. T. F. Cootes, K. Walker, C. J. Taylor, "View-Based Active Appearance Models", Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 227, 2000
20. C. Hu, R. Feris, M. Turk, "Real-time View-based Face Alignment using Active Wavelet Networks." IEEE Internal Workshop on Analysis and Modeling of Faces and Gestures, pp. 215, 2003
21. S. Z. Li, H. Zhang, S. Yan, Q. Cheng. "Multi-View Face Alignment Using Direct Appearance Models," Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 0324, 2002
22. S. Romdhani, S. Gong, A. Psarrou, "A Multi-View Nonlinear Active Shape Model Using Kernel PCA." BMVC. 1999
23. Y. Li, S. Gong and H. Liddell. "Modeling Faces Dynamically Across Views and Over Time." In Proc. IEEE International Conference on Computer Vision, Vancouver, Canada, July 2001
24. F. Dornaika, J. Ahlberg. "Face Model Adaptation for Tracking and Active Appearance Model Training". BMVC. 2003
25. J. Xiao, S. Baker, I. Matthews, T. Kanade, "Real-Time Combined 2D+3D Active Appearance Models." CVPR. vol. 2, pp. 535-542, 2004

26. M. Roberts, T. Cootes, J. Adams, "Linking Sequences of Active Appearance Sub-models via Constraints: an Application in Automated Vertebral Morphometry." In the 14th British Machine Vision Conference. vol. 1, pp. 349–358, 2003
27. C. Zhang, F.S. Cohen, "3-D Face Structure Extraction and Recognition from Images using 3-D Morphing and Distance Mapping". IEEE Trans. on Image Processing, Volume 11, Issue 11, pp. 1249-1259, Nov. 2002
28. P. Phillips, H. Moon, P. Rauss, S. Rizvi, "The Feret Evaluation Methodology for Face Recognition Algorithms", Proceedings of IEEE Computer Vision and Pattern Recognition. pp.137–143, 1997
29. The psychological image collection at Stirling. (<http://pics.psych.stir.ac.uk/>)
30. M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, "Coding Facial Expressions with Gabor Wavelets. In: Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, pp. 200–205, 1998
31. M.J.Jones and T. Poggio. "Multidimensional Morphable Models", Proceedings Of the International Conference on Computer Vision, pp. 683-688, 1998
32. T. F. Cootes and P.Kittipanya-ngam. Comparing Variations on the Active Appearance Model Algorithm. Proceedings of the British Machine Vision Conference 2002, Cardiff, UK, September 2002
33. Xun Xu, Changshui Zhang, Thomas S. Huang. "Active Morphable Model: An Efficient Method for Face Analysis", Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, May 2004
34. S. Baker, R.Gross, and I. Matthews. "Lucas-Kanade 20 years on: A unifying framework: Part 2", Technical Report CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, 2004
35. Cyberware Laboratory, Inc, Monterey, California. 4020/RGB 3D Scanner with Color Digitizer, 1990
36. B.K.P.Horn, "Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View", Ph.D. thesis, Massachusetts Inst. of Technology, 1970
37. E. Rouy, A. Tourin, "A Viscosity Solutions Approach to Shape from Shading", SIAM J. Numerical Analysis, vol. 29, no.3, pp.867-884, 1992

-
38. K. Ikeuchi, B.K.P. Horn, "Numerical Shape from Shading and Occluding Boundaries", *Artificial Intelligence*, vol. 17, nos. 1-3, pp.141-184, 1981
 39. R.T. Frankot, R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 439-451, 1988
 40. C.H. Lee, A. Rosenfeld, "Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System", *Artificial Intelligence*, vol.26, pp. 125-143, 1985
 41. A.P. Pentland, "Local Shading Analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp.170-187, 1984
 42. P.S. Tsai, M. Shah, "Shape from Shading Using Linear Approximation", *Image and Vision Computing J.*, vol. 12, no. 8, pp. 487-498, 1994
 43. A.Pentland, "Shape Information from Shading: A Theory about Human Perception, " *Proc. Int'l Conf. Computer Vision*, pp.404-413, 1988
 44. R. Zhang, Ping-Sing Tsai, J.E. Cryer, M. Shah, "Shape from Shading: A Survey", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690-706, 1999
 45. R.Bajcsy, L. Lieberman, "Texture Gradient as a Depth Cue," *Computer Graphics and Image Processing*, vol. 5, pp. 52-67, 1976
 46. A.P. Witkin, "Recovering Surface Shape and Orientation from Texture, " *Artificial Intelligence*, vol. 17, pp. 17-45, 1981
 47. B. J. Super, A.C. Bovik, "Shape from Texture Using Local Spectral Moments, " *IEEE Trans. On PAMI*, vol. 17, no. 4, pp. 333-343, 1995
 48. J. Aloimonos, "Shape from Texture," *Biological Cybernetics*, vol. 58, pp. 345-360, 1988
 49. P. Fua, Y. Leclerc, "Object-centered Surface Reconstruction: Combining Multi-image Stereo Shading," *Image Understanding Workshop*, pp. 1097-1120, 1993
 50. P. Belhumeur, D. Kriegman, "What is the Set of Images of an Object under All Possible Lighting Conditions?" *International Journal of Computer Vision*, vol. 28, no. 3, pp. 245-260, 1998
 51. R. Basri, D.W. Jacobs, "Photometric Stereo with General Unknown Lighting," *IEEE Trans. On PAMI*, vol. 25, no. 2, pp.218-233, 2003

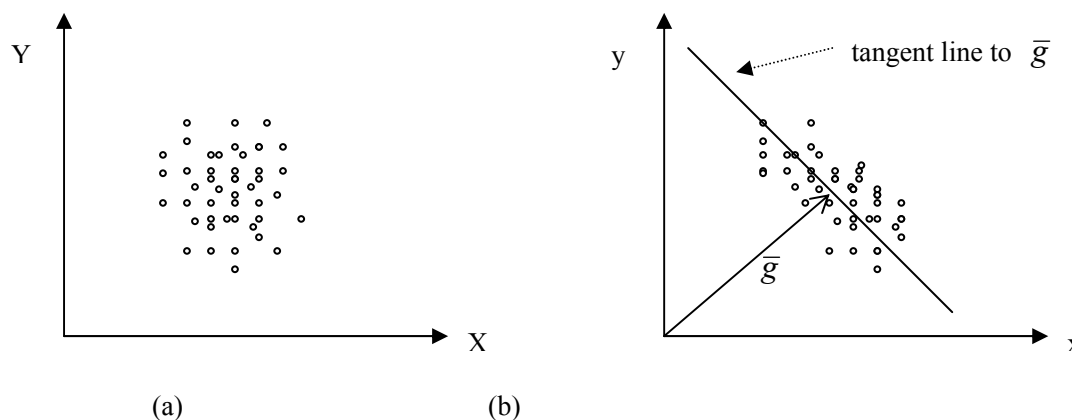
-
52. M. Rydfalk, "CANDICE, A Parametrized Face," Technical Report, Linkoping Univ., Dept. of Electrical Eng., S-581 83 Linkoping, Sweden, 1987
 53. H. Li, Roivainen, R. Forchheimer, "3D Motion Estimation in Mode-Based Facial Image Coding," IEEE Trans. On PAMI, vol 15, no. 6, pp. 545-556, 1993
 54. M. Kampmann, R. Farhoud, "Precise Face Model Adaptation for Semantic Coding of Videophone Sequences," Picture Coding Symposium, Berlin, German, Sept. 1997
 55. P. Kalra, A. Mangili, N.M. Thalmann, D. Thalmann, "Simulation of Facial Muscle Actions Based on Rational Free From Deformations," Eurographics, 1992
 56. <http://www.cs.washington.edu/research/projects/grail2/www/index.html>
 57. R. Koch, "Dynamic 3D Scene Analysis through Synthesis Feedback Control," IEEE Trans. On PAMI, VOL. 15, NO. 6, PP. 556-568, 1993
 58. F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. H. Salesin, "Synthesizing Realistic Facial Expressions from Photographs," Proceedings of SIGGRAPH98, Computer Graphics Proceedings, Annual Conference Series, Orlando, Florida, pp. 75-84, 1998
 59. M. W. Lee, S. Ranganath, "3D Deformable Face Model for Pose Determination and Face Synthesis," ICIAP, p. 260, 10th International Conference on Image Analysis and Processing (ICIAP'99), 1999
 60. F. Dornaika and J. Ahlberg, "Face Model Adaptation for Tracking and Active Appearance Model Training," 14th British Machine Vision Conference (BMVC), Norwich, UK, September 2003
 61. S. Romdhani, T. Vetter, "Efficient, Robust and Accurate Fitting of a 3D Morphable Model," IEEE International Conference on Computer Vision, 2003
 62. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, 1992
 63. D.T.Chen, A. State and D. Banks, "Interactive Shape Metamorphosis", 1995 Symposium on Interactive 3D Graphics, pp. 43-44, ACM SIGGRAPH, April 1995
 64. K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro and W. Stuetzle, "View-based Rendering: Visualizing Real Objects from Scanned Range and Color Data", Proceedings of 8th Eurographics Workshop on Rendering, pp. 23-34, St. Etienne, France, June, 1997
 65. P.Hallinan. "A Low-dimension Representation of Human Faces for arbitrary lighting conditions", IEEE Conf. on Computer Vision and Pattern Recognition, pp995-999, 1994

-
66. R. Epstein, P. Hallinan, and A. Yuille. "5+2 Eigenimages Suffice: an Empirical Investigation of Low-dimensional Lighting Models", IEEE Workshop on Physics-Based Vision", pp108-116, 1995
 67. A. Georghiades, P. Belhumeur, and D. Kriegman. "From Few to Many: Generative Models for Recognition under Variable Pose and Illumination", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 643-660, 2001
 68. R. Basri, D.W. Jacobs, "Lambertian Reflectances and Linear Subspaces", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 2, pp. 218-233, 2003
 69. R. Ramamoorthi, "Analytic PCA Construction for Theoretical Analysis of Lighting Variability in a Single Image of a Lambertian Object", IEEE Trans. on Pattern Analysis and Machine Intelligence", vol. 24, no. 10, pp. 1322-1333, 2002
 70. A. Georghiades, D. Kriegman, and P. Belhumeur, "Illumination Cones for Recognition under Variable Lighting: Faces", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 52-59, 1998
 71. Z. Wen, Z. Liu, T. S. Huang, "Face Relighting with Radiance Environment Maps", Proceedings of the 2003 IEEE Computer Society Conference on CVPR (CVPR03)
 72. Lee Nelson, "Commercialising Robust Face Recognition Capability, Polaroid & Quebec Vision Start-Up", Advance Imaging, pp. 72-73, Feb. 1998
 73. Miros: True face of security, <http://www.miros.com>
 74. R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and Machine Recognition of Faces: A Survey", Proceedings of the IEEE, vol. 83, no. 5, pp. 705-740, May 1995
 75. D. Beymer, "Face recognition under Varying Pose", Proc. IEEE Computer Vision and Pattern Recognition, pp. 756-761, Seattle, WA June, 1994
 76. D. Beymer, P. Mclauchlan, B. Coifman, and J. Malik, "A Real-time Computer Vision System for Measuring Traffic Parameters", Proc. CVPR 1997, San Juan, Puerto Rico, June 1997
 77. T. Vetter, "Synthesis of Novel Views from a Single Face Image", Max-Planck- Institut für biologische Kybernetik, Tübingen, Germany, Technical Report 26
 78. P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp 711-720, July 1997

-
79. M.S.Kamel, H. C. Shen, A. K. C. Wong, T. M. Hong, R. I. Campeanu, "Face Recognition using Perspective Invariant Features", *Pattern Recognition Letters*, pp. 877-883, September, 1994
 80. t. Sim, S.Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database", *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, pp 1615-1618, 2003
 81. b.Fasel and J. Luttin, "Automatic Facial EXPRESSION Analysis: Survey", *Pattern Recognition*, Vol. 36, No. 1, pp259-275, 2003
 82. M.Pantic and L.Rothkrantz, "Automatic Analysis of Facial Expressions: the State of the Art", *IEEE Trans. on PAMI*, Vol. 22, No. 12, pp 1424-1445, 2000
 83. A. Samal and P. Iyengar, "Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey", *Pattern Recognition*, Vol. 25, No. 1, pp 65-77, 1992
 84. T.Cootes and C. Taylor, "Statistical Models of Appearance for Computer Vision", Technical report, University of Manchester, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, Manchester M13 9PT, United Kingdom, Sep. 1999

APPENDIX A: AN EXISTING PROBLEM ABOUT THE TANGENT SPACE COORDINATE ALIGNMENT ALGORITHM

Before the shape vectors (or texture vectors) from the training set are subject to PCA for subspace construction, they need to be aligned to a common coordinate frame. T. Cootes demonstrates that the tangent space alignment method introduces less nonlinearity compared to other methods like the Procrustes analysis or norm one regulation. Each vector is transformed to the tangent space to the aligned mean so as to minimize the sum of distances between aligned samples and the mean. For shape vector, the transform is the similarity transform including scaling, rotating and tranpositioning. For texture vector, it is scaling and offsetting. The following figure gives a simplified illustration of tangent space alignment method. When the dimension of sample is 2, the tangent hyper plane becomes a tangent line as shown in the right figure.



A-1 (a) Original samples (b) Aligned samples

The tangent space passes through \bar{g} . Any vector g in the tangent space is normal to the set mean \bar{g} . They are related as:

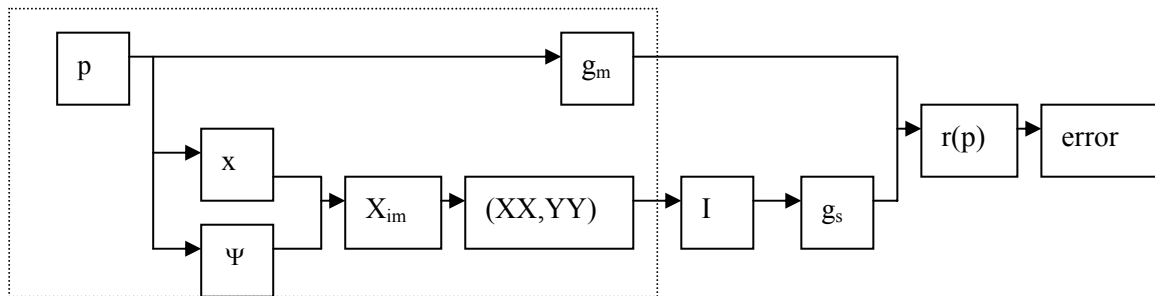
$$\bar{g} \cdot (g - \bar{g}) = 0 \quad (a.1)$$

Since \bar{g} is the aligned set mean, the aligning procedure needs to be done iteratively. Given

current \bar{g} , g is aligned by $g/\langle g, \bar{g} \rangle$. New \bar{g} is calculated as all samples are updated. Repeat the procedure till it converges. In our experiment, we noticed that there is a risk of not converging. This could be easily explained as a few of the samples are too distinguished and far away from the majority. When $\langle g, \bar{g} \rangle$ is close to zero, the aligned g will be extremely large and lead to an unstable estimate of \bar{g} . As a result, the procedure may not be able to converge. To guarantee its convergence, we better check to avoid the division-by-zero case and remove those singular samples from training set since they are not typical representatives.

APPENDIX B: DIRECT EXHAUSTIVE SEARCH FOR INITIAL MODEL PARAMETERS

The first order approximation of gradient descent matrix makes the face alignment task very sensitive to initial parameters. We develop a model based direct search strategy in order to find a suitable initial parameter set. The model parameter space is roughly sampled according to the parameter distributions learned from training set. Assume i th parameter has mean μ_i , standard deviation σ_i , 3 samples are taken at μ_i , $\mu_i + \sigma_i$ and $\mu_i - \sigma_i$ if this parameter plays an important role in the appearance interpretation. The parameters we select are respectively scale, transposition in x and y(4 samples are taken), together with the 2 parameters in the combined vector corresponding to the largest 2 eigenvalues. 5 samples are taken at μ_i , $\mu_i + \sigma_i$, $\mu_i - \sigma_i$, $\mu_i + 2\sigma_i$ and $\mu_i - 2\sigma_i$ for the 2 parameters from the combined vector. The 2 combined parameters encode 40% of the model appearance variance in the training set and allow intrinsic shape and texture variation in our search for initial parameter set. There are totally $3(\text{scale}) * 3(\text{transposition in x}) * 4(\text{transposition in y}) * 5 * 5 = 900$ combinations of model parameters. Matching all these candidate initializations with the test image sounds awkward, however we will show that the matching procedure could be speeded by doing some computation in advance. The following figure summarizes the steps of computing RMS error given model parameter vector p and test image I .



B-1 Flow chart of matching model parameters with test image

Clearly, all steps inside the dotted box could be computed in advance. For selected initial parameter set p , model texture vector g_m could be easily reconstructed. The shape vector in image frame X_{im} is decided by shape vector x in model frame and 2D pose parameter set Ψ . Warped face patch in the mean shape frame has interpolated coordinate pairs (XX,YY) in the test image and results in sample texture vector g_s . We save (XX,YY) and g_m into a table. When searching for optimal initial parameter set, each entry in the table yield a RMS error showing how it matches to the given test image. The search is very fast as only little computation is involved. This procedure could be enhanced by further sampling (random or even sampling) around each of the 342 choices. The performance of AAM is greatly improved with this initialization procedure.

APPENDIX C: 3D FACE MODEL NORMALIZATION

After a face model is transformed with cubic morphing, not only the face shape will vary, but the model might also be somehow scaled, rotated and shifted. It is necessary to regulate the morphing operation to avoid a biased estimation of the true pose parameters. This is done by aligning the morphed face to the generic face with a 3D similarity transform \mathbf{Q} . The morphed model is kept symmetric, so its center of gravity is already in the y-z plane (with the x component being zero). Then the number of unknown alignment variables is reduced from 7 to 6, including the scale factor s_a , the rotation angle θ_a and the translations (o_x, o_y, o_z) . Optimal parameters are desired so that the sum of squared distance is minimized. That is equation (4-14) as:

$$\arg \min_{s_a, \theta_a, o_x, o_y, o_z} \sum_i \left\| \mathbf{Q}(\mathbf{G}(\mathbf{P}_i^m)) - \mathbf{P}_i^m \right\|^2$$

Let $\mathbf{P}_i^m = (x_{0i}, y_{0i}, z_{0i})^T$, and $\mathbf{G}(\mathbf{P}_i^m) = (x_i, y_i, z_i)^T$, then the optimization becomes:

$$\begin{aligned} & \arg \min_{s_a, \theta_a, o_x, o_y, o_z} \sum_i \left\| s_a \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_a) & -\sin(\theta_a) \\ 0 & \sin(\theta_a) & \cos(\theta_a) \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} - \begin{bmatrix} x_{0i} \\ y_{0i} \\ z_{0i} \end{bmatrix} \right\|^2 \\ & = \arg \min_{s_a, \theta_a, o_x, o_y, o_z} \left(\sum_i \|s_a \cdot x_i + o_x - x_{0i}\|^2 + \sum_i \left\| s_a \cdot \begin{bmatrix} \cos(\theta_a) & -\sin(\theta_a) \\ \sin(\theta_a) & \cos(\theta_a) \end{bmatrix} \cdot \begin{bmatrix} y_i \\ z_i \end{bmatrix} + \begin{bmatrix} o_y \\ o_z \end{bmatrix} - \begin{bmatrix} y_{0i} \\ z_{0i} \end{bmatrix} \right\|^2 \right) \end{aligned}$$

(c.1)

The first item and the second item could be optimized separately. For the second item, it is basically the alignment of two shapes in 2D space. The solution to 2D alignment is given as the solution [84] to:

$$\begin{pmatrix} S_{yy} + S_{zz} & 0 & S_y & S_z \\ 0 & S_{yy} + S_{zz} & -S_z & S_y \\ S_y & -S_z & n & 0 \\ S_z & S_y & 0 & n \end{pmatrix} \begin{pmatrix} a \\ b \\ o_y \\ o_z \end{pmatrix} = \begin{pmatrix} S_{yy'} + S_{zz'} \\ S_{zy'} - S_{yz'} \\ S_{y'} \\ S_{z'} \end{pmatrix} \quad (c.2)$$

where

$$\begin{aligned} S_{yy} &= \sum y_i^2, & S_{zz} &= \sum z_i^2, & S_{zy'} &= \sum z_i \cdot y_{0i}, & S_{yz'} &= \sum y_i \cdot z_{0i} \\ S_{yy'} &= \sum y_i \cdot y_{i'}, & S_{zz'} &= \sum z_i \cdot z_{i'} \\ S_y &= \sum y_i, & S_z &= \sum z_i, & S_{y'} &= \sum y_{0i}, & S_{z'} &= \sum z_{0i} \end{aligned} \quad (c.3)$$

and variables a and b are related with the unknown scale factor s_a , the rotation angle θ_a as

$$a = s_a \cdot \cos(\theta_a), b = s_a \cdot \sin(\theta_a).$$

Variable o_x could be easily computed from the first item in equation (c.2).

Once the 3D similarity transformation matrix \mathbf{Q} is found, the old morphing parameters are updated by subjecting the morphed model to the 3D transformation to yield new morphing parameters. The pose parameters also need to be updated with the inverse transformation of \mathbf{Q} to cancel out the operation.

APPENDIX D: SYNTHESIZED FACE IMAGES BY VARYING THE FIRST SHAPE MODE



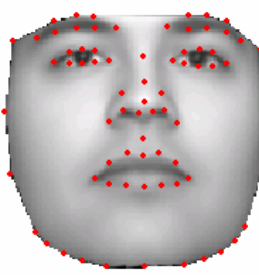
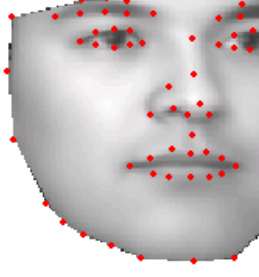
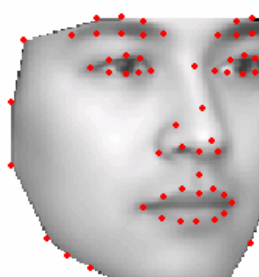
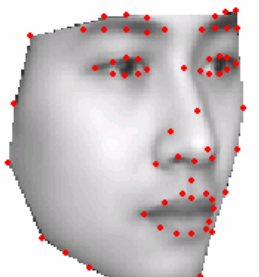
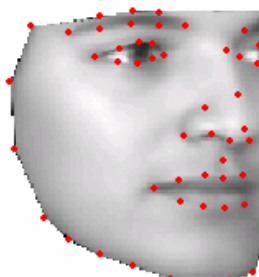
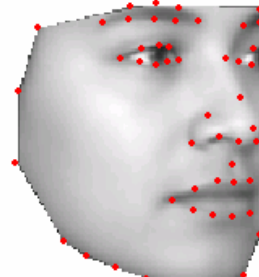

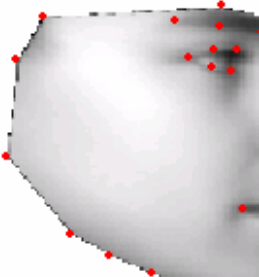
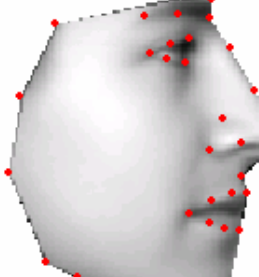
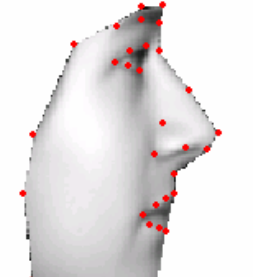
value	-3*standard deviation	0	3*standard deviation
Pose 1			
Pose 2			
Pose 3			
Pose 4			

Figure D-1 Face image by varying the principal shape mode

**APPENDIX E: SYNTHESIZED FACE IMAGES BY VARYING THE FIRST
TEXTURE MODE**


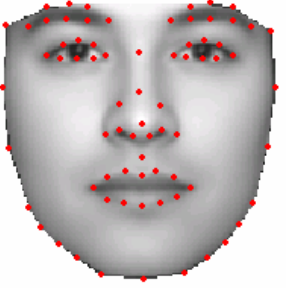
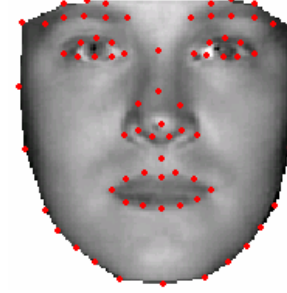
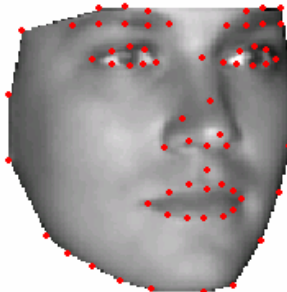
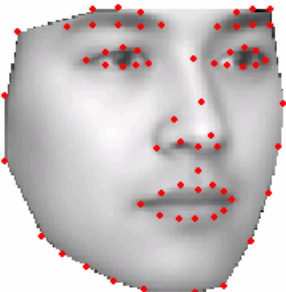
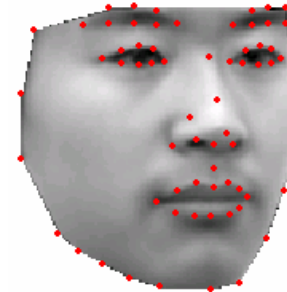
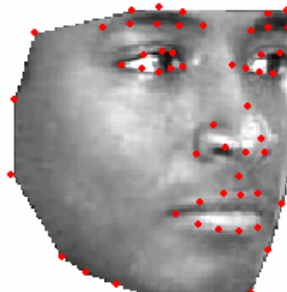
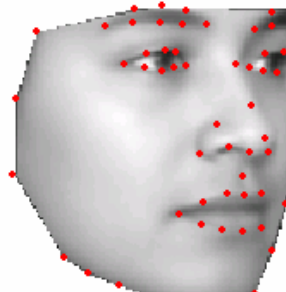
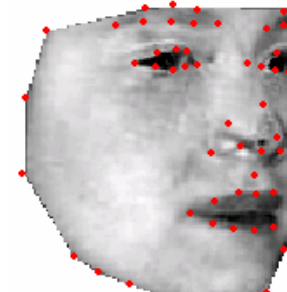
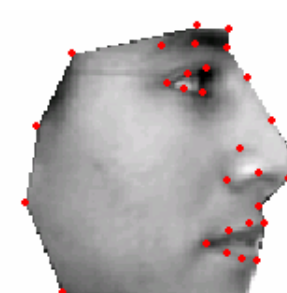
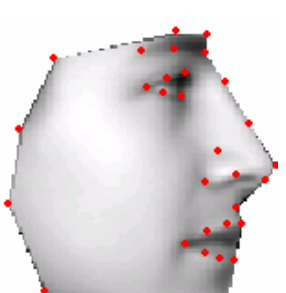
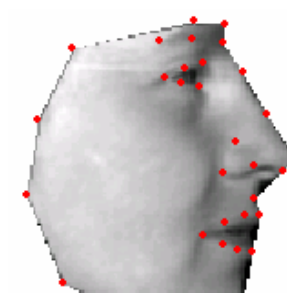
Value	-3*standard deviation	0	3*standard deviation
Pose 1			
Pose 2			
Pose 3			
Pose 3			

Figure E-1 Face image by varying the principal texture mode

VITA

Cuiping Zhang was born on October 27th, 1974 in Taixing, Jiangsu Province, China. She received her B.S. and M.S. in Electronics Engineering from Tsinghua University, Beijing, China, in 1997 and 2000 respectively. In September 2000, Cuiping entered the Ph.D. program in the Department of Electrical Engineering at Drexel University. She has been working on research in studying and developing 3D face reconstruction and recognition algorithms.