AN INTEGRATED FRAMEWORK ENHANCED WITH APPEARANCE MODEL FOR
FACIAL MOTION MODELING, ANALYSIS AND SYNTHESIS

BY

ZHEN WEN
B.Engr., Tsinghua University, 1998
M.S., University of Illinois at Urbana-Champaign, 2000

DISSEPTION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2004

Urbana, Illinois
Abstract

Human faces provide important cues of human activities. Thus they are useful for human-human communication, human-computer interaction (HCI) and intelligent video surveillance. Computational models for face analysis and synthesis are useful for both basic research and practical applications. In this dissertation, we present a unified framework for 3D face motion modeling, analysis and synthesis. We first derive a compact geometric facial motion model from motion capture data. Then it is used for robust 3D non-rigid face tracking and face animation. One limitation of the geometric model is that it can not handle the motion details, which are important for both human perception and computer analysis. Therefore, we enhance our framework with appearance models. To adapt the appearance model to different illumination conditions and different people, we propose the following methods: (1) modeling illumination effects from one single face image; (2) reducing person-dependency using ratio-image technique; and (3) online appearance model transformation during tracking. We demonstrate the efficacy of this framework by experimental results on face recognition, expression recognition and face synthesis in varying conditions. We will also show the use of this framework in applications such as computer-aided learning and very low bit-rate face video coding.
To my dear wife Xiaohui Gu.
Acknowledgments

I would like to thank Prof. Thomas S. Huang for the invaluable direction and feedback he has given me. He has provided the perfect balance of direction and freedom, allowing me to pursue my own ideas and supporting me the whole while. I would also like to thank my parents and my wife Xiaohui Gu, who have been supportive of my many years of education and the time and resources it has cost. we would like to thank the following people for discussions and collaborations: Dr. Pengyu Hong, Jilin Tu, Dr. Zicheng Liu and Dr. Zhengyou Zhang. I would thank Dr. Brian Guenter, Dr. Heung-Yeung Shum and Dr. Yong Rui of Microsoft Research for the face motion data. I would like to thank the other members of the IFP (Image Formation and Processing) group, who have provided help on numerous occasions. I am grateful to Sharon Collins and Gabriel Lopez-Walle for their great administrative support during my graduate study.

The work presented in this dissertation was supported by the National Science Foundation under contract number CDA 96-24386, and IIS-00-85980.
Table of Contents

List of Tables ........................................ ix

Chapter 1 Introduction .................................. 1
  1.1 Motivation ..................................... 1
  1.2 Overview ...................................... 2
    1.2.1 3D face processing framework overview .......... 2
    1.2.2 Geometric-based facial motion modeling, analysis and synthesis . . . 5
    1.2.3 Enhanced facial motion analysis and synthesis using flexible appearance model ......................................... 7
    1.2.4 Applications of face processing framework ............ 9
  1.3 Thesis Organization ................................ 10

Chapter 2 Related Work ................................. 12
  2.1 Geometric-based facial deformation modeling ......... 12
    2.1.1 Facial spatial deformation modeling ............... 12
    2.1.2 Facial temporal deformation modeling ............. 13
    2.1.3 Machine learning for facial deformation modeling ........... 14
  2.2 Geometric facial motion analysis ...................... 16
    2.2.1 Parameterized geometric models .................... 16
    2.2.2 FACS-based models ................................ 19
    2.2.3 Statistical models ................................ 19
  2.3 Geometric facial motion synthesis ..................... 20
    2.3.1 Performance-driven face animation .................. 20
    2.3.2 Text-driven face animation ......................... 21
    2.3.3 Speech-driven face animation ....................... 22
  2.4 Appearance-based facial motion modeling, analysis and synthesis ............ 24
  2.5 Hybrid facial motion modeling, analysis and synthesis ................ 25
    2.5.1 Issues in flexible appearance model ................. 26

Chapter 3 Learning Geometric 3D Facial Motion Model ............ 31
  3.1 Motion Capture Database .......................... 31
  3.2 Learning Holistic Linear Subspace .................... 32
  3.3 Learning Parts-based Linear Subspace .................. 34
  3.4 Animate Arbitrary Mesh Using MU .................... 38
  3.5 Temporal Facial Motion Model ........................ 41
## Contents

3.6 Summary ..................................... 42

Chapter 4 Geometric Model-based 3D face tracking .......................... 43
4.1 Geometric MU-based 3D Face Tracking ................................. 43
4.2 Applications of Geometric 3D Face Tracking ......................... 46
4.3 Summary ..................................... 48

Chapter 5 Geometric Facial Motion Synthesis ............................. 49
5.1 3D Face Modeling Tools in iFACE .................................. 49
  5.1.1 Generic face model ........................................... 50
  5.1.2 Personalized face model ..................................... 50
5.2 Text-driven Face Animation ......................................... 52
5.3 Offline Speech-driven Face Animation ............................... 55
5.4 Real-time Speech-driven Face Animation ............................ 56
  5.4.1 ANN-based real-time speech-driven face animation .......... 57
5.5 Summary ..................................... 64

Chapter 6 Flexible Appearance Model .................................. 66
6.1 Flexible Appearance Model ........................................... 67
  6.1.1 Reduce illumination dependency based on illumination modeling . 67
  6.1.2 Reduce person dependency based on ratio-image ................. 73
6.2 Summary ..................................... 75

Chapter 7 Facial Motion Analysis Using Flexible Appearance Model .... 77
7.1 Model-based 3D Face Motion Analysis Using Both Geometry and Appearance 78
  7.1.1 Feature extraction ............................................ 79
  7.1.2 Influences of lighting ....................................... 82
  7.1.3 Exemplar-based texture analysis .............................. 83
  7.1.4 Online EM-based adaptation .................................. 84
7.2 Facial Expression Recognition Experiment Results .................... 87
7.3 Enhance Geometric Tracking with Appearance Model ................. 94
  7.3.1 Problems with geometric tracking ............................ 95
  7.3.2 Proposed appearance-based enhancement ....................... 95
  7.3.3 Results ..................................................... 98
7.4 Summary ..................................... 100

Chapter 8 Face Appearance Synthesis Using Flexible Appearance Model 102
8.1 Neutral Face Relighting ............................................. 102
  8.1.1 Relighting with radiance environment maps ................. 103
  8.1.2 Face relighting from a single image ......................... 106
  8.1.3 Implementation .............................................. 109
  8.1.4 Relighting results ........................................... 111
8.2 Face Relighting For Face Recognition in Varying Lighting ........ 114
8.3 Synthesize Appearance Details of Facial Motion .................... 119
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3.1 Appearance of mouth interior</td>
<td>120</td>
</tr>
<tr>
<td>8.3.2 Linear alpha-blending of texture</td>
<td>120</td>
</tr>
<tr>
<td>8.4 Summary</td>
<td>122</td>
</tr>
<tr>
<td>Chapter 9 Application Examples of The Face Processing Framework</td>
<td>123</td>
</tr>
<tr>
<td>9.1 Model-based Very Low Bit-rate Face Video Coding</td>
<td>123</td>
</tr>
<tr>
<td>9.1.1 Introduction</td>
<td>123</td>
</tr>
<tr>
<td>9.1.2 Model-based face video coder</td>
<td>125</td>
</tr>
<tr>
<td>9.1.3 Results</td>
<td>126</td>
</tr>
<tr>
<td>9.1.4 Summary and future work</td>
<td>126</td>
</tr>
<tr>
<td>9.2 Integrated Proactive HCI environments</td>
<td>128</td>
</tr>
<tr>
<td>9.2.1 Overview</td>
<td>128</td>
</tr>
<tr>
<td>9.2.2 Current status</td>
<td>129</td>
</tr>
<tr>
<td>9.2.3 Future work</td>
<td>131</td>
</tr>
<tr>
<td>9.3 Summary</td>
<td>132</td>
</tr>
<tr>
<td>Chapter 10 Conclusion and Future Work</td>
<td>133</td>
</tr>
<tr>
<td>10.1 Conclusion</td>
<td>133</td>
</tr>
<tr>
<td>10.2 Future Work</td>
<td>134</td>
</tr>
<tr>
<td>10.2.1 Improve geometric face processing</td>
<td>134</td>
</tr>
<tr>
<td>10.2.2 Closer correlation between geometry and appearance</td>
<td>134</td>
</tr>
<tr>
<td>10.2.3 Human perception evaluation of synthesis</td>
<td>135</td>
</tr>
<tr>
<td>References</td>
<td>142</td>
</tr>
</tbody>
</table>
List of Tables

5.1 Phoneme and viseme used in face animation. ........................................ 54
5.2 Emotion inference based on video without audio track. ......................... 62
5.3 Emotion inference based on audio track. ........................................... 62
5.4 Emotion inference based on video with audio track 1. .......................... 63
5.5 Emotion inference based on video with audio track 2. .......................... 63
5.6 Emotion inference based on video with audio track 3. .......................... 63

7.1 Person-dependent confusion matrix using the geometric-feature-only method .................. 89
7.2 Person-dependent confusion matrix using both geometric and appearance fea-
tures ........................................................................................................... 89
7.3 Comparison of the proposed approach with geometric-only method in person-
dependent test together with their 95% confidence intervals. .................... 89
7.4 Comparison of the proposed appearance feature (ratio) with non-ratio-image based appearance feature (non-ratio) in person-independent recognition test. 90
7.5 Comparison of different algorithms in person-independent recognition test.  
(a): Algorithm uses geometric feature only. (b): Algorithm uses both geo-
metric and ratio-image based appearance feature. (c): Algorithm applies unconstrained adaptation. (d): Algorithm applies constrained adaptation. . . 92

9.1 Performance comparisons between the face video coder and H.264/JVT coder. 126
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Research issues and applications of face processing.</td>
</tr>
<tr>
<td>1.2</td>
<td>A unified 3D face processing framework.</td>
</tr>
<tr>
<td>3.1</td>
<td>An example of marker layout for MotionAnalysis system.</td>
</tr>
<tr>
<td>3.2</td>
<td>The markers of the Microsoft data [57]. (a): The markers are shown as small white dots. (b) and (c): The mesh is shown in two different viewpoints.</td>
</tr>
<tr>
<td>3.3</td>
<td>The neutral face and deformed face corresponding to the first four MUs. The top row is frontal view and the bottom row is side view.</td>
</tr>
<tr>
<td>3.4</td>
<td>(a): NMF learned parts overlayed on the generic face model. (b): The facial muscle distribution. (c): The aligned facial muscle distribution. (d): The parts overlayed on muscle distribution. (e): The final parts decomposition.</td>
</tr>
<tr>
<td>3.5</td>
<td>Three lower lips shapes deformed by three of the lower lips parts-based MUs respectively. The top row is the frontal view and the bottom row is the side view.</td>
</tr>
<tr>
<td>3.6</td>
<td>(a): The neutral face side view. (b): The face deformed by one right cheek parts-based MU.</td>
</tr>
<tr>
<td>3.7</td>
<td>(a): The generic model in iFACE. (b): A personalized face model based on the Cyberware™ scanner data. (c): The feature points defined on generic model.</td>
</tr>
<tr>
<td>4.1</td>
<td>Typical tracked frames and corresponding animated face models. (a): The input image frames. (b): The tracking results visualized by yellow mesh overlayed on input images. (c): The front views of the face model animated using tracking results. (d): The side views of the face model animated using tracking results. In each row, the first image corresponds to neutral face.</td>
</tr>
<tr>
<td>4.2</td>
<td>(a): The synthesized face motion. (b): The reconstructed video frame with synthesized face motion. (c): The reconstructed video frame using H.26L codec.</td>
</tr>
<tr>
<td>5.1</td>
<td>The generic face model. (a): Shown as wire-frame model. (b): Shown as shaded model.</td>
</tr>
<tr>
<td>5.2</td>
<td>An example of range scanner data. (a): Range map. (b): Texture map.</td>
</tr>
<tr>
<td>5.3</td>
<td>Feature points defined on texture map.</td>
</tr>
<tr>
<td>5.4</td>
<td>The model editor.</td>
</tr>
<tr>
<td>5.5</td>
<td>An example of customized face models.</td>
</tr>
<tr>
<td>5.6</td>
<td>The architecture of text driven talking face.</td>
</tr>
</tbody>
</table>
5.7 Four of the key shapes. The top row images are front views and the bottom row images are the side views. The largest components of variances are (a): 0.67; (b): 1.0; (c): 0.18; (d): 0.19.

5.8 The architecture of offline speech driven talking face.

5.9 Compare the estimated MUPs with the original MUPs. The content of the corresponding speech track is “A bird flew on lighthearted wing.”

5.10 Typical frames of the animation sequence of “A bird flew on lighthearted wing.” The temporal order is from left to right, and from top to bottom.

6.1 A face albedo map.

7.1 Hybrid 3D face motion analysis system.

7.2 (a): The input video frame. (b): The snapshot of the geometric tracking system. (c): The extracted texture map.

7.3 Selected facial regions for feature extraction.

7.4 Comparison of the proposed approach with geometric-only method in person-dependent test.

7.5 Comparison of the proposed appearance feature (ratio) with non-ratio-image based appearance feature (non-ratio) in person-independent recognition test.

7.6 Comparison of different algorithms in person-independent recognition test. (a): Algorithm uses geometric feature only. (b): Algorithm uses both geometric and ratio-image based appearance feature. (c): Algorithm applies unconstrained adaptation. (d): Algorithm applies constrained adaptation.

7.7 The results under different 3D poses. For both (a) and (b): Left: cropped input frame. Middle: extracted texture map. Right: recognized expression.

7.8 The results in a different lighting condition. For both (a) and (b): Left: cropped input frame. Middle: extracted texture map. Right: recognized expression.

7.9 The tracking results on a sub-sampled video of video.

7.10 The tracking results on a sub-sampled video with simultaneous 3D rotation and translation. Original video is from [22].

7.11 The tracking results on a sub-sampled video with pure 3D rotation. Original video is from [22].

8.1 Using constrained texture synthesis to reduce artifacts in the low dynamic range regions. (a): input image; (b): blue channel of (a) with very low dynamic range; (c): relighting without synthesis; and (d): relighting with constrained texture synthesis.

8.2 Detected outlier pixels with specular reflections (a): input image; (b): detected outliers.

8.3 Using constrained texture synthesis to reduce specular reflections: (a): relighting without synthesis; (b): relighting with constrained texture synthesis; (c): the pixel difference between (a) and (b). (The differences in low intensity areas are masked out).

8.4 (a): The generic mesh. (b): The feature points.
8.5 The user interface of the face relighting software. ............................... 112
8.6 The middle image is the input. The sequence shows synthesized results of
180° rotation of the lighting environment. ........................................ 113
8.7 The comparison of synthesized results and ground truth. The top row is the
ground truth. The bottom row is synthesized result, where the middle image
is the input. ............................................................................. 114
8.8 The middle image is the input. The sequence shows a 180° rotation of the
lighting environment. .................................................................. 115
8.9 Interactive lighting editing by modifying the spherical harmonics coefficients
of the radiance environment map. ................................................. 115
8.10 Relighting under different lighting. For both (a) and (b): Left: Face to be
relighted. Middle: target face. Right: result. ................................. 116
8.11 Examples of Yale face database B [52]. From left to right, they are images
from group 1 to group 5. ............................................................... 117
8.12 Relighted examples of Yale face database B [52]. From left to right, they are
images from group 1 to group 5. ..................................................... 118
8.13 Recognition error rate comparison of before relighting and after relighting on
the Yale face database. ................................................................. 119
8.14 Mapping visemes of (a) to (b). For (b), the first neutral image is the input,
the other images are synthesized. .................................................. 121

9.1 (a) The synthesized face motion. (b) The reconstructed video frame with syn-
thesized face motion. (c) The reconstructed video frame using H.26L codec. 127
9.2 The setting for the Wizard-of-Oz experiments. ..................................... 130
9.3 (a) The interface for the student. (b) The interface for the instructor. ..... 131
Chapter 1

Introduction

This thesis is concerned with the computational processing of 3D faces, with applications in Human Computer Interaction (HCI). It is a disciplinary research area overlapping with computer vision, computer graphics, machine learning and HCI. A unified framework is presented in this thesis for 3D face processing, including facial motion modeling, analysis and synthesis. In this chapter, we first discuss the motivation for 3D face processing research and then give overviews of our 3D face processing research. Finally, we outline the organization of this thesis in Section 1.3.

1.1 Motivation

Human face provides important visual cues for effective face-to-face human-human communication. In human-computer interaction (HCI) and distant human-human interaction, computer can use face processing techniques to estimate users’ states information, based on face cues extracted from video sensor. Such states information is useful for the computer to proactively initiate appropriate actions. On the other hand, graphics based face animation provides an effective solution for delivering and displaying multimedia information related to human face. Therefore, the advance in the computational model of faces would make human computer interaction more effective. Examples of the applications that may benefit from face processing techniques include: visual telecommunication [2, 95], virtual environments [81],
and talking head representation of agents [142, 105]. Recently, security related issues have become major concerns in both research and application domains. Video surveillance has become increasingly critical to ensuring security. Intelligent video surveillance, which uses automatic visual analysis techniques, can relieve human operators from the labor-intensive monitoring tasks [60]. It would also enhance the system capabilities for prevention and investigation of suspicious behaviors. One important group of automatic visual analysis techniques are face processing techniques, such as face detection, tracking and recognition.

There have been a significant amount of previous research on various aspects of face processing, such as face modeling, face recognition, facial motion analysis and synthesis etc. A unified framework, however, is needed to deal with these different aspects under diverse conditions in a systematic way. For example, there are mainly two types of approaches for facial motion analysis and synthesis: geometric-based models and appearance-based models [3, 10, 38, 41, 45, 86, 129, 131]. Comparing these two approaches, geometric facial motion models are more robust to environmental changes but can not deal with detailed appearance variations. These details are important visual cues. In contrast, appearance-based models are able to handle appearance details but sensitive to imaging conditions such as illuminations. Therefore, it is desired that a unified framework can combine the advantages of both approaches to work in diverse conditions.

1.2 Overview

1.2.1 3D face processing framework overview

In the field of face processing, there are two research directions: analysis and synthesis. Research issues and their applications are illustrated in Figure 1.1. For analysis, firstly face needs to be located in input video. Then, the face image can be used to identify who the person is. The face motion in the video can also be tracked. The estimated motion parameters can be used for user monitoring or emotion recognition. Besides, the face motion
Figure 1.1: Research issues and applications of face processing.

can also be used to as visual features in audio-visual speech recognition, which has higher recognition rate than audio-only recognition in noisy environments. The face motion analysis and synthesis is an important issue of the framework. In this thesis, the motions include both rigid and non-rigid motions. Our main focus is the non-rigid motions such as the motions caused by speech or expressions, which are more complex and challenging. We use “facial deformation model” or “facial motion model” to refer to non-rigid motion model, if without other clarification.

The other research direction is synthesis. First, the geometry of neutral face is modeled from measurement of faces, such as 3D range scanner data or images. Then, the 3D face model is deformed according to facial deformation model to produce animation. The anima-
tion may be used as avatar-based interface for human computer interaction. One particular application is model-based face video coding. The idea is to analyze face video and only transmit a few motion parameters, and maybe some residual. Then the receiver can synthesize corresponding face appearance based on the motion parameters. This scheme can achieve better visual quality under very low bit-rate.

In this thesis, we present a 3D face processing framework for both analysis and synthesis [145, 148, 146, 144]. The framework is illustrated in Figure 1.2. Due to the complexity of facial motion, we first collect 3D facial motion data using motion capture devices. Then subspace learning method is applied to derive a few basis. We call these basis Geometric Motion Units, or simply MUs. Any facial shapes can be approximated by a linear combination of the Motion Units. In face motion analysis, the MU subspace can be used to constrain noisy 2D image motion for more robust estimation. In face animation, MUs can be used to reconstruct facial shapes. The MUs, however, are only able to model geometric facial motion because appearance details are usually missing in motion capture data. These appearance details caused by motion are important for both human perception and computer analysis. To handle the motion details, we incorporate appearance model in the framework. We have focused on the problem of how to make the appearance model more flexible so that it can be used in various conditions. For this purpose, we have developed efficient methods for modeling the illumination effects and reduce the person-dependency of the appearance model. To evaluate face motion analysis, we have done facial expression recognition experiments to show that the flexible appearance model improve the results under varying conditions. We shall also present synthesis examples using the flexible appearance model.
1.2.2 Geometric-based facial motion modeling, analysis and synthesis

Accurate face motion analysis and realistic face animation demands good model of the temporal and spatial facial deformation. One type of approaches use geometric-based models [10, 38, 45, 129, 131]. Geometric facial motion model describes the macrostructure level face geometry deformation. The deformation of 3D face surfaces can be represented using the displacement vectors of face surface points (i.e. vertices). In free-form interpolation models [63, 129], displacement vectors of certain points are predefined using interactive editing tools. The displacement vectors of the remaining face points are generated using interpolation functions, such as affine functions, radial basis functions (RBF), and Bezier
volume. In physics-based models [143], the face vertices displacements are generated by
dynamics equations. The parameters of these dynamic equations are manually tuned. To
obtain a higher level of abstraction of facial motions which may facilitate semantic analysis,
psychologists have proposed Facial Action Coding System (FACS) [43]. FACS is based on
anatomical studies on facial muscular activity and it enumerates all Action Units (AUs) of
a face that cause facial movements. Currently, FACS is widely used as the underlying visual
representation for facial motion analysis, coding, and animation. The Action Units, however,
lack quantitative definition and temporal description. Therefore, computer scientists usually
need to decide their own definition in their computational models of AUs [129]. Because of
the high complexity of natural non-rigid facial motion, these models usually need extensive
manual adjustments to achieve realistic results.

Recently, there have been considerable advances in motion capture technology. It is now
possible to collect large amount of real human motion data. For example, the MotionAnalysis™
system [100] uses multiple high speed cameras to track 3D movement of reflective markers.
The motion data can be used in movies, video game, industrial measurement, and research
in movement analysis. Because of the increasingly available motion capture data, people
begin to apply machine learning techniques to learn motion model from the data. This type
of models would capture the characteristics of real human motion. One example is the linear
subspace models of facial motion learned in [76, 64, 121]. In these models, arbitrary face
deformation can be approximated by a linear combination of the learn basis.

In this thesis, we present our 3D facial deformation models derived from motion capture
data. Principal component analysis (PCA) [71] is applied to extract a few basis whose
linear combinations explain the major variations in the motion capture data. We call these
basis Motion Units (MUs), in a similar spirit to AUs. Compared to AUs, MUs are derived
automatically from motion capture data such that it avoids the labor-intensive manual work
for designing AUs. Moreover, MUs has smaller reconstruction error than AUs when linear
combinations are used to approximate arbitrary facial shapes. Based on MUs, we have
developed a 3D non-rigid face tracking system. The subspace spanned by MUs is used to constrain the noisy image motion estimation, such as optical flow. As a result, the estimated non-rigid can be more robust. We demonstrate the efficacy of the tracking system in model-based very low bit-rate face video coding. The linear combinations of MUs can also be used to deform 3D face surface for face animations. In iFACE system, we have developed text-driven face animation and speech-driven animations. Both of them use MUs as the underlying representation of face deformation. One particular type of animation is real-time speech-driven face animation, which is useful for real-time two-way communications such as teleconferencing. We have used MUs as the visual representation to learn a audio-to-visual mapping. The mapping has a delay of only 100 ms, which will not interfere with real-time two-way communications.

1.2.3 Enhanced facial motion analysis and synthesis using flexible appearance model

Besides the geometric deformations modeled from motion capture data, facial motions also exhibit detailed appearance changes such as wrinkles and creases as well. These details are important visual cues but they are difficult to analyze and synthesize using geometric-based approaches. Appearance-based models have been adopted to deal with this problem [3, 41]. Previous appearance-based approaches were mostly based on extensive training appearance examples. However, the space of all face appearance is huge, affected by the variations across different head poses, individuals, lighting, expressions, speech and etc. Thus it is difficult for appearance-based methods to collect enough face appearance data and train a model that works robustly in many different scenarios. In this respect, the geometric-feature-based methods are more robust to large head motions, changes of lighting and are less person-dependent.

To combine the advantages of both approaches, people have been investigating methods
of using both geometry (shape) and appearance (texture) in face analysis and synthesis. The Active Appearance Model (AAM) [33] and its variants, apply PCA to model both the shape variations of image patches and their texture variations. They have been shown to be powerful tools for face alignment, recognition, and synthesis. Blanz and Vetter [14] propose 3D morphable models for 3D faces modeling, which model the variations of both 3D face shape and texture using PCA. The 3D morphable models have been shown effective in 3D face animation and face recognition from non-frontal views [13]. In facial expression classification, Tian et al. [133] and Zhang et al. [158] propose to train classifiers (e.g. neural networks) using both shape and texture features. The trained classifiers were shown to outperform classifiers using shape or texture features only. In these approaches, some variations of texture are absorbed by shape variation models. However, the potential texture space can still be huge because many other variations are not modelled by shape model. Moreover, little has been done to adapt the learned models to new conditions. As a result, the application of these methods are limited to conditions similar to those of training data.

In this thesis, we propose a flexible appearance model in our framework to deal with detailed facial motions. We have developed an efficient method for modeling illumination effects from a single face image. We also apply ratio-image technique [86] to reduce person-dependency in a principled way. Using these two techniques, we design novel appearance features and use them in facial motion analysis. In a facial expression experiment using CMU Cohn-Kanade database [72], we show that the novel appearance features can deal with motion details in a less illumination dependent and person-dependent way [146]. In face synthesis, the flexible appearance model enables us to transfer motion details and lighting effects from one person to another [148]. Therefore, the appearance model constructed in one conditions can be extended to other conditions. Synthesis examples show the effectiveness of the approach.
1.2.4 Applications of face processing framework

3D face processing techniques have many applications ranging from intelligent human computer interaction to smart video surveillance. In this thesis, besides face processing techniques we will discuss applications of our 3D face processing framework to demonstrate the effectiveness of the framework.

The first application is model-based very low bit-rate face video coding. Nowadays Internet has become an important part of people’s daily life. In the current highly heterogeneous network environments, a wide range of bandwidth is possible. Provisioning for good video quality at very low bit rates is an important yet challenging problem. One alternative approach to the traditional waveform-based video coding techniques is the model-based coding approach. In the emerging Motion Picture Experts Group 4 (MPEG-4) standard, a model-based coding standard has been established for face video. The idea is to create a 3D face model and encode the variations of the video as parameters of the 3D model. Initially the sender sends the model to the receiver. After that, the sender extracts the motion parameters of the face model in the incoming face video. These motion parameters can be transmitted to the receiver under very low bit-rate. Then the receiver can synthesize corresponding face animation using the motion parameters. However, in most existing approaches following the MPEG-4 face animation standard, the residual is not sent so that the synthesized face image could be very different from the original image. In this thesis, we propose a hybrid approach to solve this problem. On one hand, we use our 3D face tracking to extract motion parameters for model-based video coding. On the other hand, we use the waveform-based video coder to encode the residual and background. In this way, the difference between the reconstructed frame and the original frame is bounded and can be controlled. The experimental results show that our hybrid deliver better performance under very low bit-rate than the state-of-the-art waveform-based video codec.

The second application is to use face processing techniques in an integrated human com-
puter interaction environment. In this project the goal is to contribute to the development of a human-computer interaction environment in which the computer detects and tracks the user’s emotional, motivational, cognitive and task states, and initiates communications based on this knowledge, rather than simply responding to user commands. In this environment, the test-bed is to teach school kids scientific principles via LEGO games. In this learning task, the kids are taught to put gears together so that they can learn principles about ratio and forces. In this HCI environment, we use face tracking and facial expression techniques to estimate the users’ states. Moreover, we use animated 3D synthetic face as avatar to interact with the kids. In this thesis, we describe the experiment we have done so far and the lessons we have learned in this process.

1.3 Thesis Organization

The remainder of the thesis is organized as follows. In the next chapter, we first give a review on the work in 3D face motion modeling, analysis and synthesis. Chapter 3 introduces our 3D facial motion database and the derivation of the geometric motion model. In Chapter 4, we describe how to use the derived geometric facial motion model to achieve robust 3D non-rigid face tracking. We will present experimental results in a model-based very low bit-rate face video coding application. We shall present the facial motion synthesis using the learned geometric motion model in Chapter 5. Three types of animation are described: (1) text-driven face animation; (2) offline speech-driven animation; and (3) real-time speech driven animation. Chapter 6 presents our flexible appearance model for dealing with motion details in our face processing framework. An efficient method is proposed to model illumination effects from a single face image. The illumination model helps reduce the illumination dependency of the appearance model. We also present ratio-image based techniques to reduce person-dependency of our appearance model. In Chapter 7 and Chapter 8, we describe our works on coping with appearance details in analysis and synthesis based on the flexible
appearance model. Experimental results on facial expression recognition and face synthesis in varying conditions are presented to demonstrate the effectiveness of the flexible appearance model. Finally, the thesis is concluded with summary and comments on future research directions.
Chapter 2
Related Work

In this chapter, we review existing research work related to our 3D face processing research. We first review works on geometric facial deformation\(^1\) modelling in Section 2.1. Next we introduce previous works on geometric facial motion analysis and synthesis in Section 2.2 and Section 2.3, respectively. Then, we review appearance-based approaches in Section 2.4. Finally, we give a review of approaches that combine geometric-based and appearance-based approaches, and discuss issues of flexible appearance model in Section 2.5.

2.1 Geometric-based facial deformation modeling

Since the pioneering work of Parke [106] in the early 70’s, many techniques have been investigated to model geometric facial deformation for 3D face tracking and animation. A good survey can be found in [108]. The key issues include (1) how to model the spatial and temporal facial surface deformation, and (2) how to apply these models for facial deformation analysis and synthesis. In this section, we introduce previous research on facial deformation.

\(^1\)In this thesis, the term “facial motion” and “facial deformation” both refer to non-rigid motion of face surface.
2.1.1 Facial spatial deformation modeling

In the past several decades, many models have been proposed to deform 3D facial surface spatially. Representative models include free-form interpolation models [63, 129], parameterized models [107], physics-based models [143], and more recently machine-learning-based models [76, 64, 121]. Free-form interpolation models define a set of points as control points, and then use the displacement of the control points to interpolate the movements of any facial surface points. Popular interpolation functions includes: affine functions [63], Splines, radial basis functions, Bezier volume model [129] and others. Parameterized models (such as Parke’s model [107] and its descendants) use facial feature based parameters for customized interpolation functions. Physics-based muscle models [143] use dynamics equations to model facial muscles. The face deformation can then be determined by solving those equations. Because of the high complexity of natural facial motion, these models usually need extensive manual adjustments to achieve plausible facial deformation. To approximate the space of facial deformation, people proposed linear subspaces based on Facial Action Coding System (FACS) [45, 129]. FACS [43] describes arbitrary facial deformation as a combination of Action Units (AUs) of a face. Because AUs are only defined qualitatively, and do not contain temporal information, they are usually manually customized for computation. Brand [16] used low-level image motion to learn a linear subspace model from raw video. However, the estimated low-level image motion is noisy such that the derived model is less realistic. With the recent advance in motion capture technology, it is now possible to collect large amount of real human motion data. Thus, people turn to apply machine learning techniques to learn model from motion capture data, which would capture the characteristics of real human motion. Some examples of this type of approaches are discussed in Section 2.1.3.
2.1.2 Facial temporal deformation modeling

For face animation and tracking, temporal facial deformation also needs to be modeled. Temporal facial deformation model describes the temporal trajectory of facial deformation, given constraints at certain time instances. Waters and Levergood [141] used sinusoidal interpolation scheme for temporal modeling. Pelachaud et al. [109], Cohen and Massaro [29] customized co-articulation functions based on prior knowledge, to model the temporal trajectory between given key shapes. Physics-based methods solve dynamics equations for these trajectories.

Recently, statistical methods have been applied in facial temporal deformation modeling. Hidden Markov Models (HMM) trained from motion capture data are shown to be useful to capture the dynamics of natural facial deformation [15]. Ezzat et al. [46] pose the trajectory modeling problem as a regularization problem [140]. The goal is to synthesize a trajectory which minimizes an objective function consisting of a target term and a smoothness term. The target term is a distance function between the trajectory and the given key shapes. The optimization of the objective function yields multivariate additive quintic splines [140]. The results produced by this approach could look under-articulated. To solve this problem, gradient descent learning [9] is employed to adjust the mean and covariances. In the learning process, the goal is to reduce the difference between the synthesized trajectories and the trajectories in the training data. Experimental results show that the learning improves the articulation.

2.1.3 Machine learning techniques for facial deformation modeling

In recent years, more available facial motion capture data enables researchers to learn models which capture the characteristics of real facial deformation.

Artificial Neural Network (ANN) is a powerful tool to approximate functions. It has
been used to approximate the functional relationship between motion capture data and the parameters of pre-defined facial deformation models. Morishima et al. [98] used ANN to learn a function, which maps 2D marker movements to the parameters of a physics-based 3D face deformation model. This helped to automate the construction of physics-based face muscle model, and to improve the animation produced. Moreover, ANN has been used to learn the correlation between facial deformation and other related signals. For example, ANN is used to map speech to face animation [78, 99, 92].

Principal Component Analysis (PCA) [71] learns orthogonal components that explain the maximum amount of variance in a given data set. Because facial deformation is complex yet structured, PCA has been applied to learn a compact low dimensional linear subspace representation of 3D face deformation [64, 76, 121]. Then, arbitrary complex face deformation can be approximated by a linear combination of just a few basis vectors. Besides animation, the low dimensional linear subspace can be used to constrain noisy low-level motion estimation to achieve more robust 3D facial motion analysis [64, 121]. Furthermore, facial deformation is known to be localized. To learn a localized subspace representation of facial deformation, Non-negative Matrix Factorization (NMF) [79] could be used. It has been shown that NMF and its variants are effective to learn parts-based face image components, which outperform PCA in face recognition when there are occlusions [84]. In this chapter, we describe how NMF may help to learn a parts-based facial deformation model. The advantage of a parts-based model is its flexibility in local facial motion analysis and synthesis.

The dynamics of facial motion is complex so that it is difficult to model with dynamics equations. Data-driven model, such as Hidden Markov Model (HMM) [115], provides an effective alternative. One example is “voice puppetry” [15], where an HMM trained by entropy minimization is used to model the dynamics of facial motion during speech. Then, the HMM model is used to off-line generate a smooth facial deformation trajectory given speech signal.
2.2 Geometric facial motion analysis

Analysis of human facial motion is the key component for many applications, such as model-based very low bit rate video coding for visual telecommunication [2], audio/visual speech recognition [127], expression recognition [3]. A large amount of work has been done on facial motion tracking. Simple approaches only utilize low-level image features. Although their computation complexity is low, they are not robust enough. For example, Goto et al. [54] extract edge information to find salient facial feature regions (e.g. eyes, lips, etc.). The extracted low-level image features are compared with templates to estimate the shapes of facial features. However, it is not robust enough to use low-level image features alone. The error will be accumulated with the increase in number of frames being tracked. High-level knowledge of facial deformation must be used to handle error accumulation problem by imposing constraints on the possible deformed facial shapes. It has been shown that robust tracking algorithm needs to integrate low-level image information and high-level knowledge. Examples of high-level constraints include: (1) parameterized geometric models such as B-Spline curves [23], snake model [73], deformable template [153], and 3D parameterized model [39]; (2) FACS-based models [45, 129]; and (3) statistical shape and appearance models, such as ASM [32] and AAM [33].

2.2.1 Parameterized geometric models

B-Spline curves

Blake et al. [11] propose parametric B-spline curves for contour tracking. The tracking problem is to estimate the control points of the B-spline curve so that the B-spline curve matches the contour being tracked as closely as possible. However, without global constraints, B-spline curves tend to match contours locally, resulting in wrong matching among contour points. The robustness of the algorithm could be improved by employing constraints on the possible solution subspace of the contour points [12]. Therefore, it prevents generating phys-
ically impossible curves. Instead of using grey-level edge information, Kaucic and Blake [74] and Chan [23] utilize the characteristics of human skin color. They propose using either Bayesian classification or linear discriminant analysis to distinguish lips and other areas of facial skin. Therefore, the contours of the lips can be extracted more reliably. It is well known that color segmentation is sensitive to lighting conditions and the effectiveness of color segmentation depends on the subject. This can be partially solved by training a color classifier for each individual. Nevertheless, these two approaches do not handle 3D rotation, translation and appearance changes of lips.

Snake model

Kass et al. [73] propose the snake for tracking deformable contours. It starts from an initial starting point and deforms itself to match with the nearest salient contour. The matching procedure is formulated as an energy minimization process. In basic Snake-based tracking, the function to be minimized includes two energy terms: (1) internal spline energy caused by stretching and bending, and (2) measure of the attraction of image features such as contours. B-Spline [11] is a “least squares” style Snake algorithm. Snakes rely on gray-level gradient information for measuring the energy terms of the snakes. However, it is possible that gray-level gradients in images are inadequate for identifying the contour. Therefore, Terzopoulos and Waters [131] highlighted the facial features by makeup to help Snake-based tracking. Otherwise, Snakes very often align onto undesirable local minima. To improve Snakes, Bregler and Konig [17] propose eigenlips that incorporate a lip shape manifold into Snake tracker for lip tracking. The shape manifold is learned from training sequences of lip shapes. It imposes global constraints on the Snake contour shape model. The local search for maximum gray-level gradients is guided by the globally learned lip shape space.
**Deformable template**

Yullie et al. [153] define a facial feature as a deformable template, which includes a parametric geometrical model and an imaging model. Deformable template poses tracking as an analysis-by-synthesis problem. The geometrical model describes how the shape of the template can be deformed and is used to measure shape distance from the template. The imaging model describes how to generate an instance of the template and is used to measure the intensity distance from the template. An energy function is designed to link different types of low-level image features, such as intensity, peaks, and edges, to the corresponding properties of the template. The parameters of the template are calculated by steepest descent. Nonetheless, the parametric facial feature models are usually defined subjectively.

**3D parameterized model**

DeCarlo and Mataxas [38] propose an approach that combines a deformable model space and multiple image cues (optical flow and edge information) to track facial motions. The edge information used is chosen around certain facial features, such as the boundary of the lips and eyes. To avoid high computation complexity, optical flow is calculated only for a set of image pixels. Those image pixels are chosen in the region covered by the face model using the method proposed by Shi and Tomasi [123]. The deformable model is a parametric geometric mesh model. The parameters are manually designed based on a set of anthropometric measurements of the face. By changing the values of the parameters, the user can obtain a different basic face shape and deform the basic face shape locally. The deformable model in [38] helps to prevent producing unlikely facial shapes during tracking. However, it is labor-intensive to construct the face deformation model, and some facial deformation (e.g. lip deformations produced during speech) may not be represented adequately using the anthropometric measurements.
2.2.2 FACS-based models

A number of researchers have proposed to model facial deformation using FACS system. The FACS based 3D models impose constraints on the subspace of the plausible facial shapes. The motion parameters include global face motion parameters (rotation and translation) and local facial deformation parameters, which correspond to the weights of AUs in [83, 129] and to the FACS-like control parameters in [45]. In these approaches, first, the movements of the vertices on the model are calculated using optical flow. The optical flow results are usually noisy. The facial deformation model is then added to constrain the noisy 2D image motion. The motion parameters are calculated by least square estimator. However, FACS was originally proposed for psychology study and does not provide quantitative information about facial deformations. To utilize FACS, researchers need to manually design the parameters of their model to obtain the AUs. This manual design process is usually labor-intensive. Li et al. [83] uses a parametric geometrical face model, called Candide. The Candide model contains a set of parameters for controlling facial shape. Tao and Huang [129] use a piecewise Bezier volume deformable face model, which can be deformed by changing the coordinates of the control vertices of the Bezier volumes. In [45], Essa and Pentland extended a mesh face model, which was developed by Platt and Badler [113], into a topologically invariant physics-based model by adding anatomy-based muscles, which is defined by FACS.

2.2.3 Statistical models

Active Shape Model (ASM) and Active Appearance Model (AAM)

Active Shape model (ASM) [32], Active Appearance model (AAM) [33], utilize variations of both contour and appearance to model the facial motion. They are both analysis-by-synthesis approaches. ASM and AAM try to achieve robust performance by using the high-level model to constrain solutions to be valid examples of the object being tracked. The appearance of the object is explained by the high-level model as a compact set of model parameters. The
models used by ASM and AAM are the eigen-features of the object. ASM models the shape variation of a set of landmark points and the texture variation in the local areas around landmark points. AAM models the contour and the appearance inside of the contour of the object. Both of them require manually labelled training data, which is labor intensive. The training data need to be carefully labelled so that the correspondences between the landmarks across training samples are physically correctly established. In order to handle various lighting conditions, the texture part of the training data should cover broad enough lighting conditions.

3D model learned from motion capture data

People have recently proposed to train facial motion subspace models from real facial motion data [5, 64, 76, 121], which can capture the real motion characteristics of facial features better than manually defined models. The approaches presented in [5, 121] only deal with lips. The trained 3D model is able to encode the information of real lip deformations. Principal component analysis are used in [64, 76, 121] to derive basis of facial deformation model. Then the basis can be used for face animation [64, 76, 121] and tracking [64].

2.3 Geometric facial motion synthesis

Based on spatial and temporal modeling of facial deformation, facial motion is usually synthesized according to semantic input, such as actor performance [150], text script [141], or speech [15, 99].

2.3.1 Performance-driven face animation

Performance-driven face animation animates face models according to visual input signals. This type of approach automatically analyzes real facial movements in the video using computer vision techniques. The analysis results are used to animate graphic face models.
Williams [150] put markers on the subject’s face and use simple tracking algorithm to estimate the motion of the markers. Guenter et al. [57] put more markers on faces and track their 3D motion to achieve high quality visual input. However, both Williams [150] and Guenter et al. [57] required intrusive markers be put on the face of the subject. This limits the conditions where there approaches can be used. Other approaches [42, 45, 129, 131] used more sophisticated facial motion analysis techniques to avoid using markers. The quality of the animation depends on the estimated facial motions. Therefore, the key issue is to achieve robust and accurate face motion estimation from noisy image motion. However, it is still a challenging problem to estimate facial motions accurately and robustly without using markers.

2.3.2 Text-driven face animation

Synthesizing facial motions during speech is useful in many applications, such as e-commerce [105], computer-aided education [31]. One type of input of this “visual speech” synthesis is text. First, the text is converted into a sequence of phonemes by Text-To-Speech (TTS) system. Then, the phoneme sequence is mapped to corresponding facial shapes, called visemes. Finally, a smooth temporal facial shape trajectory is synthesized considering the co-articulation effect in speech. It is combined with the synthesized audio-only speech signal from TTS as the final animation. Waters et al. [141] and Hong et al. [63] generated the temporal trajectory using sinusoidal interpolation functions. Pelachaud et al. [109] used a look-ahead co-articulation model. Cohen and Massaro [29] adopted Löfqvist gestural production model [87] as the co-articulation model and interactively designed its explicit form based on observation of real speech.
2.3.3 Speech-driven face animation

Another type of input for “visual speech” synthesis is speech signals. For speech-driven face animation, the main research issue is the audio-to-visual mapping. The audio information is usually represented by acoustic features such as linear predictive coding (LPC) cepstrum, Mel-frequency cepstral coefficients (MFCC). The visual information is usually represented by the parameters of facial motion models, such as the weights of AU’s, MPEG-4 FAPs, the coordinates of model control points etc. The mappings are learned from an audio-visual training data set, which are collected in the following way. The facial movements of talking subjects are tracked either manually or automatically. The tracking results and the associated audio tracks are collected as the audio-visual training data.

Some speech-driven face animation approaches use phonemes or words as intermediate representations. Lewis [82] uses linear prediction to recognize phonemes. The recognized phonemes are associated with mouth positions to provide key frames for face animation. However, the phoneme recognition rate of linear prediction is low. Video Rewrite [18] trains hidden Markov models (HMMs) [115] to automatically label phonemes in both training audio track and new audio track. It models short-term mouth co-articulation using triphones. The mouth images for a new audio track are generated by reordering the mouth images in the training footage, which requires a very large database. Video Rewrite is an offline approach and needs large computation resources. Chen and Rao [26] train HMMs to segment the audio feature vectors of isolated words into state sequences. Given the trained HMMs, the state probability for each time stamp is evaluated using the Viterbi algorithm. The estimated visual features of all states can be weighted by the corresponding probabilities to obtain the final visual features, which are used for lip animation. The advantage of using the intermediate representations is that people can make use of the knowledge about speech recognition and the phoneme-to-visual mapping in text-driven animation. The disadvantage is that it requires long enough context information to recognize phoneme or words so that
it can not achieve real-time speech-driven face animation.

Another HMM-based approach tries to directly map audio patterns to facial motion trajectories. Voice Puppetry [15] uses an entropy minimization algorithm to train HMMs for the audio to visual mapping. The mapping estimates a probability distribution over the manifold of possible facial motions from the audio stream. An advantage of this approach is that it does not require automatically recognizing speech into high-level meaningful symbols (e.g., phonemes, words), which is very difficult to obtain a high recognition rate. Nevertheless, this approach is an offline method.

Other approaches attempt to generate instantaneous lip shapes directly from each audio frame using vector quantization, Gaussian mixture model, or artificial neural networks (ANN). Vector quantization-based approach [96] classifies the audio features into one of a number of classes. Each class is then mapped onto a corresponding visual output. Though it is computationally efficient, the vector quantization approach often leads to discontinuous mapping results. The Gaussian mixture approach [120] models the joint probability distribution of the audio-visual vectors as a Gaussian mixture. Each Gaussian mixture component generates an optimal estimation for a visual feature given an audio feature. The estimations are then nonlinearly weighted to produce the final visual estimation. The Gaussian mixture approach produces smoother results than the vector quantization approach. However, neither of these two approach described consider phonetic context information, which is very important for modeling mouth coarticulation during speech. Neural network based approaches try to find nonlinear audio-to-visual mappings. Morishima and Harashima [97] trained a three layer neural network to map LPC Cepstrum speech coefficients of one time step speech signals to mouth-shape parameters for five vowels. Kshirsagar and Magnenat-Thalmann [75] also trained a three-layer neural network to classify speech segments into vowels. Nonetheless, these two approaches do not consider phonetic context information. In addition, they mainly consider the mouth shapes of vowels and neglect the contribution of consonants during speech. Massaro et al. [92] trained multilayer perceptrons (MLP) to
map LPC cepstral parameters to face animation parameters. They try to model the coarticulation by considering the speech context information of five backward and five forward time windows. Another way to model speech context information is to use time delay neural networks (TDNNs) model to perform temporal processing. Lavagetto [78] and Curinga et al. [35] trained TDNN to map LPC cepstral coefficients of speech signal to lip animation parameters. TDNN is chosen because it can model the temporal coarticulation of lips. These artificial neural networks, however, require a large number of hidden units, which results in high computational complexity during the training phrase. Vignoli et al. [138] used self-organizing maps (SOM) as a classifier to perform vector quantization functions and fed the classification results to a TDNN. SOM reduces the dimension of input of TDNN so that it reduces the parameters of TDNN. However, the recognition results of SOM are discontinuous which may affect the final output.

2.4 Appearance-based facial motion modeling, analysis and synthesis

To capture the facial appearance changes not modeled by geometric models, appearance-based approaches use all face image pixel in analysis and synthesis. To reduce the high dimensionality of the appearance space, the image samples are usually aligned first to remove the variation caused by global transformation. This alignment is mostly done manually or semi-automatically. Then, subspace analysis techniques such as Principal Component Analysis (PCA) [71], are used to find low dimensional approximation of the space.

In appearance-based facial motion analysis, Bregler and Konig [17] used PCA to learn “eigen lips” for bi-modal speech recognition. Besides PCA, the dimensionality of appearance space can be also reduced by extracting other texture-based features, such as Gabor wavelets coefficients [41]. To enhance certain features (e.g., edges), face images can be processed by filtering before extracting appearance-based features. These appearance-based features have
been shown [3, 41] to improve the recognition of AU’s which incur detailed facial motion.

Beymer et al. [8] proposed to learn the nonlinear mapping between face appearance and its “parameters” (e.g. “pose”, “expression”) using multidimensional interpolation networks. The mappings could be learned in both direction: (1) from appearance to parameters; and (2) from parameters to appearance. Therefore the framework could be used for both analysis and synthesis. Ezzat et al. [46] extended the idea of “multidimensional interpolation” to visual speech synthesis. Cosatto and Graf [34] generated text-driven animation by finding smooth trajectories in extensive samples based on phonemes.

However, the space of all face appearance is huge, affected by the variations across different head poses, individuals, lighting, expressions, speech and etc. Thus it is difficult for appearance-based methods to collect enough face appearance data and train a model that works robustly in many different scenarios. In this respect, the geometric-based methods are more robust to large head motions, changes of lighting and are less person-dependent.

2.5 Hybrid facial motion modeling, analysis and synthesis

To combine the advantages of both approaches, people have been investigating methods of using both geometry (shape) and appearance (texture) in face analysis and synthesis. Guenter et al. [57] captured both shape and texture using markers in performance-driven face animation. La Cascia et al. [22] modeled the face with a texture-mapped cylinder. 3D rigid face tracking was formulated as a texture image registration problem. The Active Appearance Model (AAM) [33] and its variants, apply PCA to model both the shape variations of image patches and their texture variations. They have been shown to be powerful tools for face alignment, recognition, and synthesis. Blanz and Vetter [14] proposed 3D morphable models for 3D faces modeling, which model the variations of both 3D face shape and texture using PCA. The 3D morphable models have been shown effective in 3D face animation
and face recognition from non-frontal views [13]. Pighin et al. [111] and Revert et al. [121] estimated facial deformation based on the discrepancy between a target face image and the image synthesized from reference face texture images. Arbitrary facial shapes were approximated by a linear combination of a set of basic shapes. Furthermore, a linear combination of a set of reference texture images was used to cope with the texture variations. However, the set of reference texture images should be similar to the target face image, for example, same person, same lighting. Moreover, they were computationally expensive because all image pixels are used in the nonlinear Levenberg-Marquardt optimization. Recently, Liu et al. [86] applied both geometric and textural changes to synthesize realistic facial expressions. The ratio image technique was used to capture the subtle appearance changes independent of the face surface albedo. In facial expression classification, Tian et al. [133] and Zhang et al. [158] proposed to train classifiers (e.g. neural networks) using both shape and texture features. The trained classifiers were shown to outperform classifiers using shape or texture features only.

In these hybrid approaches, some variations of texture are absorbed by shape variation models. However, the potential texture space can still be huge because many other variations are not modeled by the shape model. Moreover, little has been done to adapt the learned models to new conditions. As a result, the application of these methods are limited to conditions similar to those of training data.

### 2.5.1 Issues in flexible appearance model

Because the appearance of facial motions has large variations due to many factors, such poses, people and lighting conditions, it has been a difficult problem to adapt appearance models of facials motions to various conditions. In our framework, we focus on (1) the appearance model adaptation for synthesis over different illumination and people’s face albedo; and (2) online appearance model adaptation during facial motion analysis.
Illumination effects of face appearance

The analysis and synthesis of human faces under arbitrary lighting conditions has been a fascinating yet challenging problem. Despite its difficulty, great progress has been made in the past a few years. One class of methods use statistical methods (e.g. PCA) to find a low dimensional subspace to approximate the space of all possible face appearance under different illumination [22, 53, 51, 59, 122]. The PCA-based subspace can be used in (1) analysis, such as face tracking [22] and face recognition [53, 122] when illumination changes; and (2) synthesize face appearance in different lighting [51, 122, 128]. Recently, Ramamoorthi and Hanrahan [118] used an analytic expression in terms of spherical harmonic coefficients of the lighting to approximate irradiance and they discovered that only 9 coefficients are needed for the appearance of Lambertian objects. Basri and Jacobs [4] obtained similar theoretical results. Assuming faces are Lambertian, they applied the spherical harmonic basis image in face recognition under variable lighting. Ramamoorthi [117] presented an analytic PCA construction of the face appearance under all possible lighting. The results show that the whole space can be well approximated by a subspace spanned by the first five principal components.

To synthesize photo-realistic images of human faces under arbitrary lighting, another class of method is the inverse rendering [88, 36, 152, 37]. By capturing lighting environment and recovering surface reflectance properties, one can generate photo-realistic rendering of objects including human faces under new lighting conditions. To accurately measure face surface reflectance properties, however, special apparatuses such as “light stage” [37] are usually need to be built.

In this book, we present an efficient method to approximate illumination model from a single face image. Then the illumination model is used for face relighting, that is, rendering faces in various lighting conditions. This method has the advantage that it does not require the separation of illumination from face reflectance, and it is simple to implement and runs
at interactive speed.

- Illumination modeling for face recognition

Because illumination affects face appearance significantly, illumination modeling is important for face recognition under varying lighting.

In recent years, there have been works in face recognition community addressing face image variation due to illumination changes [160, 24]. Georghiades et al. [52] present a new method using the illumination cone. Sim and Kanade [125] propose a model and exemplar based approach for recognition. Nonetheless, both [52] and [125] need to reconstruct 3D face information for each subject in the training set. Then they synthesize face images in various lighting to train their face recognizer. Blanz et al. [13] recover the shape and texture parameters of a 3D Morphable Model in an analysis-by-synthesis fashion. These parameters are then used for face recognition. This method needs to compute a statistical texture and shape model from a 3D face database. The illumination effects are modeled by Phong model [49]. When fitting the 3D morphable face model to an input face image, the illumination parameters are estimated along with texture and shape parameters. However, because there are many parameters to estimate and optimization is non-linear, the fitting is computational expensive and need good initialization.

In general, appearance-based methods such as Eigenfaces [136] and AAM [33] need a number of training images for each subject, in order to deal with illumination variability. Previous research suggests that illumination variability in face images is low-dimensional e.g. [1, 4, 7, 117, 44, 59]. Using spherical harmonics presentation of Lambertian reflection, Basri et al. [4] and Ramamoorthi [117] have obtained theoretical derivation of the low dimensional space. Furthermore, a simple scheme for face recognition with excellent results is presented in [4]. However, to use this recognition scheme, the basis images spanning the illumination space for each face are required. These images can be rendered from a 3D scan of the face or can be estimated by applying PCA to a number of images of the same
subject under different illuminations [117]. An effective approximation of this basis by 9 single light source images of a face is reported in [80]. These methods need a number of images and/or 3D scan data of the subjects in the database. Therefore it would requires specialized equipment and procedures for the capture of the training set, thus limiting their applicability. Zhao and Chellappa [159] use symmetric shape-from-shading. But it suffers from the drawbacks of shape-from-shading, such as the assumption of point lighting sources. Zhang and Samaras [154] propose to recover the 9 spherical harmonic basis images from the input image. Nevertheless, the method in [154] needs a 3D database as in [13] to estimate a statistical model of the spherical harmonic basis images.

In our framework, we show that our face relighting technique can be used to normalize the illumination effects in face images. The advantage of the method is that it does not require extra information as in previous methods to model the illumination effects. In our experiment, we demonstrate that this pre-processing step helps reduce error rates in face recognition under varying lighting environments.

**Person dependency**

Facial appearance variations are highly person-dependent, because the different people have different facial surface properties and different styles of motion. To deal with the variations across different people, one approach is to collect training data from a variety of people. Nonetheless, the amount of data needed to achieve good results could be huge.

For a Lambertian object, the ratio image [86] of its two aligned images removes its surface albedo dependency thus allowing illumination terms (geometry and lighting) to be captured and transferred. Therefore, the subtle appearance changes due to detailed facial motion can be captured independent of the face albedo. Then, the appearance changes can be mapped to people with different albedos.
Online appearance model

The *Expectation-Maximization* (EM) technique [40] is a framework for optimization with partial information. It is widely applied for computing maximum likelihood estimates for parameters in incomplete data models. By alternating between an expectation step (E-step), which finds expected completions of data given the current parameterization, and a maximization step (M-step), which re-estimates parameters on the basis of completed data, the EM algorithm gradually improves the likelihood for the observed data until convergence at a local maximum.

Using the EM framework, Jepson et al. [70] propose an online appearance model, which is updated at every frame by the EM algorithm for adapting to newly observed facial appearance variations. However, only current stable mode of facial appearance is modeled and the non-rigid facial motions are not interpreted by the model.

An analogy of the appearance model adaption problem in speech recognition domain is speaker adaptation. A good survey can be found in [151]. One of the most popular schemes is to adjust model parameters such that the likelihood or posterior probability of new adaptation data is maximized. The EM algorithm can usually be used for this maximization process, treating the parameters to be adjusted as “missing” parameters of the model. One type of approach, called *Maximum Likelihood Linear Regression* (MLLR) [50], is to linearly transform the parameters of a speaker-independent model such that the likelihood of the adaptation data of a particular person is maximized. MLLR has the advantage that the same linear transformation can be used to update all parameters to be adjusted, even if the number is large. If relatively few parameters to be adjusted, MLLR will be robust and unsupervised adaptation can be used. In practice, research has shown that it is effective to adjust the mean and covariance of the acoustic GMM model using MLLR.
Chapter 3

Learning Geometric 3D Facial Motion Model

In this chapter, we introduce the method for learning geometric 3D facial motion model in our framework. 3D facial motion model describes the spatial and temporal deformation of 3D facial surface. Efficient and effective facial motion analysis and synthesis requires a compact yet powerful model to capture real facial motion characteristics. For this purpose, analysis of real facial motion data is needed because of the high complexity of human facial motion.

We first introduce the motion capture database used in our framework in Section 3.1. Section 3.2 and 3.3 present our methods for learning holistic and parts-based spatial geometric facial motion models, respectively. Section 3.4 introduces how we apply the learned models to arbitrary face mesh. Finally, in Section 3.5, we brief describe the temporal facial motion modeling in our framework.

3.1 Motion Capture Database

To study the complex motion of face during speech and expression, we need an extensive motion capture database. The database can be used to learn facial motion models. Furthermore, it will benefit future study on bi-modal speech perception, synthetic talking head development and evaluation and etc. In our framework, we have experimented on both data
collected using *MotionAnalysis*™ system, and the facial motion capture data provided by Dr. Brian Guenter [57] of Microsoft Research.

MotionAnalysis [100] **EvaRT 3.2** system is a marker-based capture device, which can be used for capturing geometric facial deformation. An example of the marker layout is shown in Figure 3.1. There are 44 markers on the face. Such marker-based capture devices have high temporal resolution (up to 300fps), however the spatial resolution is low (only tens of markers on face are feasible). Appearance details due to facial deformation, therefore, is handled using our flexible appearance model presented in chapter 6.

The Microsoft data, collected by by Guenter et al. [57], use 153 markers. Figure 3.2 shows an example of the markers. For better visualization purpose, we build a mesh based on those markers, illustrated by Figure 3.2 (b) and (c).

![Figure 3.1: An example of marker layout for MotionAnalysis system.](image)

### 3.2 Learning Holistic Linear Subspace

To make complex facial deformation tractable in computational models, people have usually assumed that any facial deformation can be approximated by a linear combination of some basic deformation. In our framework, we make the same assumption, and try to find optimal bases under this assumption. We call these bases *Motion Units* (MUs). Using MUs, a facial
Figure 3.2: The markers of the Microsoft data [57]. (a): The markers are shown as small white dots. (b) and (c): The mesh is shown in two different viewpoints.

The shape \( \mathbf{s} \) can be represented by

\[
\mathbf{s} = \mathbf{s}_0 + \left( \sum_{i=1}^{M} c_i \mathbf{e}_i + \mathbf{e}_0 \right)
\]

(3.1)

where \( \mathbf{s}_0 \) denotes the facial shape without deformation, \( \mathbf{e}_0 \) is the mean facial deformation, \( \{\mathbf{e}_0, \mathbf{e}_1, ..., \mathbf{e}_M\} \) is the MU set, and \( \{c_1, c_2, ..., c_M\} \) is the MU parameter (MUP) set.

In this thesis, we experiment on both of the two databases described in Section 3.1. Principal Component Analysis (PCA) [71] is applied to learning MUs from the database. The mean facial deformation and the first seven eigenvectors of PCA results are selected as the MUs. The MUs correspond to the largest seven eigenvalues that capture 93.2% of the facial deformation variance. The first four MUs are visualized by an animated face model in Figure 3.3. The top row images are the frontal views of the faces, and the bottom row images are side views. The first face is the neutral face, corresponding to \( \mathbf{s}_0 \). The remaining faces are deformed by the first four MUs scaled by a constant (from left to right). The method for visualizing MUs is described in Section 3.4. Any arbitrary facial deformation can be approximated by a linear combination of the MUs, weighted by MUPs. MUs are used in robust 3D facial motion analysis presented in Chapter 4, and facial motion synthesis.
presented in Chapter 5.

Figure 3.3: The neutral face and deformed face corresponding to the first four MUs. The top row is frontal view and the bottom row is side view.

3.3 Learning Parts-based Linear Subspace

It is well known that the facial motion is localized, which makes it possible to decompose the complex facial motion into smaller parts. The decomposition helps: (1) reduce the complexity in deformation modeling; (2) improve the robustness in motion analysis; and (3) flexibility in synthesis. The decomposition can be done manually based on the prior knowledge of facial muscle distribution, such as in [111, 129]. However, the decomposition may not be optimal for the linear combination model used because of the high nonlinearity of facial motion. Parts-based learning techniques, together with extensive motion capture data, provide a way to help design parts-based facial deformation models, which can better approximate real local facial motion. Recently several learning techniques have been proposed for learning representation of data samples that appears to be localized. Non-negative matrix Factorization (NMF) [79] has been shown to be able to learn basis images that resemble parts
of faces. In learning the basis of subspace, NMF imposes non-negativity constraints, which is compatible to the intuitive notion of combining parts to form a whole in a non-subtractive way.

In our framework, we present a parts-based face deformation model. In the model, each part corresponds to a facial region where facial motion is mostly generated by local muscles. The motion of each part is modeled by PCA as described in Section 3.2. Then, the overall facial deformation is approximated by summing up the deformation in each part:

\[
\Delta \vec{s} = \sum_{j=1}^{N} \Delta \vec{s}_j = \sum_{j=1}^{N} \left( \sum_{i=1}^{M_j} c_{ij} \vec{e}_{ij} + \vec{e}_{0j} \right)
\]

(3.2)

where \(\Delta \vec{s} = \vec{s} - \vec{s}_0\) is the deformation of the facial shape. \(N\) is the number of parts. We call this representation parts-based MU, where the \(j\)-th part has its MU set \(\{\vec{e}_{0j}, \vec{e}_{1j}, ..., \vec{e}_{Mj}\}\), and MUP set \(\{c_{1j}, c_{2j}, ..., c_{Mj}\}\).

To decompose facial motion into parts, we use NMF together with prior knowledge. In this method, we randomly initialize the decomposition. Then, we use NMF to reduce the linear decomposition error to a local minimum. We impose the non-negativity constraint in the linear combination of the facial motion energy. We use a matlab implementation of NMF from the web site

\textit{http://journalclub.mit.edu} (under category “Computational Neuroscience”). The algorithm is an iterative optimization process. In our experiments, we use 500 iterations. Figure 3.4(a) shows some parts derived by NMF. Adjacent different parts are shown in different patterns overlayed on the face model. We then use prior knowledge about facial muscle distribution to refine the learned parts. The parts can thus be (1) more related to meaningful facial muscle distribution, (2) less biased by individuality in the motion capture data, and (3) more easily generalized to different faces. We start with an image of human facial muscle distribution, illustrated in Figure 3.4(b) [48]. Next, we align it with our generic face model via image warping, based on facial feature points illustrated in Figure 3.7(c). The aligned
facial muscle image is shown in Figure 3.4(c). Then, we overlay the learned parts on facial muscle distribution (Figure 3.4(d)), and adjust interactively the learned parts such that different parts correspond different muscles. The final parts are shown in Figure 3.4(e). The parts are overlapped a bit as learned by NMF. For convenience, the overlap is not shown Figure 3.4(e).

![Image](image.png)

Figure 3.4: (a): NMF learned parts overlayed on the generic face model. (b): The facial muscle distribution. (c): The aligned facial muscle distribution. (d): The parts overlayed on muscle distribution. (e): The final parts decomposition.

The learned parts-based MUs give more flexibility in local facial deformation analysis and synthesis. Figure 3.5 and 3.6 show some local deformation in lower lips and right cheek, each of which is induced by one of the learned parts-based MUs. These locally deformed shapes are difficult to approximate using holistic MUs in Section 3.2. For each deformation shown in Figure 3.5 and 3.6, more than 100 holistic MUs are need to achieve a 90% reconstruction accuracy. That means, although some local deformation is induced by only one parts-based MU, more than 100 holistic MUs may be needed in order to achieve good analysis and synthesis quality. Therefore, we can have more flexibility in using parts-based MUs. For example, if we are only interested in motion in forehead, we only need to capture data about face with mainly forehead motion, and learn parts-based MUs from the data. In face animation, people often want to animate local region separately. This task can be easily achieved by adjusting MUPs of parts-based MUs separately. In face tracking, such as the system described in Chapter 4, people may use parts-based MUs to track only region of
Figure 3.5: Three lower lips shapes deformed by three of the lower lips parts-based MUs respectively. The top row is the frontal view and the bottom row is the side view.

Figure 3.6: (a): The neutral face side view. (b): The face deformed by one right cheek parts-based MU.
their interests (e.g. the lips). Furthermore, tracking using parts-based MUs is more robust because local error will not affect distant regions.

### 3.4 Animate Arbitrary Mesh Using MU

The learned MUs are based the motion capture data of particular subjects. To use the MUs for other people, they need to be fitted to the new face geometry. Moreover, the MUs only sample the facial surface motion at the position of the markers. The movements at other places need to be interpolated. We call this process “MU” fitting.

In our framework, we use the face models generated by “iFACE” for MU-based face animation. “iFACE” is a face modeling and animation system developed in [63]. The generic face model in iFACE is shown in Figure 3.7(a). Figure 3.7(b) shows a personalized model, which we customize based on the *Cyberware*\textsuperscript{TM} scanner data for that person. Figure 3.7(c) shows the feature points we define on the iFACE generic model, which we use for MU fitting.

![Figure 3.7: (a): The generic model in iFACE. (b): A personalized face model based on the *Cyberware*\textsuperscript{TM} scanner data. (c): The feature points defined on generic model.](image)

In our previous work [65], we used MUs to animate models generated by iFACE. We
dealt with the MU fitting problem by constructing a mapping between MUs and the face deformation model of iFACE. This technique allowed a key-frame based face animation system to use MUs. First we selected a set of training facial shapes with known MUPs. In a key-frame based animation system, these training shapes can be represented by linear combinations of key frames. Based on the equality between the two representations of the training shapes, the conversion between parameters of key-frame combination and MUPs could be derived as described in [65]. This method enabled us to use MUs for animation in a traditional key-frame-based animation system, such as iFACE. However, key frames of a certain system may not be expressive enough to take advantage of the motion details in MUs. Thus, the facial deformation information can be lost during conversion between parameters of key-frame combination and MUPs. Alternatively, interpolation-based techniques for re-targeting animation to new models, such as [103], could be used for MU fitting. In similar spirit to [103], we design our MU fitting as a two-step process: (1) face geometry based MU-adjustment; and (2) MU re-sampling. These two steps can be improved in a systematic way if enough MU sets are collected. For example, if MU statistics over a large set of different face geometries are available, one can systematically derive the geometry-to-MU mapping using machine-learning techniques. On the other hand, if multiple MU sets are available, which sample different positions of the same face, it is possible to combine them to increase the spatial resolution of MU because markers in MU are usually sparser than face geometry mesh.

The first step adjusts MUs to a face model with different geometry. The fundamental problem is to find a mapping from face geometry to MUs. Currently no data are available yet for MU statistics over different face geometry. We assume that the corresponding positions of the two faces have the same motion characteristics. Then, the adjustment is done by moving the markers of the learned MUs to their corresponding positions on the new face. We interactively build the correspondence of facial feature points shown in Figure 3.7(c) by labelling them via a GUI. Then, image warping technique is used to interpolate
correspondence in the remaining part. Note that the correspondences are based on only 2D facial feature locations, because only one image of a face is used in the GUI. We are working on using automatic facial feature localization techniques (e.g. [67]) to automate this step.

The second step is to derive movements of facial surface points that are not sampled by markers in MUs. This is essentially a signal re-sampling problem, for which an interpolation-based method is usually used. We use the popular radial basis interpolation function. The family of radial basis functions (RBF) is well known for its powerful interpolation capability. RBF is widely used in face model fitting [112] and face animation [57, 89, 103]. Using RBF, the displacement of a certain vertex \( \vec{v}_i \) is of the form

\[
\Delta \vec{v}_i = \sum_{j=1}^{N} w_{ij} h(\|\vec{v}_i - \vec{p}_j\|) \Delta \vec{p}_j
\]  

(3.3)

where \( \vec{p}_j, (j = 1, ..., N) \) is the coordinate of a marker, and \( \Delta \vec{p}_j \) is its displacement. \( h \) is a radial basis kernel function such as Gaussian function, and \( w_{ij} \) are the weights. \( h \) and \( w_{ij} \) need to be carefully designed to ensure the interpolation quality. For facial deformation, the muscle influence region is local. Thus, we choose a cut-off region for each vertex. We set the weights to be zero for markers that are outside of the cut-off region, i.e., they are too far away to influence the vertex. In our current system, the local influence region for the i-th vertex is heuristically assigned as a circle, with the radius \( r_i \) as the average of the distances to its two nearest neighbors. Similar to [89], we choose the radial basis kernel to be \( h(x) = (1 + \cos(\pi \frac{x}{r_i}))/2, \) where \( x = \|\vec{v}_i - \vec{p}_j\| \). We choose \( w_{ij} \) to be the normalization factor such that \( \sum_{j=1}^{N} w_{ij} h(\|\vec{v}_i - \vec{p}_j\|) = 1 \). The lips and eye lids are two special cases for this RBF interpolation, because the motions of upper parts of them are not correlated with the motions of the lower parts. To address this problem, we add “upper” or “lower” tags to vertices and markers near mouth and eyes. If a marker \( M \) and a vertex \( V \) have different tags, \( M \) has no influence on \( V \). Thus, the weight of the marker \( M \) is set to be zero in the RBF interpolation (equation 3.3) of the vertex \( V \). These RBF weights need to be computed...
only once for one set of marker positions. The weights are stored in a matrix. The matrix is sparse because marker influence is local. During synthesis, the movement of mesh vertices can be computed by one multiplication of the sparse RBF matrix based on equation 3.3. Thus the interpolation is fast.

After the MU fitting, MUs can be used to animate any 3D face models or analyze facial motion in image sequences. Deformed face image examples presented in this chapter, such as faces deformed by first four MUs in Figure 3.3, are produced after fitting MU to iFACE models. MU fitting is also used in facial motion analysis in Chapter 4.

### 3.5 Temporal Facial Motion Model

In this section, we describe the temporal facial deformation trajectory modeling in the 3D face processing framework. The model describes temporal variation of facial deformation given constraints (e.g. key shapes) at certain time instances. For compactness and usability, we adopt the spline-based approach of Ezzat et al. [46]. The advantage of this approach is that it employs only a few key facial shapes estimated from training data. This approach also utilizes the statistics of the training data in a way that covariance of the key shapes are used as weights in the spline trajectory. This helps to increase the likelihood of the generated trajectory.

In our framework, we can alternatively infer facial deformation dynamics from correlated signals, such as in speech-driven animation and visual face tracking. In that case, the facial deformation is inferred from input signals at every time instance. (See Chapter 4, Section 4.1 for visual tracking driven animation, and Chapter 5, Section 2.3.3 for speech-driven animation.) The mapping from related signal to facial deformation, however, can be many-to-many (as in speech-driven animation) or noisy (as in tracking). Thus, the temporal model is still useful. We plan to incorporate dynamics model in speech-driven animation and visual face tracking in the future.
3.6 Summary

In this chapter, a geometric 3D facial motion model is introduced. Compared to handcrafted models, the proposed model is derived from motion capture data so that it can capture the characteristics of real facial motions more easily. We have also discussed methods for applying the motion model to different subjects and face models with different topologies. The applications of the motion model in face analysis and synthesis will be presented in Chapter 4 and Chapter 5, respectively.
Chapter 4

Geometric Model-based 3D face tracking

To achieve robust 3D non-rigid face tracking, facial motion model is needed. In this chapter, we describe the geometric MU-based facial motion tracking algorithm in Section 4.1. Section 4.2 will describe applications of our geometric 3D face tracking algorithm.

4.1 Geometric MU-based 3D Face Tracking

In the conventional 3D non-rigid face tracking algorithm using FACS-based 3D facial deformation model, the subspace spanned by the Action Units (AUs) is used as the high-level knowledge to guide face tracking. Similar to MU, AUs are defined such that arbitrary facial deformation is approximated by a linear combination of AUs. However, the AUs are usually manually designed. For these approaches, our automatically learned MUs can be used in place of the manually designed AUs. In this way, extensive manual intervention can be avoided, and natural facial deformation can be approximated better.

We choose to use the learned MUs in the 3D non-rigid face tracking system proposed in [129], because the system has been shown to be: (1) robust in face of gradual background changes; (2) able to recover from temporary loss of tracking; and (3) real-time in tracking speed. For these reasons, this tracking system has been effectively used for bimodal speech recognition [157] and emotion recognition [27]. The facial motion observed in image plane
can be represented by

\[ M(\vec{R}(\vec{V}_0 + L\vec{P}) + \vec{T}) \]  (4.1)

where \( M \) is the projection matrix, \( \vec{V}_0 \) is the neutral face, \( L\vec{P} \) defines the non-rigid deformation, \( \vec{R} \) is the 3D rotation decided by three rotation angles \([w_x, w_y, w_z]^t = \vec{W}\), and \( \vec{T} \) stands for 3D translation. \( L \) is an \( N \times M \) matrix that contains \( M \) AUs, each of which is an \( N \) dimensional vector. \( \vec{P} = [p_1, ..., p_M]^t \) is the coefficients of the AUs. To estimate facial motion parameters \( \{\vec{T}, \vec{W}, \vec{P}\} \) from 2D inter-frame motion \( \vec{V}_{2D} \), the derivative of equation 4.1 is taken with respect to \( \{\vec{T}, \vec{W}, \vec{P}\} \). Then, a linear equation between \( \vec{V}_{2D} \) and \( \{d\vec{T}, d\vec{W}, d\vec{P}\} \) can be derived by ignoring high order derivatives (see details in [129]). The system estimates \( \vec{V}_{2D} \) using template-matching-based optical flow. After that, the linear system is solved using least square in a multi-resolution manner for efficiency and robustness.

In the original system, \( L \) was manually designed using Bezier volume, and represented by the displacements of vertices of face surface mesh. The design process was labor-intensive. To derive \( L \) from the learned MUs in our current system, the “MU fitting” process described in Chapter 3 is used. For the adaptation, it requires that the face be in its neutral position in the first image frame and the facial feature locations are detected. In the original system, it is already done interactively via a GUI. In the future, we plan to detect these feature locations automatically such that the whole fitting could be automatic.

In the current system, we use the holistic MUs derived in Section 3.2 of Chapter 3. Parts-based MUs could be used if a certain local region is the focus of interests, such as the lips in speech reading. The system is implemented in a PC with two 2.2 GHz Pentium 4 processors and 2GB memory. The image size of the input video is 640 × 480. With only one CPU employed, the system works at 14 frame/second for non-rigid face tracking. The tracking results can be visualized by using the coefficients of MUs, \( \vec{R} \) and \( \vec{T} \) to directly animate face models. Figure 4.1 shows some typical frames that it tracked, along with the animated face model to visualize the results. It can be observed that compared with
Figure 4.1: Typical tracked frames and corresponding animated face models. (a): The input image frames. (b): The tracking results visualized by yellow mesh overlayed on input images. (c): The front views of the face model animated using tracking results. (d): The side views of the face model animated using tracking results. In each row, the first image corresponds to neutral face.
neutral face (the first column images), the mouth opening (the second column), subtle mouth
rounding and mouth protruding (the third and fourth columns) are captured in the tracking
results visualized by animated face models.

4.2 Applications of Geometric 3D Face Tracking

The facial motion synthesis using tracking results can be used in model-based face video
coding such as [68, 135]. In our face video coding experiments [135], we track and encode the
face area using model-based coding. To encode the residual in face area and the background
for which a-priori knowledge is not generally available, we use traditional waveform-based
coding method H.26L. This hybrid approach improves the robustness of the model-based
method at the expense of increasing bit-rate. Eisert et al. [42] proposed a similar hybrid
coding technique using a different model-based 3D facial motion tracking approach. We
capture and code videos of 352 × 240 at 30Hz. At the same low bit-rate (18 kbits/s), we
compare this hybrid coding with H.26L JM 4.2 reference software. In Chapter 9, Figure 4.2
shows three snapshots of a video that we used in our experiment. This video has 147 frames.
For the video used in our experiments, the Peak Signal to Noise Ratio (PSNR) around facial
area for hybrid coding is 2dB higher compared to H.26L. Moreover, the hybrid coding results
have much higher visual quality. Because our tracking system works in real-time, it could
be used in a real-time low bit-rate video phone application. More details of the model-based
face video coding application will be discussed in Chapter 9.

Besides low bit-rate video coding, the tracking results can used as the visual features
for audio-visual speaker independent speech recognition [157], and emotion recognition [27].
The bimodal speech recognition system improves the speech recognition rate in noisy en-
vironments. The emotion recognition system is being used to monitor students’ emotional
and cognitive states in a computer-aided instruction application. In medical applications
related to facial motion disorder such as facial paralysis, visual cues are important for both
Figure 4.2: (a): The synthesized face motion. (b): The reconstructed video frame with synthesized face motion. (c): The reconstructed video frame using H.26L codec.
diagnosis and treatment. Therefore, the facial motion analysis method can be used as a diagnostic tool such as in [139]. Compared to other 3D non-rigid facial motion tracking approaches using single camera, which are cited in Chapter 2, the features of our tracking system can be summarized as: (1) the deformation space is learned automatically from data such that it avoids manual crafting but still captures the characteristics of real facial motion; (2) it is real-time so that it can be used in real-time human computer interface and coding applications; (3) To reduce “drifting” caused by error accumulation in long-term tracking, it uses templates in both the initial frame and previous frame when estimating the template-matching-based optical flow (see [129]); and (4) it is able to recover from temporary loss of tracking by incorporating a template-matching-based face detection module.

4.3 Summary

In this chapter, a robust real-time 3D geometric face tracking system is presented. The proposed motion-capture-based geometric motion model is adopted in the tracking system to replace the original handcrafted facial motion model. Therefore, extensive manual editing of the motion model can be avoided. Using the estimated geometric motion parameters, we conduct experiments on video-driven face animation and very low bit-rate face animation. The experimental results show that geometric facial motions can be effectively captured in the tracking results. The presented tracking system can be used in many other HCI applications, such as facial expressions recognition, human behavior understanding, and etc.
Chapter 5

Geometric Facial Motion Synthesis

In this chapter, we discuss the facial motion synthesis using the trained Motion Unit model. Our goal is to use learned models to synthesize plausible facial appearance for providing visual cues in synthetic face based interactions. In Chapter 3, we have discussed that linear combinations of geometric MUs can be used to approximate arbitrary face shapes. In this chapter, we focus on how to produce animation according to various input signals, such as text and speech. First, we briefly introduce the face modeling tools in our iFACE system in Section 5.1. The face models will later be used as the foundation for 3D face animation. Next, text-driven face animation and offline speech-driven animation are discussed in Section 5.2 and 5.3, respectively. Finally we describe real-time speech-driven animation in Section 5.4.

5.1 3D Face Modeling Tools in iFACE

We have developed iFACE system which provides functionalities for face modeling and face animation. It provides a research platform for the 3D face processing framework. The iFACE system takes the Cyberware\textsuperscript{TM} scanner data of a subject’s head as input and allows the user to interactively fit a generic face model to the Cyberware\textsuperscript{TM} scanner data. The iFACE system also provides tools for text-driven face animation and speech-driven face animation. The animation techniques will be described in Chapter 5.
5.1.1 Generic face model

The generic face model in the iFACE system consists of nearly all the head components such as face, eyes, teeth, ears, tongue, and etc. The surfaces of the components are approximated by triangular meshes. There are 2240 vertices and 2946 triangles. The tongue component is modeled by a Non-Uniform Rational B-Splines (NURBS) model which has 63 control points. The generic face model is illustrated in Figure 5.1.

5.1.2 Personalized face model

In iFACE, the process of making a personalized face model is nearly automatic with only a few manual adjustments necessary. To customize the face model for a particular person, we first obtain both the texture data and range data of that person by scanning his/her head using Cyberware™ range scanner. An example of the range scanner data is shown in Figure 5.2.

We define thirty-five facial feature points on the face surface of the generic head model. If we unfold the face component of the head model onto 2D, those feature points triangulate the face mesh into several local patches. The 2D locations of the feature points in the range map are manually selected on the scanned texture data, which are shown in Figure 5.3. The
system calculates the 2D positions of the remaining face mesh vertices on the range map by deforming the local patches based on the range data. By collecting the range information according to the positions of the vertices on the range map, the 3D facial geometry is decided. The remaining head components are automatically adjust by shifting, rotating, and scaling. Interactive manual editing on the fitted model are required where the scanned data are missing. We have developed an interactive model editing tools to make the editing easy. The interface of the editing tool is shown in Figure 5.4. After editing, texture map is mapped onto the customized model to achieve photo-realistic appearance. Figure 5.5 shows an example of a customized face model.
5.2 Text-driven Face Animation

When text is used in communication, e.g., in the context of text-based electronic chatting over the Internet or visual email, visual speech synthesized from text will greatly help deliver information. Recent work on text driven face animation includes the work of Cohen and Massaro [29], Ezzat and Poggio [47], and Waters and Levergood [141].

Similar to the work of Ezzat and Poggio [47] and that of Waters and Levergood [141], our framework adopts the key frame based face animation technique for text-driven face animation. The procedure of the text driven face animation is illustrated in Figure 5.6.
Our framework uses Microsoft Text-to-Speech (TTS) engine for text analysis and speech synthesis. First, the text stream is fed into the TTS engine. TTS parses the text and generates the corresponding phoneme sequence, the timing information of phonemes, and the synthesized speech stream. Each phoneme is mapped to a viseme based on a lookup table. Each viseme is a key frame. Therefore, the text is translated into a key frame sequence. A temporal trajectory is then synthesized based on the key frame sequence using the spline-based technique in [46].

In the framework, we use a label system that has forty-four phonemes. Seventeen viseme groups are designed to group visually similar phonemes together. The phonemes and their viseme group labels are shown in Table 5.1.

In our experiment, we use the motion capture data in Chapter 3 to train the key shape model for each viseme. Each shape is represented using MUPs, which are modeled using a Gaussian model. Each key shape has a mean shape and a covariance matrix. The key shape viseme models are then used in the key-frame-based face animation, such as text-driven animation. Four of the key shapes and the largest components of their variances are shown...
<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Word</th>
<th>Viseme label</th>
<th>Phoneme</th>
<th>Word</th>
<th>Viseme label</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIL</td>
<td>Silence</td>
<td>17</td>
<td>HX</td>
<td>head</td>
<td>8</td>
</tr>
<tr>
<td>IY</td>
<td>beat</td>
<td>10</td>
<td>M</td>
<td>meet</td>
<td>1</td>
</tr>
<tr>
<td>IH</td>
<td>bit</td>
<td>8</td>
<td>N</td>
<td>net</td>
<td>5</td>
</tr>
<tr>
<td>EY</td>
<td>bait</td>
<td>8</td>
<td>NX</td>
<td>sing</td>
<td>8</td>
</tr>
<tr>
<td>EH</td>
<td>bet</td>
<td>8</td>
<td>EL</td>
<td>bottle</td>
<td>11</td>
</tr>
<tr>
<td>AE</td>
<td>bat</td>
<td>13</td>
<td>D</td>
<td>debt</td>
<td>5</td>
</tr>
<tr>
<td>AA</td>
<td>pot</td>
<td>14</td>
<td>EN</td>
<td>button</td>
<td>5</td>
</tr>
<tr>
<td>AY</td>
<td>buy</td>
<td>8</td>
<td>F</td>
<td>fin</td>
<td>2</td>
</tr>
<tr>
<td>AW</td>
<td>down</td>
<td>8</td>
<td>V</td>
<td>vet</td>
<td>2</td>
</tr>
<tr>
<td>AH</td>
<td>but</td>
<td>8</td>
<td>TH</td>
<td>thin</td>
<td>3</td>
</tr>
<tr>
<td>AO</td>
<td>bought</td>
<td>15</td>
<td>DH</td>
<td>this</td>
<td>3</td>
</tr>
<tr>
<td>OW</td>
<td>boat</td>
<td>16</td>
<td>S</td>
<td>sit</td>
<td>5</td>
</tr>
<tr>
<td>OY</td>
<td>boy</td>
<td>5</td>
<td>Z</td>
<td>zoo</td>
<td>5</td>
</tr>
<tr>
<td>UH</td>
<td>book</td>
<td>7</td>
<td>SH</td>
<td>shoe</td>
<td>4</td>
</tr>
<tr>
<td>UW</td>
<td>lute</td>
<td>6</td>
<td>ZH</td>
<td>measure</td>
<td>4</td>
</tr>
<tr>
<td>YU</td>
<td>cute</td>
<td>12</td>
<td>P</td>
<td>pet</td>
<td>1</td>
</tr>
<tr>
<td>AX</td>
<td>about</td>
<td>7</td>
<td>B</td>
<td>bet</td>
<td>1</td>
</tr>
<tr>
<td>IX</td>
<td>kisses</td>
<td>5</td>
<td>T</td>
<td>test</td>
<td>5</td>
</tr>
<tr>
<td>W</td>
<td>wet</td>
<td>6</td>
<td>G</td>
<td>get</td>
<td>5</td>
</tr>
<tr>
<td>Y</td>
<td>yet</td>
<td>5</td>
<td>K</td>
<td>kit</td>
<td>5</td>
</tr>
<tr>
<td>R</td>
<td>red</td>
<td>9</td>
<td>CH</td>
<td>church</td>
<td>4</td>
</tr>
<tr>
<td>LL</td>
<td>let</td>
<td>11</td>
<td>JH</td>
<td>judge</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.1: Phoneme and viseme used in face animation.
in Figure 5.7. They correspond to phonemes: (a) M; (b) AA; (c) UH; and (d) EN. Because we only use the relative ratio of the variance values, we normalize variance values to be in the range $[0, 1]$.

![Figure 5.7: Four of the key shapes. The top row images are front views and the bottom row images are the side views. The largest components of variances are (a): 0.67; (b): 1.0; (c): 0.18; (d): 0.19.](image)

### 5.3 Offline Speech-driven Face Animation

When human speech is used in one-way communication, e.g. news broadcasting over the networks, using real speech in face animation is better than using synthetic speech. Because the communication is one-way, the audio-to-visual mapping can be done offline, i.e. the animation can come out until the end of a batch of speech. The process of offline speech driven face animation is illustrated in Figure 5.8. An advantage of offline process is that the phoneme transcription and timing information can be extracted accurately for animation.
purpose, given the text script of the speech. Recognizing phoneme using only speech signals requires a complicated continuous speech recognizer, and the phoneme recognition rate and the timing information may not be accurate enough. The text script associated with speech, however, provides the accurate word-level transcription so that it greatly simplifies the complexity and also improves the accuracy. We use a phoneme speech alignment tool, which comes with Hidden Markov Model Toolkit (HTK) 2.0 [66] for phoneme recognition and alignment. Speech stream is decoded into phoneme sequence with timing information. Once we have the phoneme sequence and the timing information, the remaining part of the procedure of the visual speech synthesis is similar to text driven face animation.

5.4 Real-time Speech-driven Face Animation

Real-time speech-driven face animation is demanded for real-time two-way communications such as teleconferencing. The key issue is to derive the real-time audio-to-visual mapping.
In Section 5.4.1 we present a real-time audio-to-visual mapping based on artificial neural network (ANN).

### 5.4.1 ANN-based real-time speech-driven face animation

In this section, we present our real-time speech-driven 3D face animation algorithm, as an animation example based on MUs in our 3D face analysis and synthesis framework. In this section, we propose a local non-linear audio-to-visual mapping based on ANN. We first classify audio features into groups. Then for each group we train an ANN for audio-to-visual mapping. In this way, the mapping can be more robust and accurate than simple classification based methods, such as VQ [96] and GMM [120]. Our multi-ANN-based mapping is also more efficient in training than methods using only a single ANN [99, 75, 92, 78].

#### Training data and features extraction

We use the facial motion capture database (described in Section 3.1 of Chapter 3) along with its audio track for learning audio-to-visual mapping. To reduce the complexity of learning and make it more robust, the visual feature space should be small. Thus for this specific application we use the holistic MUs (Section 3.2 of Chapter 3) as the visual representation. For each 33 ms short time window, we calculate MUPs as the visual features and calculate twelve Mel-frequency cepstrum coefficients (MFCCs) [116] as the audio features. The audio feature vectors of frames $t - 3$, $t - 2$, $t - 1$, $t$, $t + 1$, $t + 2$, and $t + 3$, are concatenated in the temporal order as the final audio feature vector of frame $t$. Consequently, the audio feature vector of each audio frame has eighty-four elements. The frames $t - 3$, $t - 2$, $t - 1$, $t + 1$, $t + 2$, and $t + 3$ define the contextual information of the frame $t$.

#### Audio-to-visual mapping

We modify the approaches that train neural networks as the audio-to-visual mapping [64, 99, 92]. The training audio-visual data is divided into twenty-one groups based on the audio
feature of each data sample. The number twenty-one is decided heuristically based on audio feature distribution of the training database. Particularly, one of the groups corresponds to silence because human beings are very sensitive to mouth movements if there is no sound generated. Other twenty groups are automatic generated using the k-means algorithm. Then, the audio features of each group are modelled by a Gaussian model. After that, a three-layer perceptron is trained to map the audio features to the visual features using each audio-visual data group. At the estimation phase, we first classify an audio vector into one of the audio feature groups whose Gaussian model gives the highest score for the audio feature vector. We then select the corresponding neural network to map the audio feature vector to MUPs, which can be used in equation 3.1 to synthesize the facial shape. A method using triangular average window is used to smooth the jerky mapping results.

For each group, eighty percent of the data is randomly selected for training. The remaining data is used for testing. The maximum number of the hidden neurons is ten. The minimum number of the hidden neurons is four. A typical estimation result is shown in Figure 5.9. The horizontal axes in the figure represent time. The vertical axes represent the magnitude of the MUPs. The solid trajectory is the original MUPs, and the dashed trajectory is the estimation results.

We reconstruct the facial deformation using the estimated MUPs and MUs. For both the ground truth and the estimated results, we divide the deformation of each facial feature point by its maximum absolute displacement in the ground truth data so that the magnitude of deformation is normalized to \([-1.0, 1.0]\). To evaluate the performance, we calculate the Pearson product-moment correlation coefficients (R) and the mean square error (MSE) using the normalized deformations. The Pearson product-moment correlation \((0.0 \leq R \leq 1.0)\) measures how good the global match is between the shapes of two signal sequences. The larger the Pearson correlation coefficient, the better the estimated signal sequence matches with the original signal sequence.

The Pearson product-moment correlation coefficient R between the ground truth \(\{\tilde{d}_n\}\)
Figure 5.9: Compare the estimated MUPs with the original MUPs. The content of the corresponding speech track is “A bird flew on lighthearted wing.”
and the estimated data $\{\tilde{d}_n\}$ is calculated by

$$ R = \frac{\text{tr}(E[(\tilde{d}_n - \mu_1)(\tilde{d}_n - \mu_2)^T])}{\sqrt{\text{tr}(E[(d_n - \tilde{\mu}_1)(d_n - \tilde{\mu}_1)^T]) \text{tr}(E[(\tilde{d}_n - \mu_2)(\tilde{d}_n - \mu_2)^T])}} $$

(5.1)

where $\tilde{\mu}_1 = E[\tilde{d}_n]$ and $\tilde{\mu}_2 = E[d_n]$. In our experiment, $R = 0.952$ and MSE = 0.0069 for training data, and $R = 0.946$ and MSE = 0.0075 for testing data.

**Animation result**

The whole speech-driven 3D face animation procedure contains three steps. First, we extract audio features from the input speech stream, as described in Section 5.4.1. Then, we use the trained neural networks to map the audio features of an audio frame into the visual features (i.e. MUPs). Finally, we use the estimated MUPs to animate a personalized 3D face model in iFACE, to which the MUs have been adapted using methods described in Section 3.4 of Chapter 3. A typical animation sequence is presented in Figure 5.10.

![Figure 5.10: Typical frames of the animation sequence of “A bird flew on lighthearted wing.” The temporal order is from left to right, and from top to bottom.](image)

Our real-time speech driven animation can be used in real-time two-way communication scenarios such as videophone, immersive conferencing in virtual environments [81]. On the
other hand, existing off-line speech driven animation (e.g. “voice puppetry” [15]) can be used in one-way communication scenarios, such as broadcasting, advertising. Our approach deals the mapping of both vowels and consonants, thus it is more accurate than real-time approaches with only vowel-mapping [75, 99]. Compared to real-time approaches using only one neural network for all audio features [78, 92], our local ANN mapping (i.e. one neural network for each audio feature cluster) is more efficient because each ANN is much simpler. Therefore it can be trained with much less effort for a certain set of training data. More generally, speech driven animation can be used in speech and language education [31], speech understanding aid for noisy environment and hard-of-hearing people, rehabilitation tool for facial motion disorders treatment.

**Human emotion perception study**

The synthetic talking face, which is used to convey visual cues to human, can be evaluated by human perception study. Here, we describe our experiments which compare the influence of the synthetic talking face on human emotion perception with that of the real face. We did similar experiments for 2D MU-based speech driven animation [65]. The experimental results can help the user with how to use the synthetic talking face to deliver the intended visual information.

We videotape a speaking subject who is asked to calmly read three sentences with 3 facial expressions: (1) neutral, (2) smile, and (3) sad, respectively. Hence, the audio tracks do not convey any emotional information. The contents of the three sentence are: (1) “It is normal.”; (2) “It is good.”; and (3) “It is bad.”. The associated information is: (1) neutral; (2) positive; and (3) negative. The audio tracks are used to generate three sets of face animation sequences. All three audio tracks are used in each set of animation sequence. The first set is generated without expression. The second set is generated with smile expression. The third set is generated with sad expression. The facial deformation due to speech and expression is linearly combined in our experiments. Sixteen untrained human subjects, who
Facial Expressions

<table>
<thead>
<tr>
<th>Emotion</th>
<th>Neutral</th>
<th>Smile</th>
<th>Sad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>Neutral</td>
<td>16</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>Happy</td>
<td>0</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Sad</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.2: Emotion inference based on video without audio track.

<table>
<thead>
<tr>
<th>Emotion</th>
<th>Audio 1</th>
<th>Audio 2</th>
<th>Audio 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neutral</td>
<td>16</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Happy</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Sad</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 5.3: Emotion inference based on audio track.

never used our system before, participate the experiments.

The first experiment investigates human emotion perception based on either the visual stimuli alone or the audio stimuli alone. The subjects are first asked to recognize the expressions of both the real face and the synthetic talking face and infer their emotional states based on the animation sequences without audio. All subjects correctly recognized the expressions of both the synthetic face and the real face. Therefore, our synthetic talking face is capable to accurately deliver facial expression information. The emotional inference results in terms of the number of the subjects are shown in Table 5.2. The “S” columns in Table 5.2, as well as in Table 5.4, 5.5, and 5.6, show the results using the synthetic talking face. The “R” columns show the results using the real face. As shown, the effectiveness of the synthetic talking face is comparable with that of the real face. The subjects are then asked to listen to the audio and decide the emotional state of the speaker. Each subject listens to each audio only once. Note that the audio tracks are produced without emotions. The results in terms of the number of the subjects are shown in Table 5.3.

The second and third experiments are designed to compare the influence of synthetic face on bimodal human emotion perception and that of the real face. In the second experiment, the subjects are asked to infer the emotional state while observing the synthetic talking face.
and listening to the audio tracks. In the third experiment, the subjects are asked to infer the emotional state while observing the real face and listening to the same audio tracks. We divide the subjects into two groups. Each of them has eight subjects. One group first participates the second experiment and then participates the third experiment. The other group first participates the third experiment and then participates the second experiment. The results are then combined and compared in Table 5.4, 5.5, and 5.6.

We can see the face movements (either synthetic or real) and the content of the audio tracks jointly influence the decisions of the subjects. Let us take the first audio track as an

<table>
<thead>
<tr>
<th>Facial Expressions</th>
<th>Neutral</th>
<th>Smile</th>
<th>Sad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>Neutral</td>
<td>16</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Happy</td>
<td>0</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>Sad</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.4: Emotion inference based on video with audio track 1.

<table>
<thead>
<tr>
<th>Facial Expressions</th>
<th>Neutral</th>
<th>Smile</th>
<th>Sad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>Neutral</td>
<td>14</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>Happy</td>
<td>2</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>Sad</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Not sure</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.5: Emotion inference based on video with audio track 2.

<table>
<thead>
<tr>
<th>Facial Expressions</th>
<th>Neutral</th>
<th>Smile</th>
<th>Sad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>Neutral</td>
<td>11</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>Happy</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Sad</td>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Not sure</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.6: Emotion inference based on video with audio track 3.
example (see Table 5.4). Although the first audio track only contains neutral information, fourteen subjects think the emotional state is happy if the expression of the synthetic talking face is smile. If the expression of the synthetic face is sad, fifteen subjects classify the emotional state into sad.

If the audio tracks and the facial represent the same kind of information, the human perception on the information is enhanced. For example, when the associated facial expression of the audio track 2 is smile, nearly all subjects say that the emotional state is happy (see Table 5.5). The numbers of the subjects who agree with happy emotion are higher than those using visual stimuli alone (see Table 5.2) or audio information alone (see Table 5.3).

However, it confuses human subjects if the facial expressions and the audio tracks represent opposite information. For example, many subjects are confused when they listen to an audio track with positive information, and observe a negative facial expression. An example is shown in the seventh and eighth columns of Table 5.5. The audio track conveys positive information while the facial expression is sad. Six subjects report that they are confused if the synthetic talking face with sad expression is shown. The number of the confused subjects reduces to four if the real face is used. This difference is mainly due to the fact that the subjects tend to trust real face more than synthetic face when confusion happens. Therefore, when the visual information conflicts with the audio information, the synthetic face is less capable of conveying fake emotion information.

Overall, the experimental results show that our real-time speech-driven synthetic talking face successfully affects human emotion perception. The effectiveness of the synthetic face is comparable with that of the real face though it is slightly weaker.

5.5 Summary

In this chapter, we present face animation methods based on geometric motion model. We describe three types of animation: text-driven, offline speech-driven and real-time speech-
driven face animation. In all of the scenarios, our proposed geometric facial motion model serves as a compact and effective representation of visual information. A human perception study is conducted to demonstrate that the effectiveness of our synthetic face animation in conveying emotional cues.
Chapter 6

Flexible Appearance Model

In previous chapters we have presented the geometric facial motion model, and how the model can be used in facial motion analysis and synthesis. Geometric motion model handles the macro- and meso-structure level deformations. However, facial motions also exhibit detailed appearance changes such as wrinkles and creases as well. These details are important visual cues for both human perception and computer analysis. Nevertheless, they are difficult to analyze and synthesize using geometric motion models. In this chapter, we introduce our flexible appearance model in our 3D face processing framework. It aims to deal with facial motion details, thus enhance the geometric model based analysis and synthesis. However, the space of all face appearance is huge, affected by the variations across different head poses, individuals, lighting, expressions, speech and etc. Thus it is difficult for most existing appearance-based methods to collect enough face appearance data and train a model that works robustly in many different scenarios. Compared to these approaches, the novelty of our flexible appearance model is that we propose efficient and effective methods to reduce illumination dependency and person dependency of the appearance model, based on limited data. Therefore, it is more flexible to use the appearance model in varying conditions.

In Chapter 2, we have given a review of the related works on utilizing appearance models for facial motion analysis and synthesis. In this chapter, we first introduce our flexible appearance model in Section 6.1. Section 6.1.1 describes the illumination modeling component of the flexible appearance model. Then, we discuss the technique for reducing person de-
pendency in Section 6.1.2. Analysis and synthesis using the flexible model will be presented later in Chapter 7 and 8, respectively.

6.1 Flexible Appearance Model

In this section, we present a flexible appearance model with reduced dependency on illuminations and individuals. We develop an efficient method to model illumination effects from a single face image [148]. The illumination model can be used to reduce illumination dependency of the appearance model in both analysis and synthesis. The flexible appearance model also utilize ratio-image technique to reduce person dependency in a principled way [148, 146]. We discuss the two components in the Section 6.1.1 and 6.1.2. Analysis and synthesis using this flexible model will be presented in Chapter 7 and 8, respectively.

6.1.1 Reduce illumination dependency based on illumination modeling

Complete modeling of face illumination effects is a complex task, including diffuse reflection, specular reflection (e.g. Phong reflection model), subsurface scattering etc [28, 90]. In our framework, we choose to model the diffuse refection component, which suffices in many face analysis applications and gives plausible results for face synthesis [4, 7, 52, 80, 122, 130].

Radiance environment map (REM)

Environment map [56, 94] is a technique in computer graphics for capturing environment illuminations from all directions using a sphere. The captured lighting environment can be used together with surface reflectance properties (i.e. “Bidirectional Reflectance Distribution Function” (BRDF)) to generate photo-realistic rendering of objects. For applications dealing with images, such as image/video analysis and image-based rendering, it is difficult to recover accurate surface reflectance properties from images. Another issue is that it could be time
consuming to integrate illumination and complex surface reflectance properties at rendering time.

In these scenarios, environment maps that pre-integrate with BRDF can be useful. Radiance environment map is proposed by Greene [56] and Cabral et al. [20] for real-time rendering of objects rotating in lighting environment. Given any sphere with constant “Bidirectional Reflectance Distribution Function” (BRDF), its radiance environment map records its reflected radiance for each point on the sphere. Then, for any surface with the same reflectance material as that of the sphere, its reflected radiance at any point can be found from the radiance environment map by simply looking up the normal. This method is very fast and produces photo-realistic results.

One limitation of this method is that one needs to generate a radiance environment map for each different material. For some objects where every point’s material may be different, it would be difficult to apply this method. To overcome this limitation, we observe that, for diffuse objects, the ratio image technique [122, 86] can be used to remove the material dependency of the radiance environment map, thus making it possible to relight surfaces which have variable reflectance properties. Details of the ratio-image technique will be described in Section 6.1.2.

Given a diffuse sphere of constant reflectance coefficient $\rho$, let $\mathcal{L}$ denote the distant lighting distribution. The irradiance on the sphere is then a function of normal $\bar{n}$, given by an integral over the upper hemisphere $\Omega(\bar{n})$ at $\bar{n}$.

$$E(\bar{n}) = \int_{\Omega(\bar{n})} \mathcal{L}(\omega)(\bar{n} \cdot \omega)d\omega \quad (6.1)$$

The intensity of the sphere is

$$I_{sphere}(\bar{n}) = \rho E(\bar{n}) \quad (6.2)$$

$I_{sphere}(\bar{n})$ is the radiance environment map. Notice that radiance environment map depends on both the lighting and the reflectance coefficient $\rho$. The intensity of the sphere $I_{sphere}(\bar{n})$
is usually warped to a reference plane to extract 2D image map $I_{sphere}(u, v)$, where $(u, v)$ is the coordinate system of the 2D reference plane.

We can derive similar formulas for face appearance if we assume faces are Lambertian, and ignore cast shadows which is a common assumption for environment map based techniques. Let $\vec{n}(u, v)$, $\rho(u, v)$ denote the normal and albedo of a face surface point at texture plane $(u, v)$. Suppose the face is in the same lighting environment as $I_{sphere}(\vec{n})$. The irradiance on the face is same as $E(\vec{n})$, which can be re-written using coordinate system $(u, v)$ as:

$$E(u, v) = \int_{\Omega(\vec{n}(u, v))} L(\omega)(\vec{n}(u, v) \cdot \omega) d\omega$$  \hspace{1cm} (6.3)

The intensity of the neutral face point $p$ at $(u, v)$ is

$$I(u, v) = \rho(u, v)E(u, v)$$  \hspace{1cm} (6.4)

We can observe that for sphere and face points with the same normal, the pixels intensity can be computed with the same lighting $E(u, v)$. The only difference is the different albedos for the two surfaces, which we can handle using the ratio-image technique described in Section 6.1.2.

**Approximating a radiance environment map using spherical harmonics**

A radiance environment map can be captured by taking photographs of a sphere painted with a constant diffuse material. It usually requires multiple views and they need to be stitched together. However, spherical harmonics technique [118] provides a way to approximate a radiance environment map from one or more images of a sphere or other types of surfaces.

Using the notation of [118], the irradiance can be represented as a linear combination of
spherical harmonic basis functions:

\[
E(\vec{n}) = \sum_{l \geq 0, -l \leq m \leq l} \hat{A}_l L_{lm} Y_{lm}(\vec{n})
\]  

(6.5)

In equation 6.5, \(L_{lm}\) are coefficients indicating the intensity of corresponding lighting base function; \(\hat{A}_l\) are constants whose values can be found in [118]; and \(Y_{lm}(\vec{n})\) are spherical harmonic basis. If the normal \(\vec{n}\) is specified by its cartesian components \((x, y, z)\), the first two order basis \(Y_{lm}(\vec{n})\) are given numerically by

\[
Y_{00}(\vec{n}) = 0.282095
\]

\[
(Y_{11}; Y_{10}; Y_{1-1})(\vec{n}) = 0.488603(x; y; z)
\]

\[
(Y_{21}; Y_{2-1}; Y_{2-2})(\vec{n}) = 1.092548(xz; yz; xy)
\]

\[
Y_{20}(\vec{n}) = 0.315392(3z^2 - 1)
\]

\[
Y_{22}(\vec{n}) = 0.546274(x^2 - y^2)
\]

(6.6)

Ramamoorthi and Hanrahan [118] showed that for diffuse reflectance, only 9 coefficients are needed to approximate the irradiance function. Therefore, given an image of a diffuse surface with constant albedo \(\rho\), its reflected radiance at a point with normal \(\vec{n}\) can be approximated as

\[
\rho E(\vec{n}) \approx \sum_{l \leq 2, -l \leq m \leq l} \rho \hat{A}_l L_{lm} Y_{lm}(\vec{n})
\]

(6.7)

If we treat \(\rho \hat{A}_l L_{lm}\) as a single variable for each \(l\) and \(m\), we can solve for these 9 variables using a least square procedure, thus obtaining the full radiance environment map. This approximation gives a very compact representation of the radiance environment map, using only 9 coefficients per color channel.

An important extension is the type of surface whose albedo, though not constant, does
not have low-frequency components (except the constant component). To justify this, we
define a function \( \rho(\vec{n}) \) such that \( \rho(\vec{n}) \) equals to the average albedo of surface points whose
normal is \( \vec{n} \). We expand \( \rho(\vec{n}) \) using spherical harmonics as:

\[
\rho(\vec{n}) = \rho_{00} + \Psi(\vec{n})
\]  

(6.8)

where \( \rho_{00} \) is the constant component and \( \Psi(\vec{n}) \) contains other higher order components.
Together with equation (6.7), we have

\[
\rho(\vec{n})E(\vec{n}) \approx \rho_{00} \sum_{l \leq 2, -l \leq m \leq l} \hat{A}_l L_{lm} Y_{lm}(\vec{n}) + \Psi(\vec{n}) \sum_{l \leq 2, -l \leq m \leq l} \hat{A}_l L_{lm} Y_{lm}(\vec{n})
\]  

(6.9)

If \( \Psi(\vec{n}) \) does not have first four order \((l = 1, 2, 3, 4)\) components, the second term of the
righthand side in equation (6.9) contains components with orders equal to or higher than 3
(see Appendix 10.2.3 for the explanation). Therefore, if we define \( SpharmonicProj(I) \) to
the be function which projects the face image \( I \) into the 9 dimensional spherical harmonic
space, we have

\[
SpharmonicProj(I(\vec{n})) = SpharmonicProj(\rho(\vec{n})E(\vec{n})) = \rho_{00} \sum_{l \leq 2, -l \leq m \leq l} \hat{A}_l L_{lm} Y_{lm}(\vec{n})
\]  

(6.10)

Therefore, the 9 coefficients of order \( l \leq 2 \) estimated from \( \rho(\vec{n})E(\vec{n}) \) with a linear least
square procedure are \( \rho_{00} \hat{A}_l L_{lm} \), where \((l \leq 2, -l \leq m \leq l)\). Hence, we obtain the radiance
environment map with reflectance coefficient equal to the average albedo of the surface. This
observation agrees with [119] and perception literature (such as Land’s retinex theory [77]),
where on Lambertian surface high-frequency variation is due to texture, and low-frequency
variation probably associated with illumination.
We believe that human face skins are approximately this type of surfaces. The skin color of a person’s face has dominant constant component, but there are some fine details corresponding to high frequency components in frequency domain. Therefore the first four order components must be very small. To verify this, we used SpharmonicKit [126] to compute the spherical harmonic coefficients for the function $\rho(\vec{n})$ of the albedo map shown in Figure 6.1 which was obtained by Marschner et al. [89]. There are normals that are not sampled by the albedo map, where we assigned $\rho(\vec{n})$ the mean of existing samples. We find that the coefficients of order 1, 2, 3, 4 components are less than 6% of the constant coefficient.

Approximating a radiance environment map from a single image

Given a single photograph of a person’s face, it is possible to compute its 3D geometry [14, 155]. Alternatively, we choose to use a generic geometric model (see Figure 8.4(a)) because human faces have similar shapes, and the artifacts due to geometric inaccuracy are not very strong since we only consider diffuse reflections.

Given a photograph and a generic 3D face geometry, we first align the face image with the generic face model. Details of the 2D-3D alignment is discussed in the implementation
Section 8.1.3 of Chapter 8. After the photograph is aligned with the 3D geometry, we know the normal of each face pixel and thus can compute the $Y_{lm}(\vec{n})$ term in equation 6.10. Next, we can solve for the lighting coefficients in equation 6.10 and obtain the approximated radiance environment map.

Note that it is an under-constrained problem to determine all the 9 coefficients from a single frontal image of a face according to [117]. To produce plausible illumination approximation results without conflicting with the information provided by the frontal image, we make assumptions about lighting in the back to constrain the problem. One of the assumptions that we make is to assume a symmetric lighting environment, that is, the back has the same lighting distribution as the front. This assumption is equivalent to assuming $L_{lm} = 0$ for $(l,m) = (1,0), (2,-1), (2,1)$ in equation (6.7). The rest of the coefficients can then be solved uniquely according to [117]. One nice property about this assumption is that it generates the correct lighting results for the front, and it generates plausible results for the back if faces rotate in the lighting environment. For applications which deal with only frontal lighting, the symmetric assumption produces correct results. If we want to synthesize face appearance after the face is rotated in the lighting environment, we need to make the assumption based on the scene. For example, in the cases where the lights mainly come from the two sides of the face, we use symmetric assumptions. In the cases where the lights mainly come from the front, we assume the back is dark.

6.1.2 Reduce person dependency based on ratio-image technique

Ratio image

In Section 6.1.1, we have derived the formula (equation 6.4) for the intensity of the neutral face point $p$ at $(u,v)$. After the face surface is deformed, the intensity of $p$ is

$$I'(u,v) = \rho(u,v)E'(u,v)$$  \hspace{1cm} (6.11)
We denote

\[ \mathcal{R}(u, v) = \frac{I'(u, v)}{I(u, v)} = \frac{E'(u, v)}{E(u, v)} \]  

(6.12)

It can be observed that \( \mathcal{R}(u, v) \), called the ratio image, is independent of surface reflectance property \( \rho(u, v) \) [86]. Therefore, \( \mathcal{R}(u, v) \) can be used to as a facial motion representation independent of face albedos.

**Transfer motion details using ratio image**

The albedo-independency of ratio image give a novel representation of facial motion field which is less person dependent than the original image. Liu et al. [86] use this property to map facial expressions from one person to another and achieve photo-realistic results.

Given two aligned images of person A’s face \( I_A \) and \( I'_A \), suppose \( I_A \) is the neutral face image and \( I'_A \) is the image of the deformed face. Based on equation 6.12, we have

\[ \mathcal{R}_A(u, v) = \frac{I'_A(u, v)}{I_A(u, v)} = \frac{E'_A(u, v)}{E_A(u, v)} \]  

(6.13)

For a different person B, if we know its neutral face image \( I_B \), we can compute the deformed face image of B where B has the same motion as A. Similarly, we can compute the ratio image for B as

\[ \mathcal{R}_B(u, v) = \frac{I'_B(u, v)}{I_B(u, v)} = \frac{E'_B(u, v)}{E_B(u, v)} \]  

(6.14)

We assume the images of A and B are aligned, that is, for every point on A, there is a corresponding point on B which has the same semantics (eye corners, mouth corners, etc). This alignment can be done using the techniques described in Section 6.1.1. Since human faces have approximately the same geometrical shapes, their surface normals at corresponding points are roughly the same, that is, \( \vec{n}_A(u, v) \approx \vec{n}_B(u, v) \). We also assume that the deformation of the two faces are roughly the same. Then we have \( \vec{n}'_A(u, v) \approx \vec{n}'_B(u, v) \). Based on
equation 6.3, we can derive $E_A(u,v) \approx E_B(u,v)$ and $E'_A(u,v) \approx E'_B(u,v)$. That leads to

$$R_A(u,v) \approx R_B(u,v)$$

(6.15)

From equations 6.13, 6.14 and 6.15, we can compute the unknown image $I'_B$ as

$$I'_B \approx R_A(u,v) I_B$$

(6.16)

Here the multiplication means pixel-by-pixel multiplication of the two images.

Besides computing novel face image $I'_B$ for synthesis, ratio image can also be used to design less person dependent appearance features for motion analysis. This aspect will be discussed in more details in Chapter 7.

**Transfer illumination using ratio image**

The albedo-independency of ratio image also enables illumination effects transfer across different surfaces. Wen et al. [148] use this property for face relighting from a single face image.

Following the derivation in Section 6.1.2, if the difference between $I_A$ and $I'_A$ is illumination effects instead of non-rigid motion, we can obtain the same equation 6.16 for computing novel face appearance $I'_B$ in the lighting environment of $I'_A$. More generally, $A$ could be other objects such as the sphere of radiance environment map. Then we can use the REM illumination model together with equation 6.16 for modifying illumination effects in face images.

### 6.2 Summary

In this chapter, we have reviewed related works on using appearance models for face analysis and synthesis. Then, we introduce our flexible appearance model, which contains two
components for reducing dependencies on illuminations and individuals. We will present our facial motion analysis using the flexible appearance model in Chapter 7. Flexible appearance model based synthesis will be discussed in Chapter 8.
Chapter 7

Facial Motion Analysis Using Flexible Appearance Model

In this chapter, we discuss the face motion analysis using flexible appearance model described in Chapter 6. Our goal is to utilize appearance cues to improve the detection and classification of subtle motions which exhibit similar geometric features. For this purpose, we design novel appearance features for the analysis of these detailed motions. Compared with most existing appearance features, our appearance features are less illumination dependent and less person dependent. In our facial expression classification experiment, we show that this appearance features improve the classification performance under variations in lighting, 3D poses and person.

In Section 7.1, we first describe the proposed novel appearance features and explain how the dependencies on illumination and person are reduced. Then we describe an online appearance model adaptation scheme to further improve the performance in changing conditions. Next, in Section 7.2, experimental results on facial expression recognition are presented to demonstrate the efficacy of the novel appearance features.
7.1 Model-based 3D Face Motion Analysis Using Both Geometry and Appearance

Given a face video, we can employ both geometric and appearance features to analyze the facial motions. The system diagram for the hybrid motion analysis is illustrated in figure 7.1. First, we use a geometric-based method described in Chapter 4 to estimate 3D geometric deformation. The geometric deformation features are extracted as the coefficients of MUs. Figure 7.2(b) shows a snapshot of the geometric tracking system, where a yellow mesh is used to visualize the geometric motions of the face. The input video frame is shown in Figure 7.2(a).

The geometric deformation parameters determine the registration of each image frame to the face texture map. Thus we can derive a face texture map $I(u, v)$ from each image frame, which are independent of geometric motion. Here, $(u, v)$ is the coordinate system of the texture map. Figure 7.2(c) shows the extracted texture map. From the texture maps, we
extract appearance-based features described in Section 7.1.1. These features are designed for subtle details of facial expression and independent of people’s face surface albedo. We then use the texture (appearance) features, together with shape (geometric) features, to analyze face appearance variations based on semantically meaningful exemplars.

To extend a trained appearance-based exemplar model to new conditions, an online EM-based algorithm is used to update the appearance model of these exemplars progressively.

### 7.1.1 Feature extraction

We assume faces are Lambertian. Let \( I(u, v) \), \( I'(u, v) \) denote the neutral face texture and deformed face texture, respectively. We denote

\[
\mathcal{R}(u, v) = \frac{I'(u, v)}{I(u, v)}
\]  

As pointed out by Section 6.1.2 of Chapter 6, the ratio image \( \mathcal{R}(u, v) \) is independent of surface reflectance property \( \rho(u, v) \) [86]. Therefore, \( \mathcal{R}(u, v) \) can be used to characterize facial motions of faces with different albedos.

To use \( \mathcal{R}(u, v) \) in face tracking, more compact features need to be extracted from the high dimensional ratio image. Because low frequency variation of facial motion could be captured by geometric-based methods, we extract features from \( \mathcal{R}(u, v) \) in frequency domain and
use the high frequency components as the features for motions not explained by geometric features. Second, past studies on facial motions [158, 133] have shown that there are certain facial areas where high frequency appearance changes are more likely to occur and thus suitable for texture feature extraction. We apply this domain knowledge in our feature extraction. However, because of noise in tracking and individual variation, it is difficult to locate these locations automatically with enough precision. Therefore, we extract the texture-based features in facial regions instead of points, and then use the weighted average as the final feature. Eleven regions are defined on the geometric-motion-free texture map. These eleven regions are highlighted on the texture map in Fig. 7.3. Note that these regions can be considered constant in the automatically extracted texture map, where the facial feature points are aligned by geometric tracking.

![Figure 7.3: Selected facial regions for feature extraction.](image)

Gabor wavelets are used to extract the appearance changes as a set of multi-scale and multi-orientation coefficients. In our implementation, we use two spatial frequency scales with wavelength of 5 and 8 pixels, and 6 orientations at each scale. Thus for each point, we have $2 \times 6 = 12$ Gabor wavelets coefficients. We choose to compute the Gabor wavelets coefficients of the logarithm of $\Re(u, v)$, denoted by $Z(u, v)$. Based on equation (7.1) and the linearity property of Gabor transform, we have

$$Z(u, v) = G(\log(\Re(u, v)))$$
\[ G(\log(I'(u,v))) - G(\log(I(u,v))) \]  

(7.2)

where function \( G \) denotes a Gabor transform as in [133, 158]. We impose a positive lower bound on pixel values in texture \( I' \) and \( I \) to avoid singular situations. In our approach, only the magnitudes of Gabor transform results are used because the phases are very sensitive to noise in positions. Then we note that if \( Z(u,v) < 0 \), it means the neutral face texture \( I \) contains more high frequency components than the deformed face texture \( I' \). It could be caused by any of the following reasons: (1) the misalignment of \( I' \) and \( I \); (2) high gradient of \( \log(I) \) due to low intensities of \( I \); (3) flattening of wrinkles and creases on neutral face during motion. Scenarios (1) and (2) should be considered as noise, and (3) rarely happens in common human facial motions. Thus we discard negative values of \( Z(u,v) \).

In practice, we need to account for the foreshortening effect of the texture projection. For a 3D face surface patch, the larger its visible area in input image, the higher confidence we should have on the extracted features of the corresponding texture patch. To this end, we construct a confidence map \( \kappa(u,v) \) following [22], which is based on the ratio of each 3D surface patch’s projected area in the texture plane and its area in the input image. For each facial motion region \( q \) (\( q = 1...11 \)), we compute a confidence coefficient \( c_q \) as the average of the \( \kappa(u,v) \) in this region. The resulting confidence coefficients are used to weight the features in tracking described in Section 7.1.3.

\( \Re(u,v) \) contains noise due to misalignment of \( I' \) and \( I \). To reduce the influences of noise on the appearance feature, we construct another weight map \( w(u,v) \), which tries to give large weight for features in deformed area and small weight for features in un-deformed area. We define \( w(u,v) = 1 - \text{corr}(u,v) \) in similar spirit as [86], where \( \text{corr}(u,v) \) is the normalized cross-correlation coefficient between two patches centered at \((u,v)\) from \( G(\log(I')) \) and \( G(\log(I)) \). The idea is that high frequency components of \( \log(I') \) and \( \log(I) \) should be close for un-deformed area, since \( I' \) and \( I \) are roughly aligned by geometric-feature-based tracking. We use \( w(u,v) \) to compute the weighted average of Gabor wavelets coefficients in the 11 selected
regions. For each region, an appearance feature vector of 12-dimension is computed.

### 7.1.2 Influences of lighting

Under the assumptions of Lambertian faces, distant illumination and ignoring cast shadows, the proposed appearance features are not sensitive to changes of lighting conditions. According to [4, 119], the irradiance can be represented by a linear combination of spherical harmonic basis function. For Lambertian surfaces, only the first 2 orders of the basis functions (9 basis) are needed to approximate the irradiance, that is

\[
E(u, v) \approx \sum_{l \leq 2, -l \leq m \leq l} \hat{A}_l L_{lm} Y_{lm}(\vec{n}(u, v))
\]

(7.3)

where \(\hat{A}_l\) is a constant, \(L_{lm}\) is a coefficient decided by lighting, and \(Y_{lm}\) is the spherical harmonic basis function. Assuming that neutral face and the deformed face are in the same lighting condition, we have

\[
\Re(u, v) = \frac{E'(u, v)}{E(u, v)} \approx \frac{\sum_{l \leq 2, -l \leq m \leq l} \hat{A}_l L_{lm} Y_{lm}(\vec{n}'(u, v))}{\sum_{l \leq 2, -l \leq m \leq l} \hat{A}_l L_{lm} Y_{lm}(\vec{n}(u, v))}
\]

(7.4)

The high frequency facial motion \(d_H(u, v)\) can produce high frequency differences between \(\vec{n}'(u, v)\) and \(\vec{n}(u, v)\), and therefore between \(Y_{lm}(\vec{n}'(u, v))\) and \(Y_{lm}(\vec{n}(u, v))\). The irradiance \(E'(u, v)\) will contain a linear combination of these high frequency differences weighted by the lighting coefficients. In other words, high frequency changes in \(\Re(u, v)\) are due to facial deformation details, while lighting will only modulate them in low frequency. If the neutral face and the deformed face are in different lighting conditions, the neutral face texture can be relit to the lighting of the deformed face using face relighting technique in [148]. According to [148], the relighting is a low-frequency filtering processing so that the above arguments are still true.
7.1.3 Exemplar-based texture analysis

The appearance model are designed to model facial motion details that are not captured by low dimensional geometric models. These motion details exhibit much larger variation than geometric motions across different individuals and lighting conditions. Thus a good low-dimensional subspace approximation of motion details variation may be difficult. Nevertheless, facial motions exhibit common semantic exemplars such as typical expressions and visemes, which makes it meaningful to use exemplar-based approach such as [134]. In exemplar-based approach, an observation is interpreted using the probability of the observation being each exemplar.

To explain the face appearance variation, we choose exemplars set \( \Xi = \{x_k, k = 1, ..., K\} \), which are semantically meaningful such as face expressions or visemes. A texture image is interpreted as a state variable \( X \) of the exemplars. Unlike [134], these exemplars incur both shape and texture changes. Let \( Y_S \) and \( Y_T \) denote the shape and texture features respectively. The observation is \( Y = \{Y_S, Y_T\} \). We assume \( Y_S \) and \( Y_T \) are conditionally independent given \( X \). The observation likelihood is

\[
p(Y|X) = p(Y_S, Y_T|X) = p(Y_S|X)p(Y_T|X) \tag{7.5}
\]

In addition, we assume the texture features in different facial motion regions are independent given \( X \). Their log likelihoods are weighted by confidence coefficients \( c_q \) to account for foreshortening effect. That is

\[
\log p(Y_T|X) = \sum_{q=1}^{Q} c_q \log p(Y_{T_q}|X) \tag{7.6}
\]

where \( Q \) is the number of facial motion regions (\( Q = 11 \)). \( p(Y_S|X) \) and \( p(Y_{T_q}|X) \) are modelled using Gaussian Mixture Model (GMM), assuming diagonal covariance matrices. The feature vectors are normalized by their magnitudes. If the neutral face is chosen as an exemplar, we
assign the likelihood using a neutral face classifier such as [132].

Based on the observation likelihoods in equation (7.5) and a dynamics model (e.g. the HMM-based model described in [134]), \( p_t(X_t) \equiv p_t(X_t|Y_1, ..., Y_t) \) can be computed using equation (7.7) according to [115]

\[
p_t(X_t) = \sum_{X_{t-1}=1}^{K} p(Y_t|X_t) p(X_t|X_{t-1}) p_{t-1}(X_{t-1}) \tag{7.7}
\]

In our experiment, we assume uniform conditional density \( p(X_t|X_{t-1}) \) for the dynamics model. Assuming uniform priors, we have \( p_t(X_t) \propto p(Y_{S_t}, Y_{T_t}|X_t) \). The exemplar tracking result can be displayed as \( \hat{X}_t = \arg \max p_t(X_t) \).

### 7.1.4 Online EM-based adaptation

A trained model for facial motion exemplars may work poorly if it can not adapt to lighting changes, or differences in a new individual’s exemplars. Fast adaptation algorithm is needed to avoid re-training the model from scratch. Furthermore, it is tedious to collect and label new training data for each new condition. Therefore, we propose to progressively update the model during tracking in an unsupervised way. Because the geometric features are less person-dependent and less sensitive to lighting changes, we assume the geometric component of the initial exemplar model can help to “confidently” track some new data samples. Then the Expectation-Maximization (EM) framework [40] can be applied to update the model parameters. At time \( t \), the E-step provides exemplar ownership probabilities defined as

\[
o_{k,t}(Y_t) = \frac{p(Y_t|X_t = k)}{\sum_{k=1}^{K} p(Y_t|X_t = k)} \tag{7.8}
\]

where \( k \) is the index of the exemplars. In the M-step, the model is adapted by computing new maximum likelihood estimates of its parameters. Note that we only adapt the texture part of the model because shape features are less person-dependent and not sensitive to
changes of lighting.

The idea of Maximum Likelihood Linear Regression (MLLR) can be generalized to this adaptation problem, where we estimate a linear transformation of the GMM mean vectors to maximize the likelihood of new observations. However, conventional MLLR is not an online method which requires multiple data samples for maximum likelihood optimization. In the M-step of our online EM algorithm, only one data sample is available at a time. Thus we constrain the transformation of the GMM mean vectors to be translation only. The M-step of our algorithm is then to estimate $\Delta \mu_{q,k,t}$, which denotes the translation of the GMM mean vectors from initial model, for the $q^{th}$ facial motion region, the $k^{th}$ exemplar at time $t$. To weight the current data sample appropriately against history, we consider the data samples under an exponential envelope located at the current time as in [70], $F_t(j) = \alpha e^{-(t-j)/\tau}$, for $j \leq t$. Here, $\alpha = 1 - e^{-1/\tau}$.

For the GMM model of certain $(q,k,t)$ value, suppose the GMM has $M$ components denoted by $\{N(\mu_1, \Sigma_1), ..., N(\mu_M, \Sigma_M)\}$. Here the $q, k, t$ subscripts are dropped for simplicity. Given an adaptation data sample $Y = \{Y_S, Y_T\}$, the ML estimate of the translation $\Delta \mu$ can be computed by solving equation (7.9) according to [50]:

$$\sum_{m=1}^{M} \gamma_m \Sigma_m^{-1} Y_T \mu_m^T = \sum_{m=1}^{M} \gamma_m \Sigma_m^{-1} (\mu_m + \Delta \mu) \mu_m^T$$  \hspace{1cm} (7.9)$$

where $\gamma_m$ is GMM component occupancy probability defined as the probability that $Y_T$ draws from the $m^{th}$ component of the GMM given $Y_T$ draws from this GMM. $Y_T$ is the texture feature of the current adaptation data. A closed-form solution for equation (7.9) is feasible when $\Sigma_m$ is diagonal. The $i^{th}$ element of $\Delta \mu$ can be computed as

$$\Delta \mu_i = \frac{\sum_{m=1}^{M} \gamma_m \sigma_{m,i} Y_{Ti} - \sum_{m=1}^{M} \gamma_m \sigma_{m,i} \mu_{m,i}}{\sum_{m=1}^{M} \gamma_m \sigma_{m,i}}$$  \hspace{1cm} (7.10)$$

where $\sigma_{m,i}$ is the $i^{th}$ diagonal element of $\Sigma_m^{-1}$, and $Y_{Ti}$ is the $i^{th}$ element of $Y_T$. 

85
The probability weighted average of the translation vector up to time $t$, for the $k^{th}$ exemplar of a certain facial motion region is then

$$\bar{\Delta \mu}_{k,t} = \frac{1}{\beta_{k,t}} \left( \sum_{j=-\infty}^{t} F_t(j) o_{k,t}(Y_j) \Delta \mu_{k,j} \right)$$ (7.11)

where $\beta_{k,t}$ is a normalization factor defined as $\beta_{k,t} = \sum_{j=-\infty}^{t} F_t(j) o_{k,t}(Y_j)$, and $\Delta \mu_{k,j}$ is computed using equation (7.10). Equation (7.11) can be rewritten in a recursive manner as:

$$\Delta \mu_{k,t} = \frac{(1 - \alpha) \beta_{k,t-1} \Delta \mu_{k,t-1} + \alpha o_{k,t}(Y_t) \Delta \mu_{k,t}}{\beta_{k,t}}$$ (7.12)

where $\beta_{k,t} = (1 - \alpha) \beta_{k,t-1} + \alpha o_{k,t}(Y_t)$. Equations (7.10) and (7.12) are applied to updated the translation vectors of each facial motion region.

In an online EM algorithm, the training samples can not be used iteratively for optimization. Thus, errors made by the initial model may cause the adaptation method to be unstable. Note that the chosen exemplars are common semantic symbols which have their intrinsic structure. Therefore, when adapting the exemplar model to a particular person, the translations of the mean vectors should be constrained rather than random. For the exemplars model of a facial motion region at time $t$, let $\xi$ denote the translation vector constructed by concatenating the translations of all the mean vectors, i.e., $\xi = [\Delta \mu_1^T \ldots \Delta \mu_K^T]^T$, where $K$ is the number of exemplars. In our algorithm, $\xi$ is a 72-dimension vector. We impose the constraint that the vector $\xi$ should lie in certain low dimensional subspaces. To learn such a low dimensional subspace, we first learn a person-independent exemplar model from exemplars of many people. Then a person-dependent exemplar model is learned for each person. We collect a training sample set $\{\xi\}$, consisting of the translation vectors between the mean vectors of person-independent model and person-dependent models. Finally, PCA is applied on the set $\{\xi\}$. Principal orthogonal components which account for major variation in set $\{\xi\}$ are chosen to span a low dimensional subspace. A translation vector can be projected
to the learned subspace by

\[ a = W^T (\xi - \bar{\xi}) \quad (7.13) \]

and the new constrained translation vector can be reconstructed as

\[ \hat{\xi} = Wa + \bar{\xi} \quad (7.14) \]

where \( W \) is the matrix consisting of principal orthogonal components, \( \bar{\xi} \) is the mean translation vector of the vectors in \( \{\xi\} \).

In summary, the online EM-based adaptation algorithm is as follows:

- **E-step**: compute exemplar ownership probabilities \( o_{k,t}(Y_t) \) based on equations (7.5), (7.6) and (7.8).

- **M-step**: for the \( q^{th} \) facial motion region, estimate the translator vector \( \Delta \mu_{q,k,t} \) based on equations (7.10) and (7.12). Then we construct the vector \( \xi_{q,t} \) and project it using equation (7.13). Finally, the constrained estimate of \( \hat{\xi}_{q,t} \) is given by equation (7.14).

### 7.2 Facial Expression Recognition Experiment

#### Results

We evaluate the efficacy of the proposed hybrid motion analysis method by using the extracted features in a facial expression classifications task. The public available CMU Cohn-Kanade expression database [72] is used. From the database, we selected 47 subjects who has at least 4 coded expression sequences. Overall the selected database contains 2981 frames. There are 72% female, 28% male, 89% Euro-American, 9% Afro-American, and 2% Asian. Several different lighting conditions are present in the selected database. The image size of all the data is \( 640 \times 480 \). For the Cohn-Kanade database, Tian and Bolle [132] achieved a high neutral face detection rate using geometric features only. That indicates the database
does not contain expressions with little geometric motion yet large texture variation. Using
geometric feature only on the database, Cohen et al. [27] reported good recognition results
for happiness and surprise, but much more confusion among anger, disgust, fear and sadness.
In this section, we present our experimental results showing the proposed method improves
the performance for these four expressions.

We select seven exemplars including six expressions and neutral. The six expressions are
anger, disgust, fear, happiness, sadness, and surprise. In our experiments, we first assign
neutral vs. non-neutral probability using a neutral network similar to [132], which achieved a
recognition rate of 92.8% for neutral. For the remaining exemplars, we use 4 components for
each GMM model. The tracking results are used to perform facial expression classification
as $\hat{X}_t = \arg\max p_t(X_t)$. Although this classifier may not be as good as more sophisticated
classifiers such as those in [3, 27, 41, 158], it can be used as a test-bed to measure the relative
performances of different features and the proposed adaptation algorithm.

In the first experiment, we compare the classification performances of using geometric fea-
ture only and using both geometric and ratio-image-based appearance features. We use 60%
data of each person as training data and the rest as test data. Thus it is a person-dependent
test. For all experiments we have done, geometric-feature-only method and hybrid-feature
method give similar results for “happiness” and “surprise”. That means these two expres-
sions have distinct geometric features so that appearance features are not crucial for them.
This observation is consistent with [27]. Therefore, in this section we choose to omit the
results for “happiness” and “surprise” for conciseness.

The confusion matrix for the geometric-feature-only method is presented in Table 7.1. In
Table 7.2, we show the confusion matrix for the method using both appearance and geometric
features. Note that we omit the rows of “happiness” and “surprise” for conciseness. The
analysis of the confusion between different expressions shows that the proposed appearance
features help to significantly reduce the confusion among expressions, especially the four more
easily confused expressions: anger, disgust, fear and sadness. For example, the confusion
rate between “anger” and “disgust” is reduced more than 10 percent. This improvement demonstrate the effectiveness of the proposed appearance features in handling motion details.

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Sadness</th>
<th>Happiness</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anger</td>
<td>74.8%</td>
<td>13.3%</td>
<td>0%</td>
<td>10.6%</td>
<td>0.7%</td>
<td>0.7%</td>
</tr>
<tr>
<td>Disgust</td>
<td>16.3%</td>
<td>76.5%</td>
<td>3.1%</td>
<td>4.1%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Fear</td>
<td>0%</td>
<td>0.8%</td>
<td>65.6%</td>
<td>5.0%</td>
<td>22.7%</td>
<td>5.9%</td>
</tr>
<tr>
<td>Sadness</td>
<td>13.4%</td>
<td>0.6%</td>
<td>2.6%</td>
<td>77.1%</td>
<td>0%</td>
<td>6.4%</td>
</tr>
</tbody>
</table>

Table 7.1: Person-dependent confusion matrix using the geometric-feature-only method

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Sadness</th>
<th>Happiness</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anger</td>
<td>92.7%</td>
<td>2.0%</td>
<td>0%</td>
<td>4.6%</td>
<td>0%</td>
<td>0.7%</td>
</tr>
<tr>
<td>Disgust</td>
<td>4.1%</td>
<td>85.7%</td>
<td>6.1%</td>
<td>3.1%</td>
<td>1.0%</td>
<td>0%</td>
</tr>
<tr>
<td>Fear</td>
<td>0%</td>
<td>1.7%</td>
<td>81.5%</td>
<td>0%</td>
<td>12.6%</td>
<td>4.2%</td>
</tr>
<tr>
<td>Sadness</td>
<td>3.8%</td>
<td>0%</td>
<td>3.2%</td>
<td>90.5%</td>
<td>0%</td>
<td>2.5%</td>
</tr>
</tbody>
</table>

Table 7.2: Person-dependent confusion matrix using both geometric and appearance features

In Table 7.3, we show the comparison of the recognition rates for the four easily confused expressions. In order to show the statistical significance of our results, we also present the 95% confidence intervals. A more intuitive comparison is illustrated in Figure 7.4.

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Sadness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geo-only</td>
<td>74.8 ± 3.2 %</td>
<td>76.5 ± 2.7 %</td>
<td>65.6 ± 3.2 %</td>
<td>77.1 ± 2.9 %</td>
</tr>
<tr>
<td>Proposed</td>
<td>92.7 ± 3.3 %</td>
<td>85.7 ± 3.0 %</td>
<td>81.5 ± 3.6 %</td>
<td>90.5 ± 3.1 %</td>
</tr>
</tbody>
</table>

Table 7.3: Comparison of the proposed approach with geometric-only method in person-dependent test together with their 95% confidence intervals.

In the second experiment, we compare the classification performances of ratio-image based appearance feature and non-ratio-image based appearance feature. The goal of this test is to see whether the proposed appearance feature is less person-dependent. The non-ratio-image based feature does not consider the neutral face texture, and is computed as $G(\log(I'(u, v)))$ instead of using equation (7.2). To show the advantage of ratio-image based feature in the ability to generalize to new people, the test is done in a person-independent
Figure 7.4: Comparison of the proposed approach with geometric-only method in person-dependent test.

That is, all data of one person is used as test data and the rest as training data. This test is repeated 47 times, each time leaving a different person out (leave one out cross validation). The person-independent test is more challenging because the variations between subjects are much larger than those within the same subject. To factor out the influence of geometric feature, only the appearance feature is used for recognition in this experiment. The average recognition rates are shown in Table 7.4 and Figure 7.5. We can see that ratio-image based feature outperforms non-ratio-image based feature significantly. For individual subject, we found that the results of the two features are close when the texture does not have much details. Otherwise, ratio-image based feature is much better.

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Sadness</th>
</tr>
</thead>
<tbody>
<tr>
<td>ratio</td>
<td>37.0 ± 3.2 %</td>
<td>59.6 ± 3.1 %</td>
<td>35.7 ± 3.4 %</td>
<td>41.8 ± 3.2 %</td>
</tr>
<tr>
<td>non-ratio</td>
<td>24.7 ± 2.9 %</td>
<td>22.1 ± 2.8 %</td>
<td>24.4 ± 3.2 %</td>
<td>15.6 ± 2.8 %</td>
</tr>
</tbody>
</table>

Table 7.4: Comparison of the proposed appearance feature (ratio) with non-ratio-image based appearance feature (non-ratio) in person-independent recognition test.

The third experiment again uses person-independent setting and leave one out cross validation. For each test, we use 50% of the data of the test person as adaptation data and the rest as test data. Without applying adaptation algorithm, we first compare the performances of using geometric feature only and using both geometric and ratio-image-
based appearance features. The results are shown in the rows (a) and (b) of Table 7.5. It can be observed that improvement is less significant than that in the first experiment. This is mainly due to the individual variations in facial expressions. Then, we test the performance of the proposed online EM-based adaptation algorithm. Only the models of the four easily confused expressions are adapted. In each test, we apply PCA to the training data. The first 11 principal components are selected which account for about 90% of total variations. The adaptation is online and unsupervised, without using the labels of the adaptation data. We choose $\alpha = 0.1$ for fast adaptation because the amount of the adaptation data is limited. The recognition rates the adaptation are shown in the row (d) of Table 7.5. We can see the adaptation algorithm improves the recognition rates. For comparison, we also show in the row (c) the recognition rates of adaptation without the PCA subspace constraints. It can be seen that the unconstrained adaptation is not stable. The performance with unconstrained adaptation could sometimes be worse than performance without adaptation. For the four expressions, the maximum 95% confidence intervals of all methods are $\pm 3.8\%$, $\pm 3.1\%$, $\pm 3.8\%$, and $\pm 4.3\%$, respectively. Figure 7.6 gives a more intuitive comparison of the four methods.
Table 7.5: Comparison of different algorithms in person-independent recognition test. (a): Algorithm uses geometric feature only. (b): Algorithm uses both geometric and ratio-image based appearance feature. (c): Algorithm applies unconstrained adaptation. (d): Algorithm applies constrained adaptation.

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Sadness</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>66.6%</td>
<td>65.3%</td>
<td>60.8%</td>
<td>69.8%</td>
</tr>
<tr>
<td>b</td>
<td>70.7%</td>
<td>70.2%</td>
<td>64.6%</td>
<td>72.5%</td>
</tr>
<tr>
<td>c</td>
<td>75.3%</td>
<td>59.2%</td>
<td>64.0%</td>
<td>73.1%</td>
</tr>
<tr>
<td>d</td>
<td>77.9%</td>
<td>78.1%</td>
<td>67.7%</td>
<td>77.8%</td>
</tr>
</tbody>
</table>

To test the proposed method under large 3D rigid motions and novel lighting conditions, we also collect two video sequences of a subject who is not in the training database. The frame size of the videos is 640 × 480. The first video has 763 frames and contains substantial global face translation and rotation. The second video has 435 frames and is taken under a lighting condition dramatically different from the rest of data. We manually label the image frames using the seven categories as the ground truth. Two snapshots for each sequence are...
shown in Fig. 7.7 and 7.8. The corresponding recognition results are also illustrated using one of the training example. We compare the expression recognition rates of the proposed method with geometric-feature-only method. The overall average recognition rate of our method is 71%, while the rate of the geometric-only method is 59%. Part of the tracking results is visualized in the accompanying videos [147]. In the videos, the upper left of the frame is the input video frame, the upper right is the geometric feature based tracking visualized by a yellow mesh. The exemplar $\hat{X}_t$ is shown on the bottom. We can observe that our method can still track the texture variations when there are large 3D motions, or under dramatically different lighting conditions.

Figure 7.7: The results under different 3D poses. For both (a) and (b): Left: cropped input frame. Middle: extracted texture map. Right: recognized expression.
7.3 Enhance Geometric Tracking with Appearance Model

The hybrid face motion analysis approach presented in Section 7.1 and 7.2 require robust geometric motion estimation such that the stabilized face texture can be correctly extracted. Our extensive experiences of applying our geometric tracker to various conditions show that it is not sensitive to changes of conditions. However, one exception is when the resolution of face is small, e.g. smaller than 60 × 60 pixels. In this section, we present a method to use appearance cues to improve geometric tracking in low resolutions. We first describe the problems of geometric tracking under low resolutions in Section 7.3.1. Then, we present our appearance-based enhancement in Section 7.3.2. Finally, we give experimental results in Section 7.3.3.
7.3.1 Problems with geometric tracking

The performance of the geometric facial motion tracking in our framework relies on optical flow. Optical flow assumes that the intensities of image patches are constant over short intervals. In implementations, for an image patch in the current frame, optical flow tries to find a patch with the closest match of local texture in the next frame. In practice, the optical flow performance is good when an image patch has rich local texture which can distinguish it from its neighborhood.

When the resolution of the faces become smaller, if we scale the size of the patches accordingly, each patch will contain fewer pixels so that the matching of patches is less reliable. Alternatively, if we keep the size of patches unchanged, each patch will cover a larger area of faces. As each patch contains more complex geometric shapes, self-occlusions and non-rigid motions are more likely to occur within it. Thus, the intensity constancy assume of optical flow is more likely to be violated.

In current implementation of optical flow in our geometric tracking, the fixed size of patches is used. Based on our analysis, optical flow performance will become worse as the face resolution gets lower, and therefore degrade the geometric facial motion analysis performance.

7.3.2 Proposed appearance-based enhancement

Compared to geometric methods, appearance-based approaches use constraints of the overall facial appearance. Thus at low resolution, their performance could be better than geometric methods because their assumptions are less violated by changes of resolutions. However, a flexible appearance-based approach would need to deal with the variations across different people and illuminations. La Cascia et al. [22] proposed a reliable appearance-based method for rigid face tracking. In our framework, we combine an appearance-based method similar to [22] with our geometric tracking to improve performance for rigid face tracking in low
resolutions.

In [22], face tracking is formulated as an image registration problem in the 3D face model’s texture map image. The parameters of registration include motion and illumination parameters. Let $T, T_0$ be the stabilized face texture in the current and initial frame. Assuming the inter-frame motion is small and the face texture is roughly registered using the parameters of the previous frame, the difference between $T$ and $T_0$ should be small. La Cascia et al. [22] approximate the texture differences using a linear model, where the texture differences are a linear combination of perturbation basis. Each perturbation base corresponds to one type of 3D motion or illumination changes. Using the notation in [22], we have

$$T - T_0 \approx Bq + Uc$$

(7.15)

In equation 7.15, the columns of the matrix $B = [b_1; b_2; ...; b_M]$ contains the pixels value changes due to 3D rigid motions. For example, $b_1$ corresponds to pixel value changes caused by small perturbation in $X$ axis translation. The columns of $B$ are called warping templates. They can be derived as the difference between the initial face texture $T_0$ and the texture of 3D face model warped using the perturbation parameter. $q$ is the vector of the coefficients that specify the intensities of warping templates. Similarly, the columns of the matrix $U = [u_1; u_2; ...; u_N]$ contains the pixels value changes due to illumination variations. They are called illumination templates and $c$ is the vector of the coefficients for the linear combination. During tracking, coefficients $q$ and $c$ are estimated by solving the linear system using least square method. 3D rigid motions are then specified by $q$.

In [22], the illumination templates are estimated empirically from extensive training data. In our framework, we can derive these templates analytically and efficiently using the REM illumination model in the flexible appearance model. In this way, the training phase can be avoided and the computation load is reduced. Specifically, only 9 spherical harmonic basis are proven to be good enough to approximate the diffuse appearance of faces. Based on
equation 6.10, the value of the illumination templates at a face surface point with normal $\mathbf{n}$ is given by

$$u_{lm}(\mathbf{n}) \approx Y_{lm}(\mathbf{n}), \quad (0 \leq l \leq 2, -l \leq m \leq l)$$

where the normal $\mathbf{n}$ is know for each face pixel since a 3D face model has been aligned with the first frame. The formulae of $Y_{lm}(\mathbf{n})$ are given by equation 6.6. Further details on how to compute the illumination basis can be found in Chapter 6, Section 6.1.

When there are large inter-frame motion or non-rigid motion, the linear approximation in equation 7.15 is less accurate. To deal with large inter-frame motion, we register $T$ using both the previous-frame motion parameters and current-frame motion parameters estimated by geometric tracking. Then, we can estimate two sets of motion parameters from equation 7.15. Finally, we warp the face model using the two sets of parameters and update the texture residual $T - T_0$. We choose the set of parameters with smaller texture residual as the final motion estimation. The texture residual is defined as the average absolution pixel difference

$$\frac{1}{K} \sum_{i=1}^{K} |T_{(i)} - T_{0(i)}|$$

where $T_{(i)}$ is the $i$-th pixel of texture $T$ and $K$ is the number of pixels. This approach is more reliable when there are large inter-frame motion because our geometric tracking uses a pyramid-based approach. To deal with appearance changes due to non-rigid motions, we weight each pixel in face texture image by multiplying a coefficient between 0 and 1. In Figure 7.3, we have pre-defined regions where non-rigid motions are mostly likely to occur. For pixels within these regions, we assign small weights close to 0. Otherwise, the pixel mostly undergo rigid motions, the weight is set to be 1.
7.3.3 Results

We implement the appearance-based enhancement of geometric tracking in our geometric tracking software using Microsoft Visual C++. Readers can refer to [22] for further implementation details. We test the rigid tracking performances on 20 videos used in [22] and our own video illustrated in Figure 7.7. The videos in CMU Cohn-Kanade database and our video in Figure 7.8 are not used because there are no 3D rigid motions. These videos are sub-sampled so that the resolutions of faces are below 60 × 60.

We choose the un-weighted texture residual as the metric to measure the tracking performance. The smaller the texture residual, the better the performance. In our experiments, the metric agrees with subjective judgement of users based on whether or not the tracking mask is following the face.

In our experiments, both original geometric tracker and enhanced tracker perform well when there are only either pure translation or pure rotation. The enhanced tracker is only slightly better in that condition. However, when there are simultaneous 3D rotation and translation (which is more realistic condition), the improvement of enhanced tracker is significant. Next, we show the tracking results of several representative videos.

Figure 7.9 shows the tracking results on the sub-sampled video of that illustrated in Figure 7.7. The video contains simultaneous 3D translation, rotation and non-rigid motions. The horizontal axis in the figure represents frame number. The vertical axis represents the texture residual. The blue solid trajectory is the tracking results using appearance-based enhancement, and the red dotted trajectory is the results using geometric tracking only. We can see that the original geometric tracker loses track after 200 frames, while the enhanced tracker can still track until the end.

Figure 7.10 shows the results on another video with simultaneous 3D translation and rotation. It is the sub-sampled version of one of video used in [22]. Similarly, we can observe the original geometric tracker loses tracker after about 100 frames while the enhanced tracker
Figure 7.9: The tracking results on a sub-sampled video of video.

Figure 7.10: The tracking results on a sub-sampled video with simultaneous 3D rotation and translation. Original video is from [22].

does not. The improvement of the enhanced tracking is significant.

Figure 7.10 shows results on a video with pure 3D rotation. It is also the sub-sampled version of one of video used in [22]. We can see the improvement of the enhanced tracker is not significant in this condition.

On the average, the improvement of the enhanced tracker in terms of texture residual is 5.1. For the 5 videos out the 21 videos with simultaneous rotation and translation, the average texture residual improvement is 20.4.
Figure 7.11: The tracking results on a sub-sampled video with pure 3D rotation. Original video is from [22].

7.4 Summary

In this chapter, a hybrid facial motion analysis scheme is presented. In this scheme, we propose a novel appearance feature based on our flexible appearance model. Compared to most existing appearance features, the novel appearance feature is less illumination dependent and less person dependent. An online appearance model adaptation scheme is also introduced to improve the performance in changing conditions. A facial expression recognition experiment has been conducted to demonstrate the efficacy of the proposed scheme in changing conditions. The recognition performance is improved because of the reduced dependency of the novel appearance feature and the online adaptation scheme.

To improve the geometric tracking (i.e. texture registration) under low resolutions, an appearance-based enhancement is proposed in the second half of the chapter. It adds an additional appearance-based constraint which minimizes texture residual. Experiments show that the enhanced tracker is less likely to lose track when there are simultaneous rotation and translation in low resolution face videos. The improved robustness of the geometric tracking can lead to more robust appearance-based motion analysis which is based on the geometric tracking results. In other words, a closer coupling of the geometry and appearance
processing improves the overall performance.
Chapter 8

Face Appearance Synthesis Using Flexible Appearance Model

In this chapter, we discuss the face appearance synthesis using flexible appearance model described in Chapter 6. Our goal is to use appearance model in a flexible way to synthesize plausible facial appearance variations caused by lighting and motion. The synthesized appearance can provide visual cues in synthetic face based interactions. For this purpose, the appearance models need to be applicable for different lighting conditions and people. Because we aim at reproducing important visual cues related to facial motion, we neglect less essential phenomena such as specular reflection and assume faces are Lambertian.

In Section 8.1, we first describe how to synthesize the illumination effects of neutral face appearance in different lighting conditions. Next, we describe works on using the face relighting technique for face recognition under varying lighting conditions 8.2. In Section 8.3 the synthesis of facial appearance caused by motion is discussed.

8.1 Neutral Face Relighting

We present a ratio-image based technique [148, 144] to use a radiance environment map to render diffuse objects with different surface reflectance properties. This method has the advantage that it does not require the separation of illumination from reflectance, and it is simple to implement and runs at interactive speed. In order to use this technique for human
face relighting, we have developed a technique that uses spherical harmonics to approximate
the radiance environment map for any given image of a face. Thus we are able to relight face
images when the lighting environment rotates. Another benefit of the radiance environment
map is that we can interactively modify lighting by changing the coefficients of the spherical
harmonics basis. Finally we can modify the lighting condition of one person’s face so that it
matches the new lighting condition of a different person’s face image assuming the two faces
have similar skin albedos.

8.1.1 Relighting with radiance environment maps

Using the technique described in Section 6.1.1 of Chapter 6, the radiance environment map
(REM) based illumination model can be approximated from a single face image. Based on
the illumination model, we can then synthesize face image in novel lighting environments
in various scenarios, including: (1) face rotating in the same lighting environment; (2) il-
illumination effects transfer from one face to another; and (3) interactively lighting effects
editing.

Relighting when rotating in the same lighting condition

When an object is rotated in the same lighting condition, the intensity of the object will
change due to incident lighting changes. let \( E(\vec{n}) \) denote the irradiance defined by equa-
tion 6.1. For any given point \( p \) on the object, suppose its normal is rotated from \( \vec{n}_a \) to \( \vec{n}_b \).
Assuming the object is diffuse, and let \( \rho_p \) denote the reflectance coefficient at \( p \), then the
intensities at \( p \) before and after rotation are respectively:

\[
I_{object}(\vec{n}_a) = \rho_p E(\vec{n}_a) \tag{8.1}
\]

\[
I_{object}(\vec{n}_b) = \rho_p E(\vec{n}_b) \tag{8.2}
\]
From equation 8.1 and 8.2 we have

\[
\frac{I_{\text{object}}(\vec{n}_b)}{I_{\text{object}}(\vec{n}_a)} = \frac{E(\vec{n}_b)}{E(\vec{n}_a)} \tag{8.3}
\]

From the definition of the radiance environment map (equation 6.2), we have

\[
\frac{I_{\text{sphere}}(\vec{n}_b)}{I_{\text{sphere}}(\vec{n}_a)} = \frac{E(\vec{n}_b)}{E(\vec{n}_a)} \tag{8.4}
\]

Comparing equation 8.3 and 8.4, we have

\[
\frac{I_{\text{sphere}}(\vec{n}_b)}{I_{\text{sphere}}(\vec{n}_a)} = \frac{I_{\text{object}}(\vec{n}_b)}{I_{\text{object}}(\vec{n}_a)} \tag{8.5}
\]

What this equation says is that for any point on the object, the ratio between its intensity after rotation and its intensity before rotation is equal to the intensity ratio of two points on the radiance environment map. Therefore given the old intensity \(I_{\text{object}}(\vec{n}_a)\) (before rotation), we can obtain the new intensity \(I_{\text{object}}(\vec{n}_b)\) from the following equation:

\[
I_{\text{object}}(\vec{n}_b) = \frac{I_{\text{sphere}}(\vec{n}_b)}{I_{\text{sphere}}(\vec{n}_a)} \cdot I_{\text{object}}(\vec{n}_a) \tag{8.6}
\]

Note that the REM \(I_{\text{sphere}}\) could be approximated from a single input face image.

**Comparison with inverse rendering approach**

It is interesting to compare our method with the inverse rendering approach. Here we take the face rotating scenario as an example. To use inverse rendering approach, we can capture the illumination environment map, and use spherical harmonics technique [118] to obtain \(E(\vec{n})\): the diffuse components of the irradiance environment map. The reflectance coefficient
\( \rho_p \) at point \( p \) can be resolved from its intensity before rotation and the irradiance, that is,

\[
\rho_p = \frac{I_{\text{object}}(\vec{n}_a)}{E(\vec{n}_a)} \quad (8.7)
\]

After rotation, its intensity is equal to \( \rho_p E(\vec{n}_b) \). From equation (8.7) and 8.4,

\[
\rho_p E(\vec{n}_b) = I_{\text{object}}(\vec{n}_a) \frac{E(\vec{n}_b)}{E(\vec{n}_a)}
= \frac{I_{\text{sphere}}(\vec{n}_b)}{I_{\text{sphere}}(\vec{n}_a)} \cdot I_{\text{object}}(\vec{n}_a)
\]

Thus, as expected, we obtain the same formula as equation (8.6).

The difference between our approach and the inverse rendering approach is that our approach only requires a radiance environment map, while the inverse rendering approach requires the illumination environment map (or irradiance environment map). In some cases where only limited amount of data about the lighting environment is available (such as a few photographs of some diffuse objects), it would be difficult to separate illuminations from reflectance properties to obtain illumination environment map. Our technique allows us to do image-based relighting of diffuse objects even from a single image.

**Relighting in different lighting conditions**

Let \( \mathcal{L} \), \( \mathcal{L'} \) denote the old and new lighting distributions respectively. Suppose we use the same material to capture the radiance environment maps for both lighting conditions. Let \( I_{\text{sphere}} \) and \( I'_{\text{sphere}} \) denote the radiance environment maps of the old and new lighting conditions, respectively. For any point \( p \) on the object with normal \( \vec{n} \), its old and new intensity values are respectively:

\[
I_{\text{object}}(\vec{n}) = \rho_p \int_{\Omega(\vec{n})} \mathcal{L}(\omega)(\vec{n} \cdot \omega) d\omega \quad (8.8)
\]

\[
I'_{\text{object}}(\vec{n}) = \rho_p \int_{\Omega(\vec{n})} \mathcal{L'}(\omega)(\vec{n} \cdot \omega) d\omega \quad (8.9)
\]
Together with equation 6.1 and 6.2, we have a formula of the ratio of the two intensity values

\[
\frac{I'_{\text{object}}(\vec{n})}{I_{\text{object}}(\vec{n})} = \frac{I'_{\text{sphere}}(\vec{n})}{I_{\text{sphere}}(\vec{n})}
\]  

(8.10)

Therefore the intensity at \( p \) under new lighting condition can be computed as

\[
I'_{\text{object}}(\vec{n}) = \frac{I'_{\text{sphere}}(\vec{n})}{I_{\text{sphere}}(\vec{n})} \cdot I_{\text{object}}(\vec{n})
\]  

(8.11)

**Interactive face relighting**

In [118] spherical harmonic basis functions of irradiance environment map were visualized on sphere intuitively, which makes it easy to modify lighting by manually changing the coefficients. Our radiance environment map is the irradiance environment map scaled by constant albedo. We can modify the coefficients in equation (6.7) to interactively create novel radiance environment maps. Then these radiance environment maps can be used to edit the lighting effects of the face appearance. Unlike [118], we do not need to know the face albedo.

8.1.2 Face relighting from a single image

Given a single photograph of a person’s face, it is possible to approximate the radiance environment map, using the technique described in Section 6.1.1 of Chapter 6,

Given a photograph and a generic 3D face geometry, we first align the face image with the generic face model. Note that if the input image is a face texture image, it is already aligned with geometry. From the aligned photograph and the 3D geometry, we use the method described in Section 6.1.1 of Chapter 6 to approximate the radiance environment map. To relight the face image under rotated lighting environment, we compute each face pixel’s normal (with respect to the lighting environment) before and after rotation. Then we compute ratio \( \frac{I'_{\text{sphere}}(\vec{n}_b)}{I_{\text{sphere}}(\vec{n}_a)} \), where \( \vec{n}_b \) and \( \vec{n}_a \) are the new and old normal vectors of the pixel.
respectively, and $I_{\text{sphere}}$ is the approximated radiance environment map. Finally the new intensity of the pixel is given by equation (8.6).

If we are given photographs of two people’s faces under different lighting conditions, we can modify the first photograph so that it matches the lighting condition of the second face. We first compute the radiance environment map for each face. If the two faces have the same average albedo, then the two radiance environment maps have the same albedo, and we can apply equation (8.11) to relight the first face to match the second face’s lighting condition. In practice, if two people have similar skin colors, we can apply equation (8.11) to relight one face to match the lighting condition of the other.

Given one input face photograph and a user interface to edit the 9 coefficients of the radiance environment map, we can interactively synthesize novel illumination effects on the face image based on technique described in Section 8.1.1.

**Dynamic range of images**

Because digitized image has limited dynamic range, ratio-based relighting would have artifacts where skin pixel values are too low or nearly saturated. To alleviate the problem, we apply constrained texture synthesis for these pixels. Our assumption is that the high frequency face albedo, similar to texture, contains repetitive patterns. Thus we can infer local face appearance at the places of artifacts from examples on other part of the face. We first identify these pixels as outliers that do not satisfy Lambertian model using robust statistics [61]. Then we use the remaining pixels as example to synthesize texture at the place of outliers. We use a patch-based Image Analogy algorithm [62], with the constraint that a candidate patch should match the original patch up to a relighting scale factor. Since we use a patch-based approach and we only apply it to the detected outlier regions, the computation overhead is very small. Figure 8.1 shows an example where the low dynamic range of the image causes artifacts in the relighting. (a) is the input image, whose blue channel (b) has very low intensity on the person’s left face. (c) is the relighting result without using the
Figure 8.1: Using constrained texture synthesis to reduce artifacts in the low dynamic range regions. (a): input image; (b): blue channel of (a) with very low dynamic range; (c): relighting without synthesis; and (d): relighting with constrained texture synthesis. We can see that almost all the unnatural color on the person’s left face in (c) disappear in (d).

Specular reflection in face images

Specular reflections of faces are currently not handled by our spherical harmonics based radiance environment map. Marschner et al. [90] reported that the reflections of human face is primarily Lambertian when the lighting incidence angles are small; but strong specular reflection as well as subsurface scattering would emerge at large incidence angles. Therefore, specular reflection could be strong in certain input faces images.

In our experiments, we observed that the specular reflections in the input face images may or may not produce annoying visual results, depending on the difference of the input and target lighting conditions. If multiples input images of the same face (with the same pose) in different illuminations are available, existing techniques for separating diffuse and specular reflection could be applied, such as techniques of Lin and Shum [85] and Nishino et al. [102].

In our framework, if only a single input face image is available, we alleviate the problems
caused by specular reflections using a approach similar to that described in Section 8.1.2. Using the techniques for detecting regions with poor dynamic range, we can detect outliers which do not satisfy our Lambertian model. These outliers would include pixels with strong specular reflections. After that, we use constrained texture synthesis as described in Section 8.1.2 to synthesize plausible diffuse appearance at the places of these outliers. For the example presented in Section 8.1.2, Figure 8.2(b) shows the detected outlier pixels whose intensity values higher than the values computed from our Lambertian model. Many such outliers correspond to pixels with strong specular reflection. In Figure 8.3, it can be seen that relighting the proposed constrained texture synthesis also reduce the specular reflection in the synthesized image. For better visualization purpose, pixels intensities in both Figure 8.2(b) and Figure 8.3(c) are scaled.

8.1.3 Implementation

In our implementations, we use a cyberware-scanned face mesh as our generic face geometry (shown in Figure 8.4(a)). All the examples reported in this paper are produced with this
Figure 8.3: Using constrained texture synthesis to reduce specular reflections: (a): relighting without synthesis; (b): relighting with constrained texture synthesis; (c): the pixel difference between (a) and (b). (The differences in low intensity areas are masked out).

mesh. Given a 2D image, to create the correspondence between the vertices of the mesh and the 2D image, we first create the correspondence between the feature points on the mesh and the 2D image. The feature points on the 2D image are marked manually as shown in Figure 8.4(b). We are working on automatically detecting these facial features using techniques presented in [67]. We use image warping technique to generate the correspondence between the rest of the vertices on the mesh and the 2D image. A mapping from the image pixels to a radiometrically linear space is implemented to account for gamma correction. We have developed user interface for this face relighting software. It is illustrated in Figure 8.5.

The computation of the radiance environment map takes about one second for a 640x480 input image on a PC with Pentium III 733 MHz and 512 MB memory. The face relighting is currently implemented completely in software without hardware acceleration. It runs about 10 frames per second.
8.1.4 Relighting results

We show some relighting experiment results on 2D face images in this section. The first example is shown in Figure 8.6 where the middle image is the input image. The radiance environment map is shown below the input image. It is computed by assuming the back is dark since the lights mostly come from the frontal directions. The rest of the images in the sequence show the relighting results when the lighting environment rotates. Below each image, we show the corresponding rotated radiance environment map. The environment rotates about 45° between each two images, a total of 180° rotation. The accompanying videos show the continuous face appearance changes, and the radiance environment map is shown on the righthand side of the face. (The environment rotates in such a direction that the environment in front of the person turns from the person’s right to the person’s left). From the middle image to the right in the image sequence, the frontal environment turns to the person’s left side. On the fourth image, we can see that part of his right face gets darker. On the last image, a larger region on his right face becomes darker. This is consistent with the rotation of the lighting environment.

Figure 8.7 shows a different person in a similar lighting environment. For this example,
we have captured the ground truth images at various rotation angles so that we are able to do a side-by-side comparison. The top row images are the ground truth images while the images at the bottom are the synthesized results with the middle image as the input. We can see that the synthesized results match very well with the ground truth images. There are some small differences mainly on the first and last images due to specular reflections (According to Marschner et al. [90], human skin is almost Lambertian at small light incidence angles and has strong non-Lambertian scattering at higher angles).

The third example is shown in Figure 8.8. Again, the middle image is the input. In this example, the lights mainly come from two sides of the face. The bright white light on the person’s right face comes from sky light and the reddish light on his left face comes from the sun light reflected by a red-brick building. The image sequence shows a 180° rotation of the lighting environment.

Figure 8.9 shows four examples of interactive lighting editing by modifying the spherical harmonics coefficients. For each example, the left image is the input image and the right
image is the result after modifying the lighting. In example (a), lighting is changed to attach shadow on the person’s left face. In example (b), the light on the person’s right face is changed to be more reddish while the light on her left face becomes slightly more blueish. In (c), the bright sunlight move from the person’s left face to his right face. In (d), we attach shadow to the person’s right face and change the light color as well. Such editing would be difficult to do with the currently existing tools such as Photoshop™.

We have also experimented with our technique to relight a person’s face to match the lighting condition on a different person’s face image. As we pointed out in Section 8.1.2, our method can only be applied to two people with similar skin colors. In Figure 8.10(a), we relight a female’s face shown on the left to match the lighting condition of a male shown in the middle. The synthesized result is shown on the right. Notice the darker region on the right face of the middle image. The synthesized result shows similar lighting effects. Figure 8.10(b) shows an example of relighting a male’s face to a different male. Again, the left and middle faces are input images. The image on the right is the synthesized result. From the middle image, we can see that the lighting on his left face is a lot stronger than the lighting on his right face. We see similar lighting effects on the synthesized result. In addition, the dark region due to attached shadow on the right face of the synthesized image closely matches the shadow region on the right face of the middle image.
Face recognition has important applications in human computer interaction and security. Face recognition still remains a challenging problem because the performance of almost all current face recognition systems is heavily subject to the variations in the imaging conditions [110], such as illumination variation.

In Section 2.5.1, we have reviewed approaches that have been proposed to deal with the illumination effects in face recognition. These approaches need either multiple training face image per person or 3D face model database for modeling illumination effects. Zhao and Chellappa [159] tries to use only one image by recovering 3D shape from a single face image. They use symmetric shape-from-shading but the method may suffer from the drawbacks of shape-from-shading, such as the assumption of point lighting sources. Zhang and Samaras [154] propose to recover the 9 spherical harmonic basis images from a single
Figure 8.8: The middle image is the input. The sequence shows a $180^\circ$ rotation of the lighting environment.

Figure 8.9: Interactive lighting editing by modifying the spherical harmonics coefficients of the radiance environment map.
face image. The method in [154] needs a 3D database as in [13] to estimate a statistical model of the spherical harmonic basis images. In this section, we show that our technique on face relighting from a single image can be used for face recognition. The advantage of this approach is that it only require one image for modeling illumination effects. The underline theory of this technique holds under general lighting conditions, such as area light sources. Recently, Qing et al. [114] applied similar relighting method for face recognition using CMU PIE database [124]. Their results are promising.

To normalize the illumination effects for face recognition, we propose to relight all face images into one canonical lighting condition [144]. The lighting condition of the training images can be used as the canonical lighting condition. Then we can use equation 8.11
Figure 8.11: Examples of Yale face database B [52]. From left to right, they are images from group 1 to group 5.

for relighting images into the canonical lighting condition. After that, any face recognition algorithms such as Eigenfaces (PCA) [136], Fisherfaces (LDA) [6], can be used on the pre-processed face images for face recognition.

In our preliminary experiments, we first test our approach using the public available face database: “Yale Face Database B” [52]. The database contains 5760 single light source images of 10 subjects each seen under 576 viewing conditions (9 poses × 64 illumination conditions). In our current experiment, we only consider illumination variations so that we choose to perform face recognition for the 640 frontal pose images. We choose the simplest image correlation as the similarity measure between two images, and nearest neighbor as the classifier. For the 10 subjects in the database, we take only one frontal image per person as the training image. The remaining 630 images are used as testing image.

The images are divided into five subsets according to the angles of the light source direction from the camera optical axis. For the five subset, the values of angle are: (1) less than 12 degrees; (2) between 12 and 25 degrees; (3) between 25 and 50 degrees; (4) between 50 and 77 degrees; and (5) larger than 77 degrees. The face images in the Yale database contain challenging examples for relighting. For example, there are many images with strong cast shadows; or with saturated or extremely low intensity pixel values. Figure 8.11 shows one sample image per group of the Yale face database B. The relighted sample face images are shown in Figure 8.12. It can be observed that relighted images in extreme lighting conditions could be noisy due to outliers. We can detect those outliers using the method described in
Figure 8.12: Relighted examples of Yale face database B [52]. From left to right, they are images from group 1 to group 5.

section 8.1.2. The constrained texture synthesis described in section 8.1.2 can improve the visual appearance of these outliers, however, the synthesized texture may not provide useful information in recognition. Therefore, we discard these outliers in recognition experiments.

We compare the recognition results using original images and relighted images. The experimental results are shown in Figure 8.13. We can see that the recognition error rates are reduced after face relighting in all the cases. When the lighting angles become larger, the illumination effects in original test images are more different from the training images. Therefore, the recognition error rates become larger. In these scenarios, our relighting technique significantly reduces the error rates, even in very extreme conditions (e.g. lighting angles larger than 77 degrees).

In summary, our face relighting technique provides an efficient and effective way to normalize the illumination effects for face recognition. Compared to other approaches, this method has the following advantages: (1) it does not assume simple point lighting source model; instead it works under natural illumination in the real world; (2) it only needs one training image per person to model illumination effects without the need of multiple training images or 3D face database. In the future, we plan to further improve the results under extreme lighting conditions. To deal with cast shadows, techniques using more basis images such as that in [37] will be useful. The difference between the 3D generic face geometry and the actual one may introduce artifacts. We are planning on estimating a personalized geometric model from the input image by using techniques such as those reported in [14, 155].
8.3 Synthesize Appearance Details of Facial Motion

Liu et al. [86] used ratio image technique, called Expression Ratio Image (ERI), to map one person’s face expression details to other people’s faces. More generally, this technique can be used to synthesize facial-motion-related appearance variations for new subject. In this way, the appearance examples for visemes and expressions can be used in face synthesis for new people. In this section, we discuss two issues that need extra consideration when applying the ratio image technique. First we discuss how to map appearance of mouth interior in Section 8.3.1. The mouth interior appearance is important for applications such as lip-reading, but does not satisfy the assumption of ratio image technique. Next, We describe how to create face animations with appearance variations in Section 8.3.2.
8.3.1 Appearance of mouth interior

The appearance of teeth and tongue is important for visual speech perception. However, there is no robust technique to capture their motion so far. Moreover, compared to the skin it is more difficult to measure the surface reflectance property for realistic rendering. Due to these reasons, it is more feasible to capture the motion details of mouth interior using appearance.

To use texture variation to synthesize the mouth interior motion, we simplify the geometry of mouth interior as 3 rows of triangles behind the lips as in [57]. The original tongue and teeth models are still used when we do not change the texture of mouth interior.

The ratio image technique assumes a Lambertian manifold, which does not hold for the interior of mouth. Thus, we choose to directly warp the mouth interior texture based on inner lip contours to new subject. Because ratio image technique is applied to transfer the details outside of the inner lip contours, copying the warped mouth interior texture to the new subject will not create sharp discrepancy along the inner mouth contour. The mouth interior texture can be modulated by a constant to account for different camera exposure.

In a preliminary experiment with 2D face images, we map one person’s speech-related appearance variations in mouth area to a different person. The results are illustrated in Figure 8.14.

8.3.2 Linear alpha-blending of texture

To synthesize face texture variation caused by facial motion, we use linear combinations of face appearance examples. This idea of multi-texture blending has been shown to be useful in facial motion synthesis by Pighin et al. [112] and Reveret et al. [121]. Furthermore, it is well supported by graphics hardware. For a face appearance example set \( \{I_k, k = 1...K\} \),
we represent an arbitrary texture as

\[
\hat{I} = \sum_{k=1}^{K} w_k I_k
\]  

(8.12)

In Chapter 5, we have discussed how to generate face animations using geometric MUs. To augment the synthesis with appearance variations, we propose to decide the texture blending coefficients based on the corresponding geometric MUPs. The intuition is that geometry and appearance are correlated such that partial information about appearance can be inferred from the corresponding geometry. In similar spirit, Zhang et al. [156] have demonstrated the effectiveness of synthesizing facial expression appearance details from given geometry.

Compared to [156], the geometry part of motion is fully derived from our geometric-model-based synthesis. Thus we can design a simpler formulation for the blending coefficients. In this scenario, the problem is to find an appropriate texture given geometric shape: \(s\). Suppose the corresponding geometric MUP is \(\bar{c}(s)\) and the exemplars’ geometric MUPs are \(\{\bar{c}_k, k = 1...K\}\). We define the blending the coefficient as

\[
w(s) = be^{-\lambda_k \|\bar{c}(s) - \bar{c}_k\|^2}
\]  

(8.13)
where $b$ is a constant which normalize the sum of blending coefficients to 1. In practice, we need to avoid the blurring of the blending result. For this purpose, we adjust the value of the constant $\lambda_k$ experimentally such that there are only $N, (N < K)$ nonzero blending coefficients. Other coefficients are small and can be set to zero.

### 8.4 Summary

We have described methods of face synthesis based on the flexible appearance model. In particular, we have discussed two issues: (1) how to synthesize illumination effects in face appearance; and (2) how to synthesize appearance variations in face animations. The main contribution of this chapter is that we show that our flexible appearance model can be used for synthesis in a flexible way. More specifically, It means we can synthesize appearance variations based on the model across different people and illumination environments.
Chapter 9

Application Examples of The Face Processing Framework

In this chapter, we discuss applications of our 3D face processing framework. Two application will be described in more details. The first application is model-based very low bit-rate face video coding. The other application is an integrated human-computer interaction environment. Finally, we conclude this chapter and discuss other potential applications that could benefit from our 3D face processing framework.

9.1 Model-based Very Low Bit-rate Face Video Coding

9.1.1 Introduction

Facial motions convey very important visual cues which are valuable for human-to-human communication. When the volume of video data is overwhelming and stable high-capacity bandwidth is not available, very low bit rate video coding can be a solution for teleconferencing.

To accommodate the needs of transmission of large volume of video data over limited channel, several video coding standards, such as MPEG-1, MPEG-2, H.261, H.263 and H.264 have been proposed. Because these approaches only utilize the spatial-temporal redundancy
statistics of the video waveform signal without a prior knowledge of the semantic content of the video, they are well applicable for general purpose video data compression where the scene in video frame is arbitrary. In the mean time, due to the difficulty to extract redundancy from video, a high coding rate usually also accompanies a high coding latency for certain video quality. This hinders these approaches from applications where real-time video transmission is needed.

For the applications where human face is known as the major focus of attention, model-based coding has been proposed to improve coding efficiency [2]. One example is the MPEG-4 face animation standard [101]. In these approaches, the human face geometry is characterized with a 3D mesh model. The facial motion is parameterized as rotation and translation for rigid motion, and action unit or facial muscle weights for non-rigid facial motions. These parameters together with the video background can be transmitted over channel at very low bit rate, and the video can be reconstructed via synthesis of the facial area based on the transmitted parameters. However, currently there are no completely model-based available coder yet, because it is difficult to extract these facial geometry and motion parameters from video automatically and robustly. Furthermore, the residual of the model-based coding is not transmitted in many approaches. Therefore, the differences between the original video and reconstructed video could be arbitrarily large.

Eisert et al. [42] propose a hybrid coding technique using a model-based 3D facial motion tracking algorithm. In this approach, the model-based coding results and waveform-based coding results are compared and best results are used. In this way, the two coding schemes can complement each other. In this thesis, we propose a model-based face video coder in similar spirit. Nonetheless, the proposed very low bit rate face coding method is efficient and robust because of our 3D face tracking.

We first locate the face in the video frame. Next, the generic facial geometric model is adapted to the face, and facial texture for the model is extracted from the first frame of the video. The facial motion is then tracked and synthesized. The residual error in face area
and video background are then coded with state-of-the-art waveform based coder. Finally
the facial motion parameters, coded residual error and video background are transmitted
at very low bit rate. Experiments show that our method can achieve better PSNR around
facial area than the state-of-the-art waveform-based video coder at about the same low bit
rate. Moreover, our proposed face video coder has better subjective visual effects.

9.1.2 Model-based face video coder

The face video is first sent to a face tracker that extracts face motion parameters. Then a
face synthesizer synthesize a face appearance based on the motion parameters. After the
synthesized face is obtained, the residual error can be calculated by subtracting it from the
original frame. The video frame is divided into foreground residual and background region.
For the background and foreground residual, since we do not assume prior knowledge about it
we can employ the advantage of state-of-the-art waveform-based coder to do the coding. The
advantage of this approach is that the facial motion details not captured by the geometric
motion parameters will not be lost when the video is coded at the low bit rate.

The chosen waveform-based video coder is JVT reference software JM 4.2 obtained
from [58]. It is a joint effort between ISO MPEG and ITU H.26x after the successful H.264
development, and represents the state-of-the-art in low bitrate video coding. The back-
ground will be coded as ordinary video frames by H.264/JVT codec, and the foreground
residuals are coded by Intra_16X16 mode of the H.264/JVT coder. At receiver, the decoder
synthesizes the facial motion according to the received face motion parameters, reconstructs
the foreground and background regions, and recovers the foreground facial area by summing
up the synthesized face and transmitted foreground residuals. Because most of the facial
motion details are captured by the facial motion parameters, the foreground residual tends
to have small amplitude, we can choose to code the video with very lot bit rates without
losing much information of the foreground.

For the face tracker in the system, we used the geometric 3D face tracking system de-
Table 9.1: Performance comparisons between the face video coder and H.264/JVT coder.

<table>
<thead>
<tr>
<th></th>
<th>Face Video Coder</th>
<th>H.264/JVT Coder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit-rate</td>
<td></td>
<td>18-19 Kbps</td>
</tr>
<tr>
<td>PSNR</td>
<td>29.28</td>
<td>27.35</td>
</tr>
<tr>
<td>Coding Time</td>
<td>1.4</td>
<td>5</td>
</tr>
</tbody>
</table>

scribed in Chapter 4. The face synthesizer uses the methods presented in Chapter 5. In the current model-based face video coder, we ignore the face texture variations in tracker and synthesizer. Instead, we let H.264/JVT coder to deal with them. In the future, we plan to apply our flexible appearance model to deal with these texture variations.

### 9.1.3 Results

The face tracker runs on a PC with two Pentium\textsuperscript{4} 2.2GHZ processors, G-Force 4 video card, and 2G memories. With only one processor employed, the tracking system can reach 25 frame per second (fps) in rigid tracking mode and 14 fps in non-rigid tracking mode. We capture and encode videos of 352 × 240 at 30Hz. At the same low bit-rate (18 ∼ 19 kbits/s), we compare this hybrid coding with H.26L JM 4.2 reference software. For a face video sequence with 147 frames, the performance comparisons of the two coders are presented in Table 9.1. It can be observed that our face video coder has higher Peak Signal to Noise Ratio (PSNR) for face area and is more computationally efficient. Figure 9.1 shows three snapshots of a video with 147 frames. One important result is that our face video coder results have much higher visual quality in face area.

### 9.1.4 Summary and future work

As an application of the 3D face processing framework, we present an efficient and robust very low bit rate face coding method via 3D face face tracking. The facial motion parameters can be extracted from the video and transmitted over the channel. Then the facial area residual errors and video background can be coded using waveform-based coder at very low
Figure 9.1: (a) The synthesized face motion. (b) The reconstructed video frame with synthesized face motion. (c) The reconstructed video frame using H.26L codec.
bit rates. Experiments show that our method can achieve better PSNR around facial area than H.264/JVT coder at about the same low bit rate and have better subjective visual quality.

The key issue of model-based coding is the 3D face tracker. We plan to improve the accuracy and robustness of our 3D face tracking algorithm. Because our tracking system works in real-time, we can combine it with real-time waveform-based coder to make the overall system real-time. Then it could be used in a real-time low bit-rate video phone application. Finally, we plan to model the face texture variation so that the residual can be further reduced.

9.2 Integrated Proactive HCI environments

Face tracking and expression recognition techniques help computers monitor users’ states in a human-computer interaction (HCI) environments. On the other hand, the face synthesis technique can be used to create synthetic avatar to interact with users. Here we describe an integrated HCI environment where our face processing techniques are used.

9.2.1 Overview

In the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign, there is an interdisciplinary project called: "Multimodal Human Computer Interaction: Towards a Proactive Computer". The goal of the project is to (1) identify ways to provide the computer with real-time information about the human users cognitive, motivational and emotional state; and (2) explore ways of making the computer take actions proactively based on the states information. In this way, the computer will no long only take commands from the user passively. Instead, it will initiate actions based on user’s states and thus make the human-computer interaction more effectively.

The current application area for the project is science education, in which children learn
about the properties of gears via LEGO™ games. By learning how to put gears together, the children can learn scientific principles about ratio, forces and etc. The educational philosophy is such that learning through exploration is valued. That is, rather than directing the child in carrying out learning tasks, the goal is to (1) encourage exploration of gears and gear trains in which principles of gears are learned; and (2) ask questions that encourage thought and insight. The final goal is a HCI learning environment that engages children in activities with the tasks, tracks their path and progress in the problem space, and initiates communication (e.g. asking questions, making comments, showing examples, etc.). These activities will then encourage exploration, maintain and develop interest and move the child toward new understandings.

In this learning environment, the information exchanged among different participants is multimodal. The computer is used to analyze the multimodel input, including facial expression analysis, prosody analysis, context-based word spotting, visual tracking of the task states and eye-gaze tracking. Based on the multimodal input analysis, the computer can estimate the user states. Then the computer can map the user states to actions to be taken. The mapping can be designed by domain expert or learned from extensive examples. Finally, the computer executes the action to guide the children in the next step of exploration. One type of useful output from the computer is synthetic face animation. The synthetic face avatar can be used to represent the computer to interact with children and help engage user in the learning session.

9.2.2 Current status

Initially, human tutoring sessions with children were videotaped and extensively analyzed. As a result a set of annotated tapes and transcripts are produced. These collected data may serve as input to computer learning algorithms that are attempting to establish a mapping from user’s states information to tutor actions.

Currently, the integrative human-computer interaction control system is being developed.
Figure 9.2: The setting for the Wizard-of-Oz experiments.

The results of the multimodal input and output are displayed for the sessions operating in a Wizard-of-Oz environment. The environment is illustrated in Figure 9.2. In this environment, the child and the tutor are in separate rooms. The student is not aware of the presence of a human tutor. Instead, the child supposes he or she is interacting with the computer via the face avatar. The avatar outputs synthesized speech, shows emotional expression and directs the student’s gaze to selected regions. Meanwhile their behaviors are recorded for further study. On the other hand the instructor can see the multimodal signal analysis results and initiate appropriate actions. The interfaces for the child and the instructor are shown in Figure 9.3. This system is currently being used in educational/psychological research. Experimenting are being carried out to classify multimodal input into user state categories. The classification is beginning to replace manual analysis of some video data.

As shown in Figure 9.3, 3D face tracing and expression recognition techniques are now used as part of the cues to estimate the states information. Other useful cues include speech prosody, states of the task, and etc. On the other hand, the synthetic face animation is used as the avatar to interact with the child. Preliminary results suggest that the synthetic avatar
helps the children more patient in the learning session.

9.2.3 Future work

In this proactive HCI environment, the facial motions of the children are very challenging to analyze compared to database collected in controlled laboratory conditions. One reason is that real facial motions tend to be very fast at certain occasions. These fast motions can cause the current face tracker lose track. In the future, we need to future improve the speed of the face tracker so that it can capture fast facial motions in real-life conditions. Another reason is that real facial expressions observed in real-life environment are more subtle. Therefore, the performance of the face expression recognition can be affected. We plan to carry out extensive studies of facial expression classification using face video data with spontaneous expressions. In this process, we could be able to improve the expression classifier.

Another future direction of improvements is to make the synthetic face avatar more active so that it can better engage the children in the exploration. For example, we plan to synthesize head movements and facial expressions in the context of task states and user
states. In this way, the avatar can be more lifelike and responsive to users’ actions in the interaction. One possible direction is explore the correlation of speech and head movements as the work by Graf et al. [55].

9.3 Summary

In this chapter, we have described two applications of our 3D face processing framework. One application is to use face processing techniques to encode face visual cues in communication. Experiment shows that this approach achieves higher PSNR and better visual quality in very low bit-rate conditions. The other application is an integrated HCI environment for computer-aided education. In this environment, face analysis techniques is used to understand the users’ state and synthetic face is used as interface to help engage users.

More generally, our face processing framework can be applied in many applications related to human faces. Besides the two applications described in this chapter, other examples of potential applications include: (1) intelligent video surveillance; and (2) diagnosis and rehabilitation tools for face-related medical problems. In security-related video surveillance, human faces provide valuable visual cues to identify people and understand human activities. In face-related medical problems, such as disorders of facial muscles and associated nervous system, facial visual cues are important input for diagnosis of the problems. Our face processing framework provides a possibility for automating the diagnosis. On the other hand, one of the rehabilitation techniques is presenting appropriate audio-visual stimuli (e.g. videos of normal facial expressions and talking faces) to patients. The face synthesis techniques can help to generate and manipulate these stimuli more easily.
Chapter 10

Conclusion and Future Work

In this thesis, we have presented a unified 3D face processing framework. Various aspects of face processing research have been discussed in the context of multi-modal human computer interaction and intelligent video analysis. In this chapter, we summarize the contributions of this thesis and outline the future research directions.

10.1 Conclusion

In this thesis, we have presented a unified 3D face processing framework. In Chapter 5.1, we describe tools for building 3D geometric models of neutral faces. Using these tools, we can create personalized 3D face models for 3D face tracking and animation. Then we discuss the geometric facial motion models in the framework in Chapter 3. These motion models are derived from motion capture data of real face motions. Thus they can capture the characteristics of real face motions. After that, we present our approaches for 3D non-rigid face tracking and animation in Chapter 4 and Chapter 5, respectively. We demonstrate the efficacy of tracking and animation, using experimental results in very low bit-rate face video coding, speech-drive face animation, and etc.

In Chapter 6, we present flexible appearance model to deal with appearance details which are lost in the geometric facial motion models. In out experiments, these details are shown to be important for computer analysis of subtle facial expressions [146], and human perception
of synthesized face animations [148]. Compared to most existing appearance models for face motions, our flexible appearance model is less illumination dependent, and less person-dependent. It also requires less data for estimating the parameters of the flexible appearance model. Therefore, our appearance model can be more flexibly used in various environments.

10.2 Future Work

To improve the 3D face processing framework, future research should be conducted in the following several directions.

10.2.1 Improve geometric face processing

The geometric face processing can be improved by utilizing more statistics of increasingly available 3D face data. One direction is to estimate better 3D face geometry from a single face image following the approach of Blanz and Vetter [14]. The improved 3D face estimation can provide a better 3D-model-fitting for the first video frame in the non-rigid face tracking. The more accurate 3D face model will also improve the performance of face relighting techniques described in Section 8.1 of Chapter 8.

Another direction is to collect motion capture data of more subjects so that the model derived from data can better describe the variations across different people. As a result, facial motion analysis can be used for a larger variety of people. For synthesis, such database would enable the study the personalized styles in visual speech or facial expression synthesis.

10.2.2 Closer correlation between geometry and appearance

In our current 3D face processing framework, we first use geometric model to process the geometric-level motion. Next, the remaining appearance details are handled by the flexible appearance model. In this procedure, we assume that the geometric processing part gives reasonable results so that face textures are correctly aligned with the geometry. However, this
assumption is not always true. For example, in 3D face motion analysis, if the geometric tracking gets lost, the extracted face texture would be wrong and the appearance-based analysis would then fail. Therefore, we proposed an appearance-based enhancement of the geometric tracker in Chapter 7, Section 7.3. This enhanced geometric tracker improves performance for low resolution face video.

In the future, we plan to generalize the appearance constraint to deal with appearance variations under large out-of-plane rotation and non-rigid motions. Recently, Vacchetti et al. [137] use the face appearances in a few key frames and the preceding frame as constraints for estimating the 3D rigid geometric face motions. These constraints help to reduce drifting and jittering, even when there are large out-of-plane rotations and partial occlusions. Following the spirit of [137], we plan to use multiple key-frames to derive appropriate warping templates for different range of views. The appearance-based enhancement of geometric tracking can also be extended to handle non-rigid facial motions. For each Motion Unit (MU) in our framework, one corresponding warping template need to be included in equation 7.15.

### 10.2.3 Human perception evaluation of synthesis

Human perception evaluation of face synthesis should be done in the context of specific applications such as lip reading. Hypotheses about visual factors need be created and tested in the evaluation. These hypotheses can then be used to improve face synthesis so that the synthetic animation can be more effective for applications.

### Previous work

To evaluate the quality of synthetic face animation, one approach is to compare the synthetic face with the original face in terms of reconstructed error. This approach is used by (1) low bit-rate coding oriented face animation such as MPEG-4 FAPs [42, 135]; and (2) machine learning-based data-driven face animation such as [15, 57].

However, the synthetic face motion can be different from the real motion but still looks
natural. Consequently, human subject evaluations are important. Human evaluation with a small set of subjects were used in [15, 46]. On the other hand, in many scenarios mimicking real human face motion is not the only goal for synthetic face animation. For example, non photo-realistic styles and abstraction can be created to convey certain information in face animation [19, 25]. Another type of face animation applications is emerging from the interdisciplinary Human Computer Interaction research areas, such as using synthetic face animation for language training [31] and psychological studies [91]. For these scenarios, the naturalness of face animation can be compromised, while certain motions can be exaggerated for application purpose (e.g. mouth motion for lip-reading applications). Therefore, human subject perception experiments are usually carried out within the application context, to evaluate and guide further improvement of the face animation. Massro et al. [91] have developed a face modeling and animation system, called “Baldi”. “Baldi” has been used to generate synthetic stimuli for bimodal speech perception research. Moreover, it has been applied in language training for school children [31].

For multi-modal human speech perception, a 3-process model was proposed by Massro et al. [91]. The three processes are: (1) “evaluation”, which transforms the sources of information to features; (2) “integration”, where multiple features are integrated both between modalities and within a modality; and (3) “decision”, which makes perception decision based on integration results. For integration, a Fuzzy Logical Model of Perception (FLMP) was proposed. It is mathematically equivalent to Bayes’ theorem, which is widely used in pattern recognition. FLMP has been shown to be effective in integrating multiple cues for multi-modal speech perception.

“Baldi” and the FLMP model were used to test hypotheses on bimodal speech perception. Some important results include:

- The auditory and visual channels could be asynchronous to certain degrees without affecting the performance of speech perception. In Massro’s work, 100 ~ 200 ms delays of one channel did not interfere with the performance. Some words/phonemes
allow longer delays than the others. This result agrees with other researchers’ findings. McGrath and Summerfield [93] reported that a delay up to 80 ms did not disrupt performance. Pandey, Kunov and Abel [104] found an upper limit of 120 ms. A result of 200 ms as the upper limit was reported by Campbell and Dodd [21].

The speech reading performances are relatively robust under various conditions, including different peripheral views, image resolution, viewing angles and distances. First, experiments show that speech reading performances do not degrade even when the perceiver is not looking directly at the mouth. Second, as the resolution of face image decreases, phonemes with large scale motion such as /W/ can still be reliably recognized. However, phoneme featuring more detailed motion such as the interdental motion of /TH/, can get confused with other phonemes. Next, it is also found that speech reading performances degrade little when viewing non frontal faces. Profile views even improve performances for certain phonemes. But on the average, frontal views are still the best. Finally, the performances remain fairly good, when the distance from the synthetic talker to the perceiver is within 4 meters.

Other researchers also reported that speechreading is robust when dynamics of visual speech is changed. IJsselijk [69] found that performance speechreading degraded little even when the temporal sequence is slowed down four times. Williams and et al. [149] reported that the recognition rates of visemes degraded less than 5%, even when the sampling rate of visual speech was only $5 \sim 10$ frame per second.

It is also reported that adding extra visual features could improve speech reading after a few rounds of training. For example, in their experiments color bars were used to show whether the current sound is “nasal”, “voicing” or “friction”. The bars were displayed besides the talking face. The results showed that speech reading correct rates improved significantly after presenting the material five times to the perceivers. It implies that perceivers could adapt to the extra visual features in a fairly short
period of time.

Although “Baldi” has been shown to be a useful animation tools for research and applications in speechreading, it has the following limitations:

- Only macrostructure level geometric motions are modeled in current system. Therefore, it loses important visual cues such as the shading changes of the lips and area within the mouth. These visual cues are important to perceive lip gestures such as rounding and retraction, and relative positions of articulators such as teeth and tongue. As a result, subtle phonemes (e.g. /R/) are more easily confused. To deal with this problem, one possible way is to use significantly more detailed geometry and advanced rendering to reproduce these subtle appearance changes. Modeling detailed geometric motion of mouth can be very expensive, because it involves complex wrinkles, surface discontinuities and non-rigid self collisions. Furthermore, the mouth interior is difficult to measure. For real-time rendering, it is also expensive to model the diverse material properties of lip, teeth and tongue and perform ray-tracing. Therefore, modeling these visual cues as texture variation is more feasible for speech perception applications.

- “Baldi” is a complex animation system with a great number of parameters. For basic tongue movement alone, there are more then 30 parameters. If a user wants to create customized face shapes for applications, it would be difficult and time-consuming unless the user has in-depth knowledge of “Baldi”. Therefore, it is desired that systematic approaches be developed to simplify the use of the animation system. For example, one possible systematic approach could be to customize the face shapes from real data using machine learning techniques.

- The development of “Baldi” is not based on an open framework. That is, the components, such as spatial deformation model and temporal deformation model, are created and tuned exclusively for “Baldi”. Therefore, it is difficult for other researchers to incorporate components of “Baldi” into other animation systems. It would be highly
desirable that the development of “Baldi” is formulated as more general methodology such that other researchers could re-use and refine its components in the future.

Our ongoing and future work

Compared to “Baldi”, one of the goals of our research is to provide a general, unified framework to guide the development of face motion modeling, analysis and synthesis. It could result in compact and efficient animation tools, which can be used by users with various backgrounds (e.g. psychologists) to create animation suitable for their applications. On the other hand, we make use of feedback from those applications to devise general principles to guide the refinement of the synthesis.

The current target application for evaluating our face synthesis is lip-reading. In this application, face animations synchronized with speech are generated and presented to hearing-impaired people. If the face animations are lip-readable, it will help the hearing-impaired people better understand the speech. We plan to conduct human perception studies to identify hypotheses about visual factors that are important to lip-reading. Then these hypotheses can be used to guide the improvement of the face synthesis.

In our preliminary experiments, we first create animations for isolated digits. Then these animations are presented to human subjects. The current subjects include one PhD student and one faculty member who have lip-reading experiences. In the first test, we test the lip-readability of face animation produced using geometric motion model only. We find the following factors limit the lip-readability: (1) the animation lacks wrinkles and shading changes in lip area so that it is difficult for the perception of lip rounding and protrusion when their durations are small; (2) the crafted tongue and teeth motions do not provide enough visual cues to recognize interdental phoneme like /TH/ in “three”. Besides, certain un-natural synthesis results of mouth interior are distracting for lip-reading. In the second test, we augment the animation by using appearance model to synthesize texture variation. The results show that the perception of subtle lip rounding, protrusion and stretching is
considerably improved because of the added appearance variations. The appearance model also handles the complex details inside the mouth. As a result, the recognition of interdental phonemes such as /TH/ is improved. However, subtle dynamic appearance changes, such as the fast tongue tip movement in “nine”, can not be synthesized by the appearance model. The reason is that only one image is currently used to model a phoneme, thus the dynamic details are lost.

In the future, we plan to conduct human perception test using more subjects so that the experiment results can be more statistically rigorous. We need to identify people whose visual speeches are highly lip-readable, and derive better motion models from their data (both geometrical and appearance model). We also plan to investigate methods using increased appearance samples to model the dynamic appearance changes.
Appendix

The multiplication of two spherical harmonic basis satisfies the following relation according to [30]:

\[ Y_{l_1m_1}(\vec{n})Y_{l_2m_2}(\vec{n}) = \sum_{l=|l_1-l_2|}^{l_1+l_2} \sum_{m=-l}^{l} \{ C(l_1, l_2 : 0, 0|l, 0) \cdot C(l_1, l_2 : m_1, m_2|l, m)Y_{lm}(\vec{n}) \} \]

(1)

where \( C(l_1, l_2 : m_1, m_2|l, m) \) is the Clebsch-Gordan coefficients. The coefficient of \( Y_{lm}(\vec{n}) \) in the righthand side is non-zero if and only if \( m = m_1 + m_2, l \) range from \( |l_1 - l_2| \) to \( l_1 + l_2 \) and \( l_1 + l_2 - l \) is even.

We then look at the second term of the righthand side in equation 6.9. Suppose \( Y_{l_1m_1}(\vec{n}) \) comes from \( \Psi(\vec{n}) \), and \( Y_{l_2m_2}(\vec{n}) \) comes from the factor corresponding to lighting. For any \( Y_{l_2m_2}(\vec{n}) \), we have \( 0 \leq l_2 \leq 2 \). If \( l_1 > 4 \) for any \( Y_{l_1m_1}(\vec{n}) \), we have \( l \geq |l_1 - l_2| > 2 \). That proves the claim that if \( \Psi(\vec{n}) \) does not have the first four order (\( l = 1, 2, 3, 4 \)) components, the second term of the righthand side in equation 6.9 contains components with orders equal to or higher than 3.
References


Vita

Zhen Wen was born on January 18, 1975, in Kunming, P. R. China. He received the Bachelor of Engineering degree in Computer Science from Tsinghua University, Beijing, China, in 1998. In August 1998, Mr. Wen was admitted to the Department of Computer Science of the University of Illinois at Urbana-Champaign, Urbana, Illinois, US. Since 1999, he has worked as a research assistant in the Image Formation and Processing research group at the Beckman Institute for Advanced Science and Technology. He received his Master of Science degree in Computer Science in 2000. His research interest includes image and video processing, human computer interaction, computer graphics, computer vision and pattern recognition, machine learning, and multimedia systems. Mr. Wen’s research focuses on pattern recognition, computer vision and computer graphics with their applications in Human Computer Interaction. His dissertation works on 3D face motion modeling, analysis and synthesis enhanced with flexible appearance models.