

Facial Landmark Detection and Sketch Recognition

BY

SHAUN CANAVAN

BS, Youngstown State University, 2006  
MCIS, Youngstown State University, 2008

DISSERTATION

Submitted in partial fulfillment of the requirements for  
the degree of Doctor of Philosophy in Computer Science  
in the Graduate School of  
Binghamton University  
State University of New York  
2015

© Copyright by Shaun Joseph Canavan 2015

All Rights Reserved

Accepted in partial fulfillment of the requirements for  
the degree of Doctor of Philosophy  
in the Graduate school of  
Binghamton University  
State University of New York  
2015

April 29, 2015

Dr. Lijun Yin, Faculty Advisor  
Department of Computer Science, Binghamton University

Dr. Les Lander, Member  
Department of Computer Science, Binghamton University

Dr. David Lu, Member  
Department of Computer Science, Binghamton University

Dr. Scott Craver, Outside Examiner  
Department of Electrical and Computer Engineering, Binghamton University

## **Abstract**

***Biometrics*** refers to technologies that measure and analyze human body characteristics, such as voice patterns, fingerprints, gait pattern, eye retinas and irises, facial patterns, DNA, and hand measurements, for authentication purposes. It has a wide range of applications and impacts on our human society, ranging from security, communication, health-care, law-enforcement, entertainment, education, and so on. In the past decades, biometrics research has been focused on automatic recognition of human body traits including fingerprint, iris, ear, periocular, face, palm, handwriting, gait, voice, and other modalities, as well as multi-modal biometrics and new biometrics based on novel sensing technologies. While each of these topics has been studied, facial characteristics attracted the most intensive investigations due to its non-intrusive nature with the most common measure in communication and the ease of data acquisition for individual identification and authentication.

In terms of face biometrics, multiple sensing technologies have been developed for capturing facial data in multiple dimensional spaces, such as 2D, 3D, and 4D in recent years. However, one type of data is not obtainable with existing sensors, whereas the visual information can be obtained only by witness' description and rendered by forensic artists, which is call Face Sketch. Sketch recognition is extremely important for law enforcement and security. Conventional face recognition systems can be used in a real-time setting where cameras are able to capture a potential suspect, while sketch

recognition has to be used in a scenario where a witness has given a description of a suspect and the only means of identification is through the match of the sketch and a face database. These technologies have had great advances to the state of the art in recent years; however, there are still many unsolved problems that plague this field. Some of those problems include occlusion from things like glasses, hair, and makeup; changes in pose; ambiguity in shapes; uncertainty in description and representation, and variations in lighting and aging. An important step for tackling those problems is to investigate robust methods to precisely represent, detect, and track facial features, which is a main focus of this dissertation, the other focus being the classification of faces and sketches.

We first propose a novel method for 2D feature detection for determining eye directions. This method utilizes the detected 2D facial features to construct a 3D model of the eyeball and iris. A natural extension to 2D feature tracking is to extend those ideas to 3D and 4D data. To this extent, we also present two novel algorithms for detecting and tracking 3D/4D features. The first method that is presented relies on the explicit shape of 3D data. A so-called *Action-Based Statistical Shape Model (ASSM)* of the 3D shape is created, by sampling features on the testing data, to create a smooth deformation of the training data. These deformations allow us to fit our model to unseen input data to robustly detect and track facial features. The second method for detecting 3D/4D features is to make use of the explicit geometric shape of the face, but it also enforces a local constraint by sampling patches of data around each of the training landmarks. A novel *Shape Index-based Statistical Shape Model (SI-SSM)* is proposed. Each of these methods has shown improvement over state of the art methods. This dissertation also shows an

application of the detected 3D/4D landmarks by using a new *Dynamic Curvature* descriptor for 3D facial activity analysis.

Much of sketch recognition research is done using 2D data. In this dissertation, we propose an innovative approach by moving from the 2D domain to the 3D domain for such a task, which is the first of this kind in the biometrics research community. We present a novel method to construct 3D sketch from 2D data, significantly increasing the realism of sketch representation. Experimental results show that 3D sketch is advantageous in solving the problems of sketch recognition. In addition, we investigate face identification under strong shadow, which is a very challenging problem. We present an analysis of a fusion-based face recognition method. This method achieves approximately double the recognition rate as compared to the conventional methods which are based on a single image only.

In general, this dissertation addresses two important issues of face biometrics: landmark detection and sketch recognition in multi-dimensional spaces. The presented new methods with the experimental validation show the advancement to the state of the art in terms of both theoretical significance and practical applications.

In dedication to my wife, Amy, to my daughter, Scarlett,  
to my mother, Susan, and the memory of my father, Joseph.

## **Acknowledgements**

I would like to thank my advisor, Dr. Lijun Yin. He has provided me with a great deal of guidance, teaching, and support throughout this work. His invaluable advice has taught me how to approach a new problem and what questions to ask. I have learned an immeasurable amount about the craft of research and presenting those results in a publishable format. His intelligence and drive have been great inspirations to me throughout my incredible journey of pursuing a Ph.D.

I would also like to thank my first graduate advisor, Dr. Yong Zhang. He first showed me what was possible with research and helped solidify my decision to pursue a Ph.D. I am grateful for the guidance and advice he gave throughout our short, but fruitful time together.

To the members of the Graphics and Image Computing Lab that I have collaborated with, you have been an amazing source of inspiration and knowledge throughout my journey. I would like to thank Michael Reale, Peng Liu, Kaoning Hu, and especially Xing Zhang for our many in-depth, and always enjoyable, conversations on a wide range of topics.

Lastly, I would like to thank my wife, Amy, my daughter, Scarlett, and my mother, Susan. Without your support and guidance this would have been a much different, and ultimately less enjoyable, experience. Without you this would mean nothing.



## Table of Contents

|   |      |
|---|------|
| List of Tables .....  | xiii |
| List of Figures .....   | xiv  |
| List of Abbreviations .....   | xvii |
| 1. Introduction .....   | 1    |
| 1.1 Problems and Open Questions in the Biometrics Community .....         | 2    |
| 1.2 Background and Motivation .....                                       | 3    |
| 1.3 Objective and Contributions .....                                     | 6    |
| 1.3.1 2D tracking for eye viewing direction .....                         | 7    |
| 1.3.2 Image fusion for face recognition under shadow .....                | 8    |
| 1.3.3 3D face sketch recognition .....                                    | 8    |
| 1.3.4 Dynamic curvature description for 3D facial activity analysis ..... | 9    |
| 1.3.5 3D landmark detection tracking .....                                | 9    |
| 2. Dynamic Face Appearance Modeling and Sight Direction Estimation .....  | 11   |
| 2.1 Introduction .....  | 11   |
| 2.2 Tracking with Person-Dependent AAM .....                              | 13   |
| 2.3 Scale-Space Topographic 3D Modeling .....                             | 14   |
| 2.3.1 Dynamic 3D appearance for modeling .....                            | 14   |
| 2.3.2 3D Iris modeling and sight direction estimation .....               | 17   |
| 2.4 Experimental Results .....  | 18   |
| 2.4.1 Evaluations .....   | 19   |

|  |    |
|--|----|
| 2.5 Discussion .....   | 21 |
| 3. Fusion Based Face Classification Under Strong Shadows ..... | 22 |
| 3. 1 Introduction .....  | 22 |
| 3.2 Related Works .....  | 23 |
| 3.3 Multi-Frame Fusion Based Method .....                      | 24 |
| 3.3.1 Video Dataset .....                                      | 24 |
| 3.3.2 Frame Selection .....                                    | 25 |
| 3.3.3 Training, Gallery, and Probe sets .....                  | 25 |
| 3.3.4 Fusion Schemes .....                                     | 27 |
| 3.3.5 Measuring Inter-frame Variation .....                    | 27 |
| 3.4 Experimental Results and Discussions .....                 | 29 |
| 3.4.1 Improvement by Multi-frame Fusion .....                  | 29 |
| 3.4.2 Inter-frame Variation .....                              | 30 |
| 3.5 Discussion .....   | 32 |
| 4. 3D Face Sketch Modeling and Recognition .....               | 33 |
| 4.1 Introduction .....   | 33 |
| 4.2 Source Data .....  | 34 |
| 4.3 3D Sketches Creation from 2D Sketches .....                | 37 |
| 4.3.1 3D sketch reconstructions .....                          | 37 |
| 4.3.2 3D sketch accuracy evaluation .....                      | 39 |
| 4.3.2.1 Comparison: 3D HD sketches vs. 3D scans .....          | 39 |
| 4.3.2.2 Comparison: 3D MD sketches vs. 3D scans .....          | 40 |
| 4.3.2.3 Comparison: 3D HD vs. MD .....                         | 41 |

|   |    |
|---|----|
| 4.4 3D Sketch Face Recognition .....                            | 42 |
| 4.4.1 Component region segmentation .....                       | 42 |
| 4.4.1.1 Edge Vertices (EV) determination .....                  | 43 |
| 4.4.1.2 Component regions determination .....                   | 44 |
| 4.4.2 3D Component Feature Representation .....                 | 45 |
| 4.4.3 Spatial HMM Model Classification .....                    | 46 |
| 4.5 Experiments of Face Recognition .....                       | 46 |
| 4.5.1 4DFE: 3D sketch (training) vs. 3D sketch (testing) .....  | 46 |
| 4.5.2 4DFE: 3D scans (training) vs. 3D sketches (testing) ..... | 47 |
| 4.5.3 YSU: 3D sketch (training) vs. 3D sketch (testing) .....   | 47 |
| 4.6 Discussion.....   | 49 |
| 5. Facial Activity Analysis in 3D/4D Space .....                | 50 |
| 5.1 Introduction .....  | 50 |
| 5.2 3D Shape Tracking Model .....                               | 52 |
| 5.3 Dynamic Curvature Based Approach .....                      | 53 |
| 5.3.1 Shape description and quantization .....                  | 54 |
| 5.3.2 Dynamic Curvature Based Descriptor .....                  | 55 |
| 5.3.3 Classification .....                                      | 57 |
| 5.4 Experiments and Evaluation .....                            | 58 |
| 5.4.1 Database .....  | 58 |
| 5.4.2 Facial Activity Classification .....                      | 58 |
| 5.4.3 Comparison .....  | 60 |
| 5.5 Discussion .....  | 61 |

|   |     |
|---|-----|
| 6. 3D/4D Feature Detection Using Action-based Statistical Shape Models .....      | 62  |
| 6.1 Introduction .....  | 62  |
| 6.2 3D Action-Based Statistical Shape Model .....                                 | 65  |
| 6.3 Fitting 3D and 4D Range Data .....  | 68  |
| 6.3.1 Fitting 3D Range Data Using an ASSM .....                                   | 68  |
| 6.3.2 Fitting 4D Range Data Using an ASSM .....                                   | 75  |
| 6.4 Experiments and Evaluation .....  | 79  |
| 6.4.1 Databases .....   | 79  |
| 6.4.2 Database Error Statistics .....   | 80  |
| 6.4.3 Subject and Expression Verification .....                                   | 85  |
| 6.4.3.1 3D Component Feature Representation .....                                 | 85  |
| 6.4.3.2 Spatial HMM Model Classification .....                                    | 86  |
| 6.4.3.3 Temporal HMM Model Classification .....                                   | 87  |
| 6.4.3.4 Subject Verification and Face Expression Classification ..                | 88  |
| 6.4.4 Expression Segmentation (Action vs. Non-Action) .....                       | 89  |
| 6.4.5 Pose Estimation .....   | 90  |
| 6.4.6 Gesture Recognition .....   | 92  |
| 6.5 Discussion .....  | 93  |
| 7. 3D/4D Feature Detection Using Shape Index-based Statistical Shape Models ..... | 95  |
| 7.1 Introduction .....  | 95  |
| 7.2 Shape Index-based Statistical Shape Model (SI-SSM) .....                      | 99  |
| 7.2.1 Global Face Shape .....   | 99  |
| 7.2.2 Local face Shape .....  | 101 |

|  |     |
|--|-----|
| 7.2.3 Combined Global and Local Feature Model .....  | 102 |
| 7.3 3D/4D Landmark Detection and Tracking .....  | 103 |
| 7.4 Experiments and Evaluation .....   | 107 |
| 7.4.1 Databases .....  | 107 |
| 7.4.2 Feature Detection and Tracking on Five Databases .....                                     | 109 |
| 7.4.3 Performance Evaluation .....   | 111 |
| 7.4.3.1 Spontaneous Expression Segments .....  | 111 |
| 7.4.3.2 Rotation Sequences .....   | 113 |
| 7.4.3.3 Low Quality Sequences .....  | 115 |
| 7.4.4 Comparison to the State-of-the-Art .....   | 118 |
| 7.5 Applications .....   | 120 |
| 7.5.1 Posed and Spontaneous Facial Expression Classification .....                               | 120 |
| 7.5.1.1 Approach .....   | 120 |
| 7.5.1.1.1 3D Component Feature Representation .....  | 120 |
| 7.5.1.1.2 Component based Spatial-Temporal<br>HMM Model .....                                    | 120 |
| 7.5.1.2 Experiment results on face expression classification using<br>spatial-temporal HMM ..... | 121 |
| 7.5.2 Pose Estimation .....  | 122 |
| 7.6 Discussion.....  | 122 |
| 8. Conclusion .....  | 123 |
| 8.1 Findings .....   | 123 |
| 8.2 Applications .....   | 124 |

|                                      |     |
|--------------------------------------|-----|
| 8.3 Limitations .....                | 124 |
| 8.4 Discussion and Future Work ..... | 126 |
| References .....                     | 130 |

## List of Tables

|  |     |
|--|-----|
| Table 1. Training gallery and probe sets .....   | 26  |
| Table 2. Exhaustive fusion tests .....   | 27  |
| Table 3. Statistics of rank-1 fusion tests .....   | 28  |
| Table 4. Recognition rates for neutral/non-neutral expressions .....   | 59  |
| Table 5. Recognition rates for six universal expressions .....   | 59  |
| Table 6. Average separation rates of low/high intensities .....  | 60  |
| Table 7. Confusion matrix of individual expressions for intensity<br>(low/high) separation .....   | 60  |
| Table 8. Recognition rate from low intensity (LI) expressions and high intensity (HI)<br>expressions using different approaches respectively ..... | 61  |
| Table 9. ASSM fitting algorithm .....  | 71  |
| Table 10. Summary of databases .....   | 80  |
| Table 11. Average error in point spacings for different databases and resolution in<br>number of vertices per model.....                           | 81  |
| Table 12. Object and action type for hand and arm ASSM .....   | 93  |
| Table 13. SI-SSM fitting algorithm .....   | 106 |
| Table 14. Summary of 3D/4D databases .....   | 108 |
| Table 15. Error rates for all five databases .....   | 110 |
| Table 16. Comparisons of SI-SSM, TDSM, Sun et al., and 2D CLM mapped to 3D ...   | 119 |

## List of Figures

|   |    |
|---|----|
| Figure 1. System diagram of eye sight estimation .....                          | 12 |
| Figure 2. Keys points on face for eye sight estimation .....                    | 14 |
| Figure 3. Eye-ball sphere illustration .....                                    | 18 |
| Figure 4. High-resolution face video example .....                              | 20 |
| Figure 5. Low-resolution videos with eye sight direction estimations .....      | 21 |
| Figure 6. Sample frames of different views with different illuminations .....   | 24 |
| Figure 7. Large difference between gallery and probe sets .....                 | 26 |
| Figure 8. Relationship between rank-1 rate and number of frames in fusion ..... | 30 |
| Figure 9. CMC curves of fusion test series .....                                | 30 |
| Figure 10. Inter-frame variation and the FIR relationship .....                 | 31 |
| Figure 11. Examples of 3D scans from 4DFE .....                                 | 35 |
| Figure 12. Examples of 2D sketches and 3D sketch models (4DFE) .....            | 36 |
| Figure 13. Examples of 2D sketches and 3D sketch models (YSU) .....             | 37 |
| Figure 14. Illustration of 459 points on a sample face .....                    | 37 |
| Figure 15. Error Statistics of 84 testing vertices (HD) .....                   | 40 |
| Figure 16. Error Statistics of 84 testing vertices (MD) .....                   | 41 |
| Figure 17. Error Statistics of 84 testing vertices (HD-MD) .....                | 42 |
| Figure 18. Component region samples .....                                       | 44 |
| Figure 19. ROC curve of 3D sketch face recognition .....                        | 48 |
| Figure 20. Example of tracked 83 features on surprise expression .....          | 53 |



|  |    |
|--|----|
| Figure 21. Shape quantization to nine values .....                                   | 55 |
| Figure 22. Illustration of dynamic curvature descriptor on eight local regions ..... | 57 |
| Figure 23. Example k-frame ASSM .....  | 68 |
| Figure 24. Sample frames from ASSM fitting process .....                             | 72 |
| Figure 25. Best, worst, and ground truth for ASSM fitting process .....              | 72 |
| Figure 26. ASSM fit on models with roll, pitch, and yaw .....                        | 73 |
| Figure 27. Kinect data fit with ASSM .....   | 73 |
| Figure 28. ASSM fit frames with occlusion .....                                      | 74 |
| Figure 29. Eurecom data fit with ASSM .....  | 74 |
| Figure 30. 4DFE sequence fit with ASSM .....   | 77 |
| Figure 31. Multiple databases fit with ASSM .....                                    | 78 |
| Figure 32. Models with roll, pitch, and yaw fit with ASSM .....                      | 78 |
| Figure 33. Sample frames of non-face data fit with ASSM .....                        | 79 |
| Figure 34(a). ASSM 3DFE fit compared to ground truth .....                           | 81 |
| Figure 34(b). ASSM 4DFE fit compared to ground truth .....                           | 82 |
| Figure 35. ASSM comparison to Sun et al .....  | 82 |
| Figure 36(a). Mean normalized error of ASSM .....                                    | 83 |
| Figure 36(b). Mean normalized error of Nair .....                                    | 83 |
| Figure 37. ASSM comparison with MKFT algorithm .....                                 | 84 |
| Figure 38. Sample of component regions .....   | 85 |
| Figure 39. Spontaneous expression segmentation .....                                 | 90 |
| Figure 40. Illustration of landmarks used for pose estimation .....                  | 91 |
| Figure 41. Roll, pitch, and yaw with large deformations .....                        | 92 |

|  |     |
|--|-----|
| Figure 42. Arm and hand models from Microsoft Kinect .....                   | 93  |
| Figure 43. Sample uv landmarks and corresponding 3D shape index values ..... | 101 |
| Figure 44. Sample shape index values on 3D model .....                       | 102 |
| Figure 45. Correlation search .....  | 107 |
| Figure 46. SI-SSM tracked frames with angry expression .....                 | 107 |
| Figure 47. SI-SSM algorithm on Eurecom data .....                            | 109 |
| Figure 48. SI-SSM fit on multiple databases .....                            | 109 |
| Figure 49. Average error of spontaneous expression sequences .....           | 112 |
| Figure 50. SI-SSM tracked joyful sequence .....                              | 112 |
| Figure 51. SI-SSM tracked startled sequence .....                            | 112 |
| Figure 52. SI-SSM average occlusion error .....                              | 113 |
| Figure 53. SI-SSM average error in relation to rotation degree .....         | 114 |
| Figure 54. Si-SSM fit to large rotations .....                               | 114 |
| Figure 55. SI-SSM fit to roll, pitch, and yaw .....                          | 115 |
| Figure 56. SI-SSM tracked surprise expression .....                          | 116 |
| Figure 57. SI-SSM tracked large occlusions .....                             | 116 |
| Figure 58. SI-SSM fit Kinect data with large occlusions .....                | 117 |
| Figure 59. SI-SSM average errors for low quality data .....                  | 117 |
| Figure 60. Illustration of SI-SSM vs. 2D CLM mapped to 3D .....              | 119 |
| Figure 61. MNE of SI-SSM and Nair et al. ....                                | 119 |

### **List of Abbreviations**

|                  |   |
|------------------|---|
| AAM              | Active Appearance Model                       |
| ASM              | Active Shape Model                            |
| ASSM             | Action-based Statistical Shape Model          |
| BP4D-Spontaneous | Binghamton-Pittsburgh 4D Spontaneous          |
| BU-4DFE          | Binghamton University 4D Facial Expression    |
| BU-3DFE          | Binghamton University 3D Facial Expression    |
| DAGSVM           | Directed Acyclic Graph Support Vector Machine |
| FIR              | Fusion Improvement Ratio                      |
| FRGC             | Face Recognition Grand Challenge              |
| FRVT             | Face Recognition Vendor Tests                 |
| HD               | Hand Drawn                                    |
| HMM              | Hidden Markov Model                           |
| ICP              | Iterative Closest Point                       |
| LDA              | Linear Discriminant Analysis                  |
| MD               | Machine Drawn                                 |
| MKFT             | Microsoft Kinect Face Tracking                |
| PCA              | Principal Component Analysis                  |
| ROC              | Receiver Operating Characteristic             |
| SI-SSM           | Shape Index-based Statistical Shape Model     |
| SVM              | Support Vector Machine                        |

3DMM

3D Morphable Model

YSU

Youngstown State University

## **Chapter 1**

### **Introduction**

The field of biometrics is an important one, as being able to measure and analyze human body characteristics has a wide range of applications. These applications range from security, communication, health-care, law enforcement, entertainment, and education. The biometrics community is very active and extremely varied in their approach, due to the wide range of specific topics that this field has. These fields include face, ear, iris, gait, palm, fingerprints, voice, handwriting, DNA, and, recently, studies into multi-modal biometrics (the combination of multiple modalities – ear, face, voice, etc.) have become popular. In recent decades a major focus of the biometrics community has been on studying facial characteristics, such as face recognition, as it is non-intrusive and a large amount of public data is readily available for testing. In studying facial characteristics (recognition, verification, classification, etc.) on 3D data, there is a major pre-processing step that needs to be performed, namely registration of the data. One way to perform this registration is to robustly detect and track 3D features on face models.

While this dissertation aims to study multiple problems and questions in the biometrics community, there are two main focuses. The first being the study of 2D, 3D, and 4D facial feature detection, and the second being the classification of faces and sketches with applications to law enforcement and security.

## **1.1 Problems and Open Questions in the Biometrics Community**

There are many problems and open questions that drive the biometrics community. Some of these problems and questions include:

- (1) How to robustly and accurately identify a human subject?
- (2) How to accurately register 3D models as a pre-processing step for identifying subjects?
- (3) How to perform 3D face sketch recognition without any publicly available test data?
- (4) What is the best domain to perform biometrics analysis in (2D vs. 3D, or a combination of both)?

These are all challenging and important questions that need to be answered to help further extend the current state of the art. This dissertation gives some studies and results for each of the above questions respectively:

- (1) Creating pseudo-3D data from 2D allows for the accurate and robust recognition of subjects.
- (2) Statistical model-based methods can accurately detect and track 3D facial features for registration.
- (3) 3D face sketch data can be accurately created from 2D data to study the challenging task of 3D face sketch recognition.
- (4) 3D data is shown to out-perform 2D data, as well as alleviate the problems inherent with 2D.

These questions help motivate the studies and algorithm development included in this dissertation.

## **1.2 Background and Motivation**

Facial feature detection and tracking is an important first step for many applications that make use of 3D data. Some of these applications include face recognition [15], and expression analysis [151]. The current state-of-the-art methods in this research area include active shape models [57], active appearance models [2], and constrained local models [127]. While these are successful and widely used methods, they are currently limited to 2D data. 2D data has some inherent problems that that can be solved by utilizing 3D data such as pose changes and variations in lighting. This dissertation presents methods that extend these 2D concepts and utilizes 3D data. While 3D data can solve some of the problems inherent with 2D data, it is a non-trivial task in robustly detecting landmark features.

A major component of facial feature analysis (2D, 3D, and 4D) is feature detection and tracking. It is useful in applications such as tele-conferencing, face recognition, entertainment, 3D model registration, and facial expression analysis. Facial feature detection is an extremely important first step in realizing any of the above listed applications. Detecting facial features allows for matching of landmarks between images and models in face recognition. Video segmentation can also be performed by making use of specific facial features that have been detected. Entertainment makes great use of

detected features by matching human facial features to animated models in 3D space. This type of method is utilized heavily in movies.

There has been a number of successful feature detection methods developed in 2D. Active shape models [57] and active appearance models [2] are two seminal and highly successful methods for performing 2D feature detection and tracking. These methods create a statistical model of the object (face) being modeled. These methods have been used for the past two decades and have influenced a significant amount of the current research. One such influence and extension is the creation of a so-called constrained local model [127]. These methods have also heavily influenced algorithms in 3D as well. One very successful method, in 3D, that has influences here is the 3D Morphable Model [58]. This method is used to create 3D face reconstructions from single images, as well as photo-realistic image manipulations.

Although there has been a great deal of research for feature detection and tracking in both 2D and 3D, there are still many open questions in both. Specifically when dealing with 3D data, a major problem/question when using 3D data for feature detection is: What is a good feature to use? In this dissertation we propose two novel methods to define good features to use. The first makes use of the explicit shape of the 3D model to create a so-called *temporal deformable shape model (TDSM)* [96], while the second extends this idea and combines the explicit shape of the 3D model with local data around each of the shape features, to create a so-called *shape index-based statistical shape model (SI-SSM)*.



Solving the problems with using 3D data for features detection is a major motivation for this dissertation.

Another important research topic addressed in this dissertation is face sketch recognition. Law enforcement can use this technology to aid in the apprehension of criminal suspects. Generally, sketches used in forensic investigation are derived from one of two methods. They are generated either from forensic artists (hand-drawn sketches) or from computer software (machine-drawn sketches). Both of these methods are generally used after an eyewitness has given a verbal description of the suspect. Once the sketches have been constructed, the main use of them is to post them in a public place with the hope that someone will recognize the suspect. This process can be extremely slow and inefficient, which gives way to significant need to be able to accurately, quickly, and automatically match a sketch photo to a database of suspect mugshots.

Recently there has been promising work done in 2D face sketch recognition. Tang *et.al* [152][154], perform face sketch recognition by matching sketches to images through turning the face images into sketches to decrease the differences between the forensic sketch and the original image. Liu *et. al* [153], use the idea of local linear embedding to preserve the geometry between the photo and sketch images. Li *et. al* [155] propose to create a realistic face image from a sketch by using a hybrid subspace method. This approach has the benefit of being feasible to use in real-time. Recently, Zhang *et. al* [156] conducted a study of the comparison between human subjects recognition of sketches versus an automated principal component analysis based method. The results of this

study showed that while humans generally recognized the sketches better, the automatic machine-based approach was able to outperform human recognition when the sketches contained less distinctive features.

Being able to accurately identify/classify subjects in both sketches and images is highly demanded and finding a novel way to automate this process is an important and active research topic. Since much of the work in being done entirely in 2D, this begs the question: Is 2D data enough to accurately identify/classify a subject? This question is even more difficult to answer for sketch recognition, as there is no readily available 3D face sketch database. The lack of this type of data poses an interesting problem: how do we access this type of data, to make use of 3D face recognition methods? This dissertation attempts to answer this question by proposing a novel approach to creating 3D sketch data from 2D sketches with detected facial landmarks [147], which is the first work of its kind in the biometrics community. These questions regarding face and sketch recognition/classification are the second major motivation for this dissertation.

### **1.3 Objective and Contributions**

The main focuses of this dissertation involve 2D, 3D, and 4D feature detection and tracking, as well as face and sketch recognition/classification. While there are many problems and questions that need to be answered regarding facial feature detection, and face/sketch recognition/classification, the objective of this dissertation is to make strides in answering the following questions:

- (1) Can we utilize 3D data for instances where 2D data is not sufficient?
- (2) How can we study 3D face sketch recognition and classification when this type of data is not readily available?
- (3) What is a good 3D/4D feature to use for detection and tracking?
- (4) What applications can make good use of the subsequent features and data?
- (5) How do we track extreme data (expression, pose, etc.)?
- (6) Can the variance of a large set of data be, adequately, modeled in a statistical model for feature detection?

To answer these questions this dissertation proposes multiple innovative and novel methods in the fields of feature detection and tracking, and the study of facial characteristics. The major contributions of this dissertation are summarized in the following sub-sections.

### **1.3.1. 2D tracking for eye viewing direction**

Previous works in this field utilized 2D holistic bases approaches or local component based approaches [1][4][5][7][8]. In the first major contribution of this dissertation I propose to extend the state of the art via a novel method of constructing a 3D face model and eyeball from 2D data [145]. A scale-space topographic feature representation is used to model the 3D face and iris. Utilizing this newly constructed 3D face and eyeball we can then accurately determine the subject's eye viewing direction. This 3D-based approach has many advantages compared to utilizing 2D data including resistance to image noise, the eye can be represented in a high level of detail, robustness to eyelids, and no camera calibration is required.

### **1.3.2. Image fusion for face recognition under shadow**

Face recognition is a challenging area of study, and while a great deal of progress has been made it still suffers from problems such as shadows (lighting); occlusion from hair, glasses, and makeup; and changes in pose. This dissertation focuses on the concept of fusing data to increase the recognition rates. Current state of the art methods for fusion based face recognition make use of probabilistic approaches [17], as well as manifolds, and Hidden Markov Models [18][25], however, using these types of methods can have a high cost. I propose a novel method of fusing data by using frames from rotated heads in videos [146]. This approach creates uses the implicit 3D data that is given from using 2D videos of rotated heads. I also propose to study what effects this type of fusion will have on the recognition rates of images under strong shadow, which is a very challenging problem. When comparing the results of fusing 10 frames of rotated heads with using a single frame of the subject; the recognition rate was almost double from 40% to 80%.

### **1.3.3. 3D face sketch recognition**

Face sketch recognition is an important and challenging problem to solve. It can be extremely useful for security and law enforcement. Currently the work being done in this field is from 2D data alone, as there are no readily available 3D face sketch databases. The work being done in 2D involved both sketches done from an artist, as well as from software [33][34]. Face sketch recognition has important implications to forensic applications. This type of data can be useful in a court room setting where the only means of identifying a suspect comes from a witness description via a sketch drawing. I innovate and extend the state of the art by proposing a novel method of constructing 3D

face sketch data from 2D data [147]. In order to construct these 3D models, facial landmarks are firstly detected on 2D sketches, and then a scale-space topographic feature representation is combined with a 3D reference model. 3D sketch recognition experiments are conducted on both 3D models created from artist drawings, as well as sketches created from software. A recognition rate of approximately 92% was achieved.

#### **1.3.4. Dynamic curvature description for 3D facial activity analysis**

Much of the work in facial activity analysis is done in 2D [77][78][79] which suffer from the limitations of pose and lighting variations. I aim to study facial activity analysis in 3D by proposing a new so-called dynamic curvature descriptor [148]. This new descriptor used temporal information (3D+time), by constructing histograms of shape-index information to give an accurate representation of facial activity in a sequence. This newly proposed curvature descriptor is shown to outperform current state of the art facial activity analysis methods.

#### **1.3.5. 3D landmark detection and tracking**

Facial feature tracking has been extensively researched in 2D giving us successful algorithms such as AAM [2], and CLM [127]. Some promising results have been published for 3D facial feature tracking [149][55], however they have large error rates, and are relatively invariant to large pose changes and expressions. I extend the current state of the art by proposing two new statistical-based methods to detect and track features in 3D and 4D. Firstly, I propose a so-called *temporal deformable shape model (TDSM)* [96]. This TDSM makes use of the explicit shape of 3D data by directly

modeling the variation in shape for the training data. This newly proposed method is shown to out-perform current state of the art methods. Secondly, I propose an innovative extension to the TDSM algorithm, by a so-called *shape index-based statistical shape model*. This newly proposed extension not only makes use of the explicit shape of the 3D data, but also the local shape-index data around local features in the model. For both of the methods, I have tested on approximately 100,000 different 3D range models. Both methods are shown to have smaller error rates, and out-perform current state of the art methods. The efficacy of both methods is shown to have applications in entertainment, facial expression analysis, subject identification and verification, pose estimation, and 3D video segmentation.

The rest of this dissertation is organized as follows: (1) first, a study of 2D feature tracking for eye viewing direction estimation is performed in Chapter 2; (2) a study in fusion-based face recognition is given in Chapter 3; (3) 3D facial activity analysis is shown in Chapter 4; (4) Chapter 5 details my innovative 3D face sketch modeling and recognition approach; (5) Chapters 6 and 7 detail my two new statistical model-based algorithms for detecting and tracking features in 3D and 4D; (6) finally, a conclusion for findings, applications, limitations, and a discussion is given in Chapter 8.

## **Chapter 2**

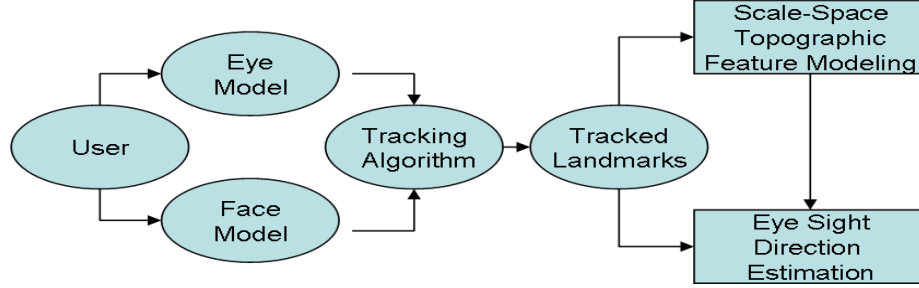
### **Dynamic Face Appearance Modeling and Sight Direction Estimation**

#### **1. Introduction**

Facial appearances and sight orientations can be modeled in a 3D space. Existing 3D dynamic imaging systems [6][13] require a rigorous setup (e.g., short range of capture, user intervention for calibration of multiple cameras, lengthy pose-processing, and strict user cooperation, etc.), thus limiting their applications for human computer interaction. In this chapter, we present a system to model facial dynamic appearance and eye sight direction using a single video camera. We create dynamic 3D models from tracking information obtained from active appearance models and scale-space topographic features, and map them to a 3D space to create a 3D representation for each frame of a face video. We model both the 3D facial region and 3D iris region dynamically and explicitly, allowing an accurate estimation of eye sight directions through a dynamic video. The system framework is outlined in Figure 1.

To model a face and its iris in a dynamic 3D space, feature tracking is the first step needed for the topographic model creation. Our system allows the user to either track the entire face or the subject's eye region separately. The user can select which model they would like to use (face or eye). Here we use an active appearance model (AAM) [2] to track 459 feature points which are defined in the facial region. Since the subsequent eye

modeling requires a multi-scale space topographic representation and multi-size surface patch fitting for topographic label classification, we need to restrict the region of interest for efficient computation. Therefore we further track 8 landmarks to determine the region of interest for eyes.



**Figure 1. System Diagram.**

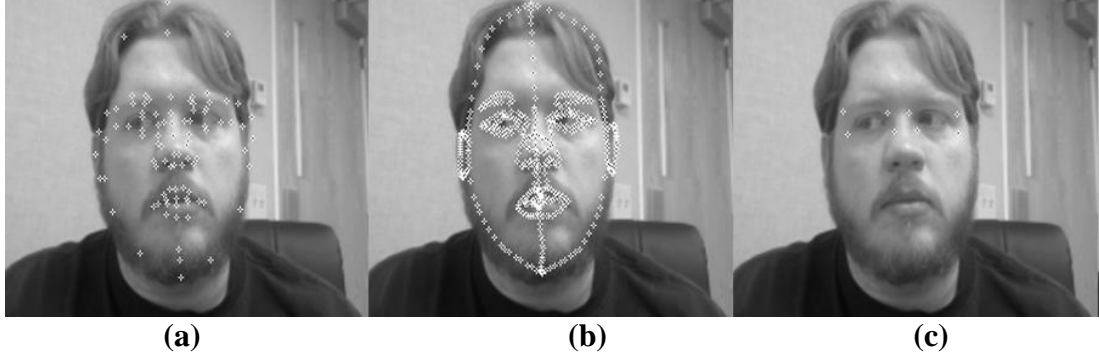
Extended from our previous work on topographic analysis for facial feature modeling [10][12], we propose a new scale-space topographic feature representation approach to model the dynamic facial appearance and iris sphere explicitly. We use a 3D geometric reference model (including a 3D facial surface mesh and a 3D eye mesh) to model individual faces and individual eyes. A multi-step dynamic mesh adaptation method is applied on both facial regions and eye regions to instantiate the model across video sequences. Note that unlike the conventional methods [4][1][5][7][8] for eye tracking and eye gaze estimation, which have used 2D holistic based approaches or local component based approaches, we estimate the eye viewing direction through the explicit 3D iris modeling. This allows for more flexible and reliable eye sight detection under various poses, expressions, and imaging conditions. The rest of the chapter will describe the components for tracking and modeling separately.



## 2. Tracking With Person-Dependent AAM

Active appearance models were introduced by Cootes *et al.* [2][2]. It consists of two separate types of models; one is the variation of the face shape, the other is the variation of the gray level of that shape. These two models are combined together to create a statistical appearance model. During the training phase the user manually selects landmarks that correspond to the most important features on each of the images that will be used for training. After the landmarks are selected each of the landmarks from the images in the training set are warped to match the mean shape. Each set of landmarks are represented as a vector and PCA is applied to them. This can be approximated by the following formulas:  $x = \bar{x} + Q_s c_s$  for shape and  $g = \bar{g} + Q_g c_g$  for texture. In the shape formula  $\bar{x}$  is the mean shape,  $Q_s$  represents the modes of variation and  $c_s$  defines the shape parameters. In the texture formula  $\bar{g}$  is the mean gray level.  $Q_g$  represents the modes of variation and  $c_g$  defines the texture parameters. In various works pertaining to active appearance models 95% - 98% of the variance is usually kept. To conduct our experiments we chose to retain 95% of the variance.

To track the entire face, in real-time, 459 landmark points are used that cover the entire face (Figure 2 (b)). To create a training model where each image contained 459 landmarks would be a cumbersome and time consuming process. To alleviate this challenge we select 92 key points in each of the training set images (Figure 2 (a)). We then interpolate to the required 459 points to track and eventually create the 3D model. The interpolation is done using a Catmull-Rom spline.



**Figure 2. (a) original 92 key points; (b) interpolated 459 points; (c) 8 points for eye region**

To track the eye region the model consists of 8 key points around the eyes (Figure 2 (c)). The points create a “boxed in” region around both of the eyes. This region allows us to set the ROI for a separate eye tracking and eye model creation.

### **3. Scale-Space Topographic 3D Modeling**

#### **3.1 Dynamic 3D appearance for modeling**

Given the feature points tracked, we apply a reference model to align with the tracked points. However, in order to create a 3D model representation for each individual frame, and to estimate the eye sight orientation, we deform the reference model into the non-rigid (non-feature) regions of the face. To do so, we extend our previous work based on an adaptive mesh [12] to a hierarchical topographic scale-space. Here we used three-levels of topographic representations with coarse, medium, and fine structures respectively.

We treat a face image as a topographic terrain surface, and each pixel can be categorized into one of the twelve primitive surface features[10]. The composition of these basic

primitives provides a fundamental representation of different skin surface details. Based on the topographic primal sketch [10] we have developed a topographic face labeling approach to represent and model facial surfaces, and created individual face models by adjusting a generic model [12]. Here is the brief overview of our existing approach. Given an input image, we can determine the topographic feature on each pixel location using a surface patch approximation approach [10]. A continuous surface  $f(x,y)$  is used to fit the local  $N$  by  $N$  patch centered at  $(x,y)$  with the least square error. The topographic label is classified according to the extrema values of the second directional derivative of the surface. After obtaining the first-order and second order derivatives at  $(x; y)$ , we can construct a 2 by 2 Hessian matrix [10]. The feature labeling is based on the values of eigenvalues and eigenvectors, and the gradient magnitude [12][12].

The results of topographic labeling represent different levels of feature details, depending on the variance of the Gaussian smoothing function ( $\sigma$ ) and the fitting polynomial patch size ( $N$ ) (both  $\sigma$  and  $N$  are known as *scales*). The topographic label map associated with the scales is defined as *topographic scale-space*. The existing applications of topographic analysis are limited in a “still” topographic map with a selected scale. As we know, every label may represent various features in a specific image. Various features (e.g., features of the human face) may be “screened out” with various “optimal” scales. A small scale could produce too much noise or fake features. A large scale may cause the loss of important features. Our previous work also shows that too many fake features could cause the model adaptation to be distracted. More seriously, it could make the adaptation unstable, even causing it to not converge. Too few features will not attract the generic

model into the local facial region with expected accuracy. Due to the difficulty to select an “optimal” scale, here we propose to represent the facial features in the topographic scale space, and modeling faces in a hierarchical structure from a coarse level, to a medium level, and a fine level. Such a procedure will ensure the stable convergence of the dynamic mesh to the face region with a constraint of the upper level topographic space, thus resulting in an accurate estimation of 3D facial appearances and their sight directions.

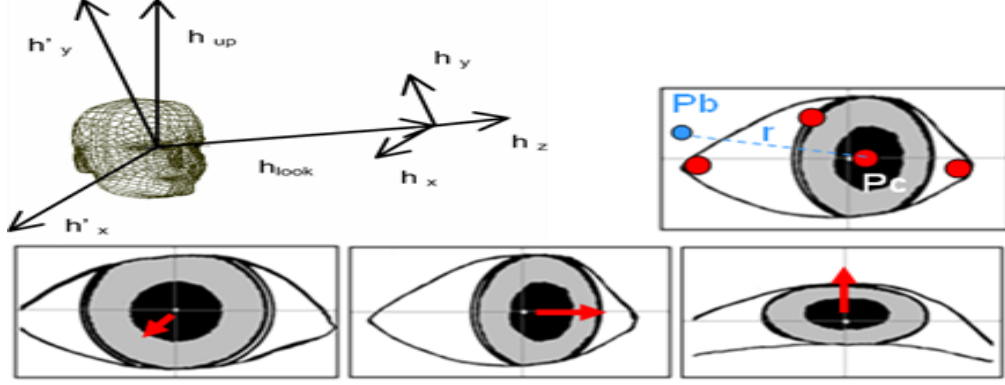
In our modeling process in the topographic scale-space domain, the dynamic meshes are moved by not only the 2-D external force (*e.g.* topographic gradient) but also the depth force (*e.g.* topographic curvature) for model deformation in *multiple scales*. Here we take the model as a dynamic structure in which the elastic meshes are constructed from nodes connected by springs. The external forces of the nodes are used to link the dynamic mesh to the observed face image data. The motion for the dynamic node system is formulated by a second-order differential equation [9] where the node motion is driven by both internal force (*e.g.*, mesh spring stiffness and topographic gradients) and the external force (*e.g.*, topographic curvature and the topographic labels.) .

The model adaptation process is performed by three stages: a coarse adaptation onto the coarse scale of the topographic map, a medium scale adaptation onto a medium topographic map, and a fine adaptation onto the fine scale of the topographic map. The three stages employ the similar adaptation algorithm as described in [12], except for additional constraints assigned to each level of adaptation. Specifically, the second stage

(medium level) requires the node motion in the restricted local topographic region which has been defined by the coarse topographic map, and the node motion for the fine level adaptation is restricted in the regions which have been defined in the medium topographic map. This strategy will prevent the mesh from distraction, and thus result in a stable adaptation. As a result, the mesh can distribute itself in both salient feature areas and facial surface “wave” areas.

### **3.2 3D Iris modeling and sight direction estimation**

Extending the topographic analysis of face features, we applied a scale-space topographic context to conduct an eye model adaptation within the eye region. The procedure is the same as the face model creation procedure as described in Section 3.1. After mapping a model onto the eye region, we can project a ray from the center of the eyeball sphere to the iris center to estimate the eye sight direction. The two 3-D points: centre of eye-ball ( $P_b$ ) and centre of pupil ( $P_c$ ) are illustrated in Figure 3 (upper row). The line linking the two points represents the direction of the eye sight. Note that given the four 3D points obtained from two eye-corners, pupil center ( $P_c$ ), and an arbitrary point on the iris boundary, the eye-ball sphere parameters, center  $P_b$  and radius  $r$ , can be uniquely determined.



**Figure 3. Upper Row: eye sight direction and eye-ball sphere determination based on four points; Lower row: eye sight direction in different views.**

#### 4. Experimental Results

In order to test the accuracy of our system we used three different cameras with different resolutions to capture and track our data. We tested our system with a low, medium, and high resolution setting. For our low resolution tests we used a Logitech QuickCam Orbit AF with a resolution of 320x240 (as shown in Figure 5). We used a Sony network camera SNC-RZ30N with a resolution of 640x480 for our medium resolution tests (as shown in Figure 6). Finally, for our high resolution tests we used a Di3D [6] capturing system that creates texture images with a resolution of 1040x1392 (as shown in Figure 4).

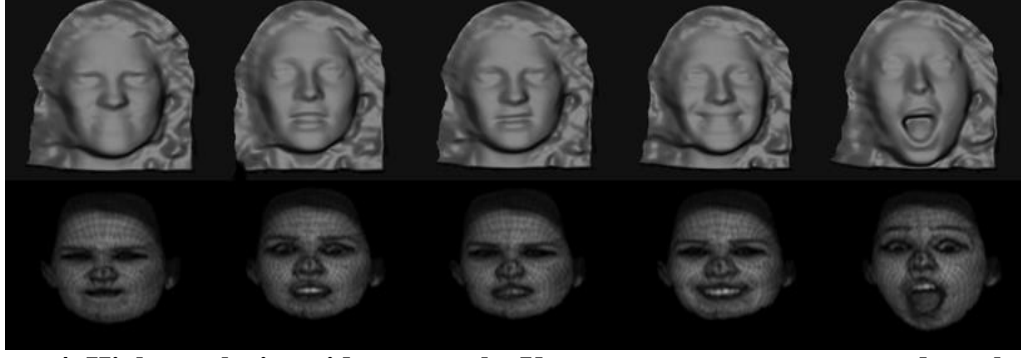
Although the eye region is contained in the entire face, we found that it is beneficial for us to track the eye region separately. Since the only information that we need is where the landmarks are located, we have found it easier to only select the landmarks around the eyes instead of extracting this information from the face. Also, there are instances where we found it difficult to successfully track a subject's face but we were able to track the eye region. We believe that this is due to our use of a person-independent active

appearance model. Gross *et al.* [3] noted that it is harder to fit a generic AAM compared to a person specific AAM due to the high dimensionality of the shape model.

#### **4.1 Evaluations**

In order to evaluate the accuracy of the geometric shape of our created models, we used the 3D dynamic range scans [13][13] captured from Di3D imaging system [6][6] as the ground-true data for comparison (Figure 4).

The ground true face model contains 35,000 vertices; our created model has about 2,900 vertices. We used both 3D range model scans and our generated models (300 frame models), and manually selected 92 feature points on each model in areas of mouth, facial contour, nose sides, nose bridge, eyes, eyebrows and cheek. After normalizing all the models into a range of (-50, +50) in three coordinates of x, y and z, we calculate the mean square error (MSE) between the two sets of 3D surface feature points. The result shows that the average MSE of 300 frames models is 6.74. This is much less than the MSE (=12.7) when we compare the coarse models to the range models. In addition, the estimated eye directions from our generated models are also compared to the eye directions of the range models. Among 300 frames, 249 frames show less than 5 degree difference between two data sets.



**Figure 4. High-resolution video example. Upper: range scans as ground-truth; Lower: generated models(3D meshes overlapped on textures).**

There are three major advantages of the proposed 3D model based approach: (1) the modeling procedure relies on the multiple-scale model adaptation in a global face space rather than very few individual points in local facial regions. It is more resistant to image noises under various imaging conditions; (2) the three-levels of topographic features allow the face and eye representations in a high level of detail, and (3) the eye sphere estimation is based on the four points including the eye center and eye corners and excluding the eyelid points. It has certain robustness to occlusion from eyelids. Unlike other conventional 2D tracking systems, our 3D model based eye sight estimation does not require any calibration of cameras.





**Figure 5. Low-resolution videos for two subjects. Tracked feature points; generated models; and detected eye sight directions(shown as red arrows).**

## **5. Discussion**

In this chapter we have presented a scale-space topographic modeling approach to model the dynamic facial appearance and eye sight directions. The experimental results are encouraging. While we are able to track face movements and eye sight orientations under various resolutions, backgrounds, and expressions, the tested pose changes are still in a small range. Issues dealing with pose change can better be handled with explicit 3D data which we will detail in later chapters. While this is an effective method to create the 3D models, the actual deformation of the generic mesh is a fairly expensive operation and further study into parallel algorithms will be needed for a real-time application. Next we will study the effect multi-frame fusion on face classification.

## Chapter 3

### Fusion Based Face Classification Under Strong Shadows

#### 1. Introduction

The majority of face recognition researches dealt with still images acquired under a somewhat controlled setting. The performance improvement of recognition technologies using those images has been impressive, as evidenced by the results of Face Recognition Vendor Tests [14]. However, the current methods still have difficulties handling data obtained under more challenging conditions, such as strong shadows, severe occlusions, or large pose variations. To deal with those problems, various approaches have been proposed, including 3D face methods [15], video-based methods [16][17][18], correlation-filters[30], multi-view methods [19][20], and multi-sample/multi-instance methods [21][22][23].

In this chapter, we examine the performance of a fusion method that integrates multiple frames selected from rotating head videos. The objective was to determine whether and how the multi-frame fusion can overcome the adverse shadow effect to achieve a significantly better recognition rate. We addressed two fundamental issues: (i) How effective is multi-frame fusion in handling shadowed faces, if a sophisticated pre-processing or fusion method (such as a probability density based method) is not involved? (ii) Does a multi-frame fusion yield a consistent performance gain? More importantly, can we quantify its performance in terms of its data composition?

This study has several features: (i) It used a video dataset of 257 subjects, which is comparable to that of Multi-PIE database [24]; (ii) Frames of ten pose angles were automatically selected; (iii) Because of the regular frame interval, the temporal continuity is preserved that characterizes a full head rotation; (iv) A large number of fusion tests were conducted.

## **2. Related Works**

Video-based face recognition bears resemblance to the methods of using multiple still images, but the former may deal with a much larger number of frames. Chellappa *et al.* [17] have developed a probabilistic framework that explores the temporal continuity of face motion. Other approaches of using manifolds and hidden Markov models were also proposed [18][25]. A probabilistic approach has several advantages: (i) It tackles tracking and recognition simultaneously; (ii) It is flexible to handle both video-to-image and video-to-video matches; (iii) A 3D model can be incorporated. However, the computational cost could be high, especially if a very small frame interval is required to satisfy continuity constraints. Using a high resolution 3D model (e.g., a deformable finite element model) in a video-to-video scenario is even more demanding.

Another popular strategy is to utilize a small number of representative images and consolidate the results through a fusion. Many methods can be put into this category, such as multi-view method, multi-instance method and multi-sample method. Thomas *et al.* [26] and Canavan *et al.* [27] found that recognition rate can be greatly improved using fused video frames. Faltemier *et al.* [21] applied a similar strategy to a 3D face dataset

and found that the multi-instance method outperforms a component based method. Recently, a mosaicing approach was proposed that utilizes a composite model from images of different poses [19].

### 3. Multi-Frame Fusion Based Method

#### 3.1 Video Dataset

Videos of 257 subjects were collected in two sessions. The second session occurred about 5-9 weeks after the first one. 167 subjects attended both sessions and 90 subjects appeared in the first session only. During each session, subjects rotated their heads in the range of  $0^\circ$  to  $90^\circ$ . Two illumination conditions were considered: (i) Normal indoor lighting; (ii) Strong shadow. Figure 6 shows some examples.



**Figure 6. Upper four rows: Samples showing the different views with two different illuminations. Bottom row shows an example of two-sessions: Left four: first session with two different illuminations; Right two: second session of the same subject.**

### **3.2 Frame Selection**

Ten frames were selected from each video corresponding to ten pose angles ( $0^\circ$ ,  $10^\circ$ ,  $20^\circ$ ,  $30^\circ$ ,  $40^\circ$ ,  $50^\circ$ ,  $60^\circ$ ,  $70^\circ$ ,  $80^\circ$ ,  $90^\circ$ ), with  $0^\circ$  for the frontal view and  $90^\circ$  for the profile view. Both manual and automatic methods were used. Manually selected frames were used to benchmark the automatically selected ones. We applied a PCA approach for automatic pose estimation. We collected training data from BU-3DFE database [31] with ten different views from  $0^\circ$  to  $90^\circ$ . After applying the PCA transformation, we obtain the eigen-faces with different views. In the eigen-space, ten clusters are clustered corresponding to ten poses. Given a face image, we project it to the eigen-space and classify to one of the cluster using a K-NN classifier. Following this procedure, we detect ten poses from the video input (see [32] for details). The automatic pose detection process allows us to study the multiple-pose fusion performance in the subsequent experiment.




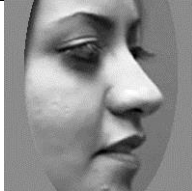
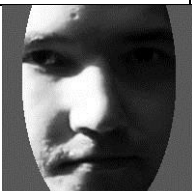



### **3.3 Training, Gallery and Probe sets**

The training set contains 90 subjects who appeared only in the first session. The gallery/probe sets include 167 subjects who enrolled in both sessions. The gallery has the frames of normal lighting condition, while the probe has frame of shadows (Table 1). This protocol is similar to that of FRVT 2006 [14], which ensures the independence between the training and test data.

**Table 1. Training, Gallery, and Probe sets.**

| Training  | Gallery   | Probe   |
|---|---|---|
| 90 subjects.<br><br>In the 1st session only.<br><br><b>Normal + Shadow.</b> | 167 subjects.<br><br>In the 1st session.<br><br><b>Normal lighting.</b> | 167 subjects.<br><br>In the 2nd session.<br><br><b>Strong shadow.</b> |

It should be emphasized that, besides shadows, a few other factors make the dataset very challenging. As shown in Figure 7, there exist large discrepancies between the appearances of the same person in gallery and probe, which could be caused by facial expressions, glasses, jewelry, mustaches and long hair.

| Pose    | 0°  | 20°   | 40°  | 60°   |
|---------|---|---|--|---|
| Gallery |  |  |  |  |
| Probe   |  |  |  |  |
| Factors | Shadows   | Glasses   | Expression   | Long Hair   |

**Figure 7. Large differences between faces in the gallery and probe sets that could cause problems to the methods that use a single image per subject.**

### 3.4 Fusion Schemes

Each of the facial poses provides a matching score, which is a similarity measure (e.g., distances) between the images. We used a score level fusion method [22] that was implemented in two steps. In the first step, ten basic score matrices were generated using a PCA (Principle Component Analysis) eigen-face method [28][29], one for each of the ten pose angles. For example, to create a basic score matrix for the  $20^\circ$  pose angle, a PCA test would be run using only the frames of  $20^\circ$  in the training, gallery and probe sets. In the second step, fusions were carried out by combining the subsets of ten basic matrices with the *sum rule* [22][27] (i.e., summation of the scores.) Therefore, an exhaustive evaluation requires a total of 1023 fusion tests:  $1023 = C(10, 1) + C(10, 2) + \dots + C(10, 10)$ , where  $C(n, k) = n!/(k!(n-k)!)$  is the binomial coefficient (see Table 2).

**Table 2. Exhaustive Fusion Tests.**

| Fusion Group     | Examples of frame combinations  |
|------------------|---|
| $C(10, 1) = 10$  | $(0^\circ), (10^\circ), (20^\circ), (30^\circ), (40^\circ), (50^\circ), (60^\circ), (70^\circ), (80^\circ), (90^\circ)$ |
| $C(10, 2) = 45$  | $(0^\circ, 10^\circ), (80^\circ, 90^\circ)$   |
| $C(10, 3) = 120$ | $(0^\circ, 10^\circ, 20^\circ), (40^\circ, 80^\circ, 90^\circ)$   |
| $C(10, 4) = 210$ | $(0^\circ, 10^\circ, 20^\circ, 30^\circ), (10^\circ, 20^\circ, 60^\circ, 90^\circ)$                                     |
| $C(10, 5) = 252$ | $(0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ), (10^\circ, 30^\circ, 40^\circ, 60^\circ, 80^\circ)$                 |
| $C(10, 6) = 210$ | $(0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ)$   |
| $C(10, 7) = 120$ | $(0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ)$   |
| $C(10, 8) = 45$  | $(0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ, 70^\circ)$                                       |
| $C(10, 9) = 10$  | $(0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ, 70^\circ, 80^\circ)$                             |
| $C(10, 10) = 1$  | $(0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ, 70^\circ, 80^\circ, 90^\circ)$                   |
| Total = 1023     |   |

### 3.5 Measuring Inter-frame Variation

In order for a multi-frame method to be effective, the frames used in a fusion should be as diverse as possible (i.e., smaller similarity). So, we adopted a similarity measure based on

the mutual information. For two frames  $A, B$ , let  $P_A(a)$  be the probability density that a point chosen (uniformly) at random is of intensity  $a$  in frame  $A$ , and let  $P_{A,B}(a,b)$  be the joint probability density that a point chosen at random is of intensity  $a$  in frame  $A$ , and the same point is of intensity  $b$  in frame  $B$ . Then the similarity measure  $I(A,B)$  is defined as follows:

$$I(A,B) = \iint P_{A,B}(a,b) \log \left( \frac{P_{A,B}(a,b)}{P_A(a)P_B(b)} \right) da db \quad (1)$$

Using  $I(A,B)$ , we devised an inter-frame variation metric for a 2-frame fusion:

$$\tau_2(i,j) = \frac{\sum_{k=1}^N \left( \frac{1}{I_k(i,j)} \right)}{N}, i,j \in [0^\circ, 10^\circ, \dots, 90^\circ] \quad (2)$$

where  $\tau_2$  denotes inter-frame variation,  $N$  is the size of a data set. In other words,  $\tau_2$  measures the dissimilarity of two frames averaged over all subjects in a data set. In case that a fusion has more than two frames, we first calculate the  $\tau_2$  values of all possible 2-frame pairs and then take their average as the  $\tau$  of that fusion.

**Table 3. Statistics of Rank-1 Fusion Tests.**

| Fusion Group     | Rank-1 Rate |      |         |           |
|------------------|-------------|------|---------|-----------|
|                  | Min         | Max  | Average | Std. Dev. |
| $C(10, 1) = 10$  | 0.31        | 0.48 | 0.39    | 0.05      |
| $C(10, 2) = 45$  | 0.41        | 0.62 | 0.53    | 0.05      |
| $C(10, 3) = 120$ | 0.50        | 0.71 | 0.62    | 0.05      |
| $C(10, 4) = 210$ | 0.56        | 0.78 | 0.67    | 0.04      |
| $C(10, 5) = 252$ | 0.59        | 0.81 | 0.71    | 0.04      |
| $C(10, 6) = 210$ | 0.66        | 0.81 | 0.74    | 0.03      |
| $C(10, 7) = 120$ | 0.70        | 0.81 | 0.76    | 0.03      |
| $C(10, 8) = 45$  | 0.73        | 0.81 | 0.77    | 0.02      |
| $C(10, 9) = 10$  | 0.75        | 0.81 | 0.78    | 0.02      |
| $C(10, 10) = 1$  | 0.78        | 0.78 | 0.78    | N/A       |

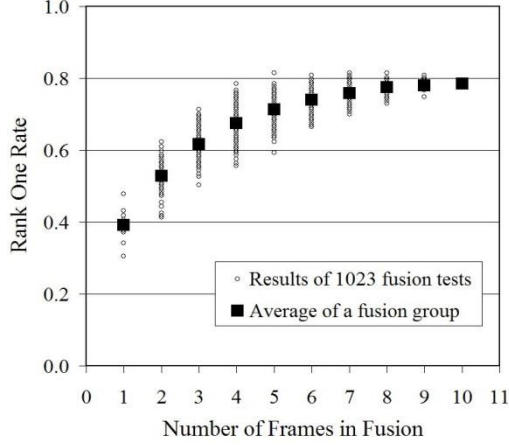


## **4. Experimental Results and Discussions**

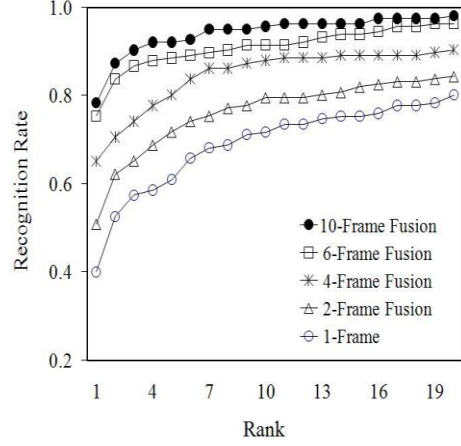
### **4.1 Improvement by Multi-frame Fusion**

The rank-1 rates of 1023 fusion tests were summarized in Table 3, and were plotted in Figure 3 and Figure 4 for CMC curves of a fusion test series. It is clear that the performance of multi-frame fusion steadily improves as the number of frames increases. On average, the fusion method almost doubled the recognition rate, from 40% with a single frame to 80% with ten frames. This is a significant improvement, considering that the dataset used is quite challenging.

In a fusion group that has the same number of frames, the recognition rate showed some fluctuations. For example, in the 3-frame group, the fusion of  $(0^\circ, 40^\circ, 90^\circ)$  had the highest recognition rate of 0.713, while the fusion of  $(70^\circ, 80^\circ, 90^\circ)$  had the lowest value of 0.503. However, as the number of frames in a fusion increased, the differences among individual fusion tests became less noticeable. At the same time, the fusion performance also leveled off. Adding more frames would not lead to a sizable performance gain. This saturation effect was also observed in other studies [26][27], suggesting the existence of a performance upper-bound that is likely dependent upon the quality of dataset being used as well as the efficiency of recognition and fusion algorithms.



**Figure 8. Relationship between the rank-1 rate and the number of frames used in fusion. For each group of fusion tests that contains the same number of frames, its average is also shown.**



**Figure 9. The CMC curves of a fusion test series:  $(0^\circ)$ ,  $(0^\circ, 010^\circ)$ ,  $(0^\circ, 10^\circ, 20^\circ)$ , ...,  $(0^\circ, 10^\circ, 20^\circ, \dots, 80^\circ 90^\circ)$ . For visualization purposes, only 1-frame, 2-frame, 4-frame, 6-frame, and 10-frame tests are shown.**

#### 4.2 Inter-frame Variation

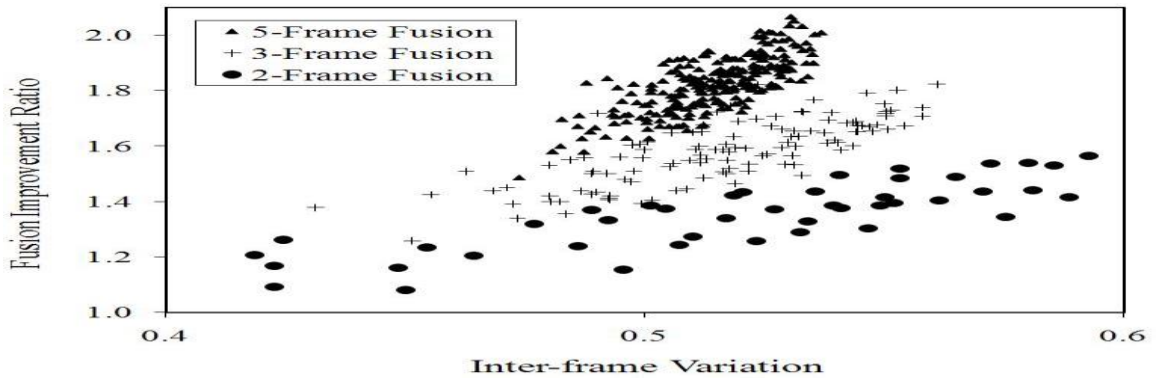
Since a fusion group of the same number of frames but different combinations showed large recognition rate variations, it is important to seek the underlying cause in a quantitative fashion. To this end, we calculated an inter-frame variation value for each fusion test using Eq. (2). The results of three representative groups (2-frame, 3-frame and 5-frame) were plotted against the Fusion Improvement Ratio (FIR) in Figure 10. The FIR was computed by:

$$FIR = \frac{R_m}{\left( \frac{\sum_{i=1}^m r_i}{m} \right)} \quad i \in m \quad (3)$$

where  $R_m$  is the recognition rate of an  $m$ -frame fusion,  $r_i$  is the single-frame recognition rate using the  $i$ th member of the  $m$  frames. So, FIR measures the performance improvement of an  $m$ -frame fusion over the average of its individual members.

A positive correlation between the FIR and the inter-frame variation can be observed (Figure 10). This suggests that a fusion of more diverse samples is likely to produce a better recognition rate. For example, in a 4-frame group, the fusion of  $(0^\circ, 20^\circ, 40^\circ, 90^\circ)$  had the highest recognition rate of 0.784, while the fusion of  $(60^\circ, 70^\circ, 80^\circ, 90^\circ)$  gave the lowest rate of 0.557. Apparently,  $(0^\circ, 20^\circ, 40^\circ, 90^\circ)$  is more representative of a full 90 degree head rotation than  $(60^\circ, 70^\circ, 80^\circ, 90^\circ)$  is, because the faces in  $60^\circ, 70^\circ, 80^\circ$ , and  $90^\circ$  poses are very similar to each other (see Figure 1). In other words, the first fusion combination reveals more about the 3D shape of a face than the second one does.

The above observations is also extendible to the multi-sample approach, multi-enrollment approach, and even the multi-modal approach, where the selection of samples or biometric modalities should be guided by certain inter-sample or inter-modality variation index in order to maximize the performance gain.



**Figure 10. Inter-frame variation and the FIR (Fusion Improvement Ratio) relationship.**

## **5. Discussion**

This chapter presented a multi-frame fusion study and evaluation that exploits the coherent intensity variations in head rotation videos to facilitate recognition under adverse shadow conditions. It is a fairly efficient algorithm as the score based fusion is fast and has minimal overhead. An interesting extension would be parallel algorithms for the recognition. Each of the degrees could be recognized in parallel to help speed up the algorithm even more. Based on the tests of 1023 fusion combinations using 257 subjects and 10 frames per subject, the following observations can be made: (i) multi-frame fusion is an effective method to improve video face recognition. In a multi-frame to multi-frame scenario, the recognition rate was almost doubled; (ii) the performance of a particular fusion choice has a strong connection to its inter-frame variation.

## **Chapter 4**

### **3D Face Sketch Modeling and Recognition**

#### **1. Introduction**

Face sketches can be drawn either by a trained police artist or using a composite software kit [33][34]. Both types of sketches have been studied in the context of searching or matching a sketch to a subject's face in a database of photos or mug-shots [35][36][37][38][39][40]. Since all existing works were based on 2D sketches, issues of pose variations are still challenging. Recently, 3D face recognition has attracted much attention [14] [15][43][44]. Along the same vein, 3D sketch models reconstructed from 2D sketches may improve sketch recognition performance. In order to increase the accuracy of geometric surface matching and efficiency of similarity measurement between 3D faces and probe sketch data, it is highly demanded to have 3D sketches matched up with the 3D scan models. Nevertheless, there is little investigation reported on 3D sketch modeling and 3D sketch recognition in the past.

In this chapter, we address the issue of 3D sketch model construction from 2D sketches, and compare the 3D sketch models with the corresponding 3D facial scans. We further validate the models by conducting 3D face sketch identification on two 3D face databases. Note that there is no existing graphic tool for 3D sketch model construction from witness' description directly. One solution is to create 3D sketch models based on 2D sketches from hand-drawings by artists or conversion from 2D images [41][42].

To build 3D sketch models, we applied a scale-space topographic feature representation approach to model the facial sketch appearance explicitly. We initially tracked 92 key facial landmarks using an active appearance model (AAM) [2], and then interpolated to 459. From the interpolated landmarks, we used a 3D geometric reference model to create individual faces. Based on the topographic features obtained from the sketch images, we applied a mesh adaptation method to instantiate the model.

In order to assess the quality of created 3D sketch models, we conducted a comparison study between the created 3D sketch models and their corresponding ground-true 3D scans. We show the difference between two data sets as well as the difference between the 3D sketch models created from hand-drawn sketches and the 3D sketch models created from machine-derived sketches. Moreover, in order to validate the utility of the 3D sketch models, we propose a new approach to decompose the 3D model into 6 independent component regions, and apply a spatial HMM model for sketch model recognition. The 3D sketch face recognition experiment is conducted on two databases: BU-4DFE [13] and YSU sketch database [46].

## **2. Source Data**

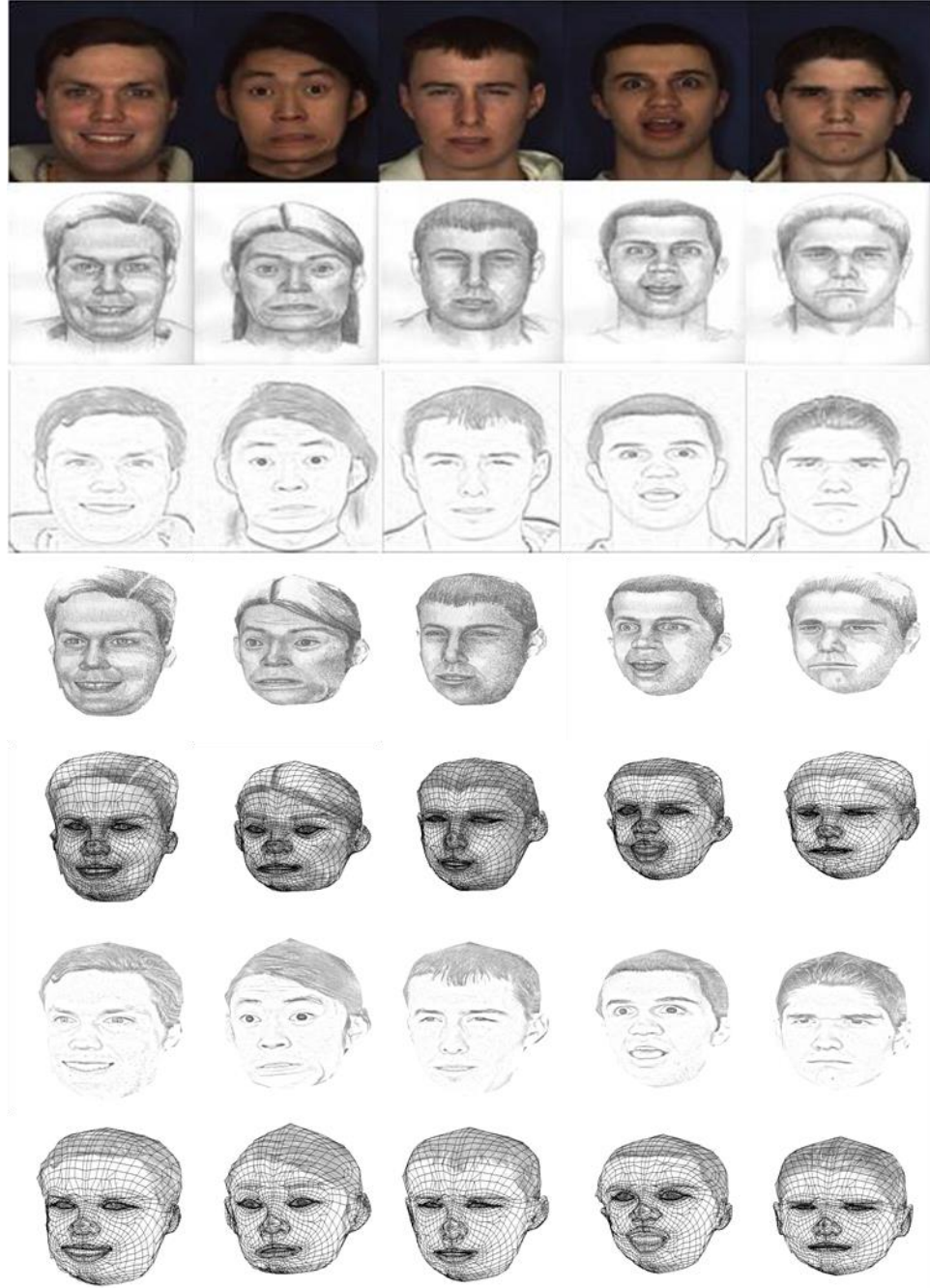
Face database BU-4DFE [13] and YSU sketch database [46] have been used as our data source. Sample 3D scans of BU-4DFE are shown in Figure 11. The corresponding 2D textures are shown in Figure 2 (first row).

Based on six subjects of 4DFE, two forensic artists from Youngstown State University drew the corresponding 2D sketches of the same subjects. Thus we obtained Hand-Drawn (*HD*) sketch images of six subjects. Figure 2 (row-2) shows several samples of HD 2D-sketches.

Due to the time-consuming and intensive work of artist drawing, we created 2D sketches from 2D texture images of the 4DFE database. Our method can simulate the pencil sketch effect. The texture to sketch conversion follows a three-step image processing procedure: First, the image is processed by a de-saturation process, then the image is inverted. After applying a color dodge, the image is blurred with a Gaussian filter. Finally, the radius of pixels is adjusted to get an ideal sketch effect. Figure 12 (row-3) shows examples of MD 2D-sketch images. We have also used YSU hand-drawn 2D facial sketches with 250 sketches. Figure 13 (row-1) shows examples of YSU HD 2D-sketches.

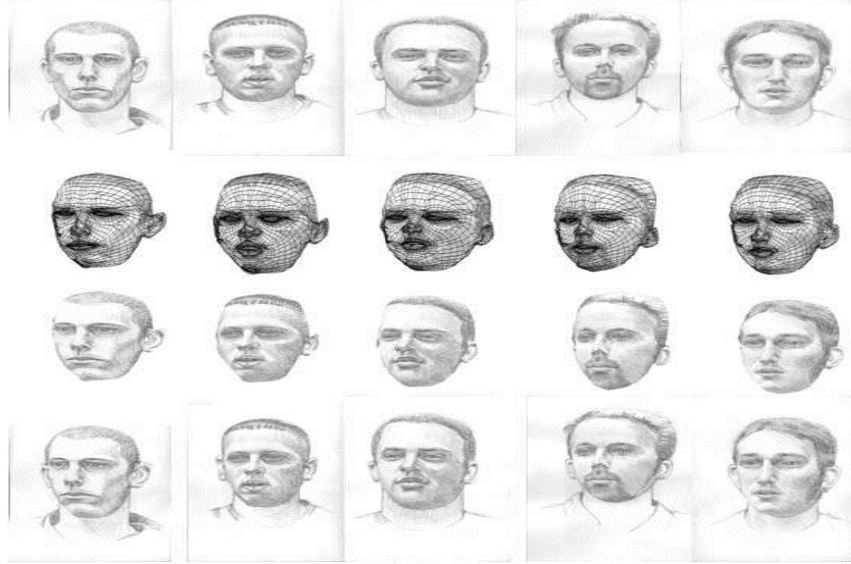


**Figure 11. Examples of 3D scans from 4DFE: textured models shown in upper row and shaded models in bottom row**



**Figure 12. Examples of 2D images of 4DFE from top to bottom: Original textures (row-1); hand-drawn(HD) 2D sketches (row-2), and Machine-derive(MD) 2D sketches(row-3). Rows 4-5: Created 3D sketches from HD sketches with textures and mesh models in different views. Rows 6-7: Created 3D sketches from MD sketches with textures and mesh models in different views.**



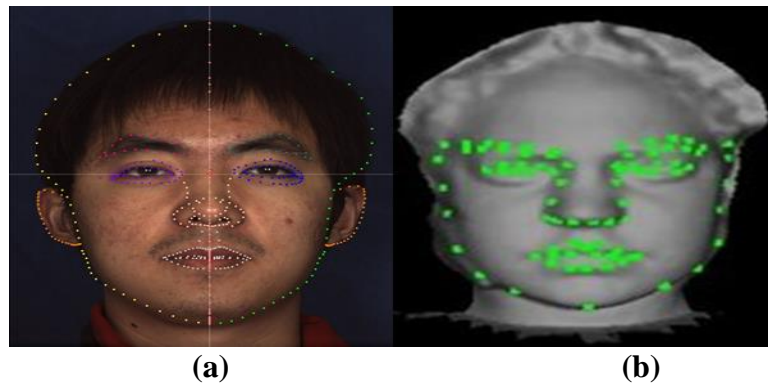


**Figure 13. Samples of YSU 2D sketch database(row-1) and the reconstructed 3D sketches(rows 2-3). Row-4 shows synthesized 3D sketches with rotated heads on the corresponding shoulders.**

### **3. 3D Sketches Creation from 2D Sketches**

#### **3.1 3D sketch reconstructions**

To build 3D sketch models, we developed a scale-space topographic feature representation approach to model the facial sketch appearance explicitly. Using an AAM we initially tracked 92 key facial landmarks, and then interpolated them to 459 using a Catmull-Rom spline [12] (as shown in Figure 14 (a)).



**Figure 14. (a) Illustration of 459 point on a sample face; (b) 83 features point for evaluation.**

From the interpolated landmarks, we used a 3D geometric reference model to create individual faces. The reference model consists of 3,000 vertices. Based on the topographic labels [45] and curvatures obtained from the sketch images, we then applied a spring-mass motion equation [9] to converge the reference model to the sketch topographic surfaces in both horizontal and depth directions. Existing topographic labeling approaches can create different levels of feature detail, depending on the variance of the Gaussian smoothing function ( $\sigma$ ) and the fitting polynomial patch size ( $N$ ) (both  $\sigma$  and  $N$  are called *scales*). The existing applications of topographic analysis are limited in a “still” topographic map with a selected scale. Every label may represent various features in a specific image, various features (e.g., on the organs in the human face) may be “screened out” with various “optimal” scales. The idea of scale is critical for a symbolic description of the significant changes in images. A small scale could produce too much noise or fake features. A large scale may cause the loss of important features. Too many fake features could cause the model adaptation be distracted. More seriously, it could cause the adaptation to be unstable, (e.g., even not converge). Too few features will not attract the generic model into the local facial region with expected accuracy. Due to the difficulty to select an “optimal” scale, here we use a multi-scale analysis approach to represent the topographic features from a coarse level to a fine level as the scale varies. Applying the topographic labeling algorithm with different scales, we generated the topographic label maps of facial images at different levels of detail. Different scales will be applied to different levels of details of sketch images (e.g., hand-drawn (fine details) or machine derived (coarse details)).

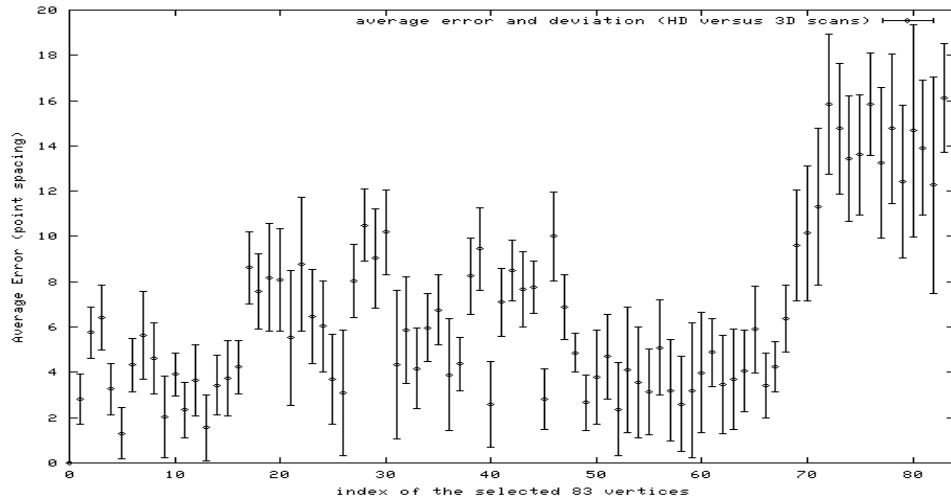
In order to deform the face model into the non-rigid facial area, we applied the adaptive mesh [9] to the facial areas in the *multi-scale* topographic domain. Such dynamic meshes are moved by not only the 2-D external force (*e.g.* topographic gradient) but also the depth force (*e.g.* topographic curvature) for model deformation in *multiple scales*. We take the model as a dynamic structure, in which the elastic meshes are constructed from nodes connected by springs. The 3D external force is decomposed into two components: the gradients of the topographic surface are applied to the image plane, and the curvatures of the topographic surface are applied as a force to pull or deflect meshes in the direction perpendicular to the image plane. As a result, the 3D shape of mesh becomes consistent with the face surface. This procedure was performed based on a series of numerical iterations until the node velocity and acceleration were close to zero. Such a mesh adaptation method was applied to sketch regions to instantiate the model. Figure 2 and Figure 3 shows examples of 3D sketch models reconstructed from 2D sketches for both HD and MD data.

### 3.2 3D sketch accuracy evaluation

#### (1) Comparison: 3D HD sketches vs. 3D scans

We also conducted an objective evaluation, by which we calculated the error between the feature points on the individualized sketch models and the corresponding manually picked points on the face scans. We selected 83 key points as the ground truth for assessment (see Figure 14 (b)). After creating sketch models from 4DFE, we conduct a quantitative measurement as follows: First, we normalize all the models into a range of  $[-50, 50]$  in three coordinates of  $x$ ,  $y$ , and  $z$ . We then calculate the mean square error (MSE)

between the feature points of the 3D sketch and the ground truth of a set of models. We define the one-point spacing as a closest pair of points on the 3D scans, which is approximately 0.5mm on the geometric surface of the 4DFE models. The mean error of two models can be computed by the average of point differences between two models. Figure 15 shows the error statistics, which is the average error and standard deviation on each of the 83 key points. The result shows that the MSE of the examined points is 8.71 point spacings. The average error ranges from 1 to 16 point spacings, with the most of points being less than 10 point spacings. The errors mainly lie in the left side and right side of the face contour and chin area, which are points 69-83.

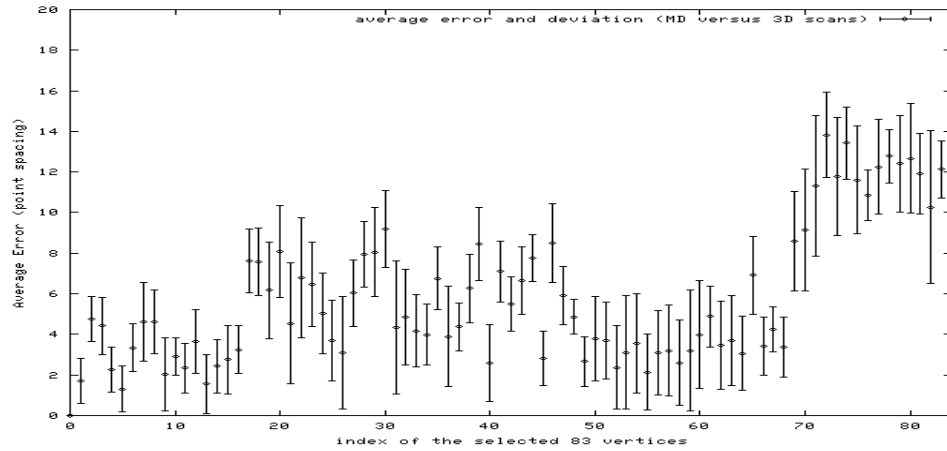


**Figure 15. Error statistics of selected 83 testing vertices of a set of models (3D HD-sketch models and 3D scans). Mid-point of each line represents an average error of the vertex (MSE). The standard deviation is shown by the length of the line.**

## (2) Comparison: 3D MD sketches vs. 3D scans

Similar to the above assessment, we also compare the difference between the 3D scans and 3D sketch models created from the MD sketches. Figure 16 shows the error statistics of the 83 points of 100 models. The MSE of the examined points is 6.84 point spacings.

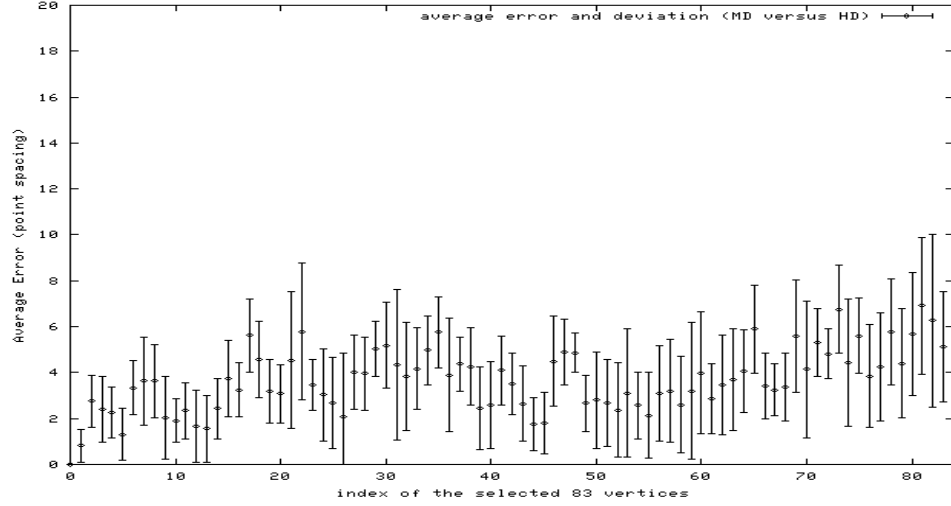
The MD sketch models show more accuracy than the HD sketches. The reason is that the MD sketches are derived from the 2D textures of 4DFE, thus, a better alignment can be obtained between the 3D sketches and 3D scans.



**Figure 16. Error statistics of selected 83 testing vertices of a set of model (3D MD-sketch models and 3D scans).**

### (3) Comparison: 3D HD vs. MD

In order to examine the similarity of the HD sketches and MD sketches. We compare the 3D sketch models created from hand drawn sketch image to the 3D sketch models created from machine generated sketch images. The MSE of the 83 points among those models is 3.78 point-spacings, which are very similar to each other. Figure 17 shows the error statistics. The results justify the approximate equivalence of both MD models and HD models, which can be used by subsequence study for face recognition.



**Figure 17. Error statistics of selected 83 testing vertices of a set of models (3D MD-sketch and 3D HD-sketch models).**

#### **4. 3D Sketch Face Recognition**

In order to validate the utility of the created 3D sketch models, we conducted experiments of 3D sketch model identification. To do so, we segment each sketch model and each scan model into six component regions. A conventional set of surface label features are used for the spatial HMM classification.

##### **4.1 Component region segmentation**

Given a 3D sketch model and the tracked feature points, we can easily segment the facial model into several component regions, such as the eyes, nose and mouth. However, without any assumption of feature points detected on the 3D scans, it is needed to automatically segment facial regions by a more general approach. We developed a simple yet effective approach for 3D facial component segmentation. This approach is general enough to be applicable to other kinds of mesh models, including 3D sketch models. The

component segmentation works on the geometric surface directly. It includes mainly two steps:

(1) *Edge Vertices (EV) determination*: Since the edge-feature-rich regions of a 3D facial model lie in regions of eyes, mouth, and nose, we detect edge vertices based on their vertex normals. To do so, a normal mapping scheme is used, where each vertex is assigned by a pseudo-color  $\mathbf{p}_c = (r, g, b)$ .  $\mathbf{p}_c$  is assigned by the corresponding vertex normal  $\mathbf{n}$ , i.e.,  $\mathbf{p}_c = \mathbf{n} = (n_x, n_y, n_z)$ . Emulating the color to grayscale conversion, each vertex is assigned by an attribute value  $v_a$ :

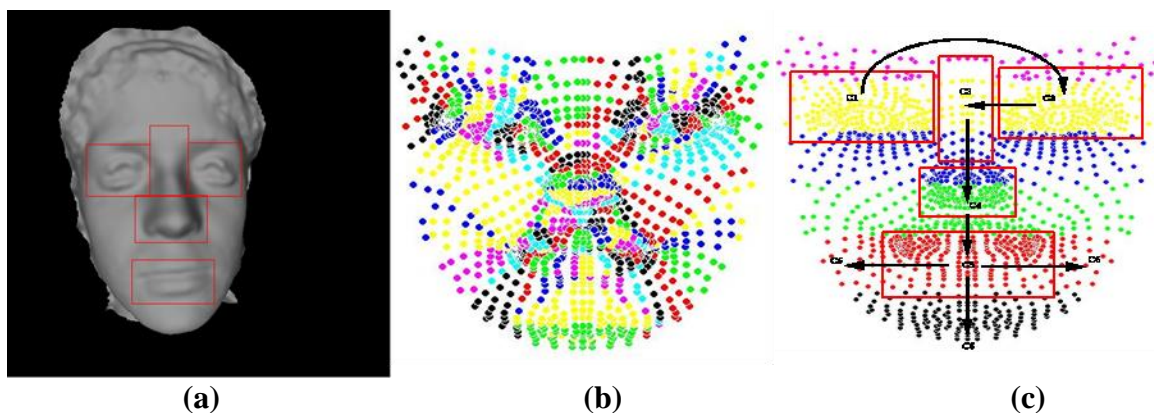
$$v_a = 0.299 |n_x| + 0.587 |n_y| + 0.114 |n_z| \quad (4)$$

Initially, an edge vertex can be calculated by iterating through the neighbors of each vertex and calculating the difference ( $d_a$ ) between  $v_a$  of the vertex and the average  $v_a$  of its neighbors. Thresholding on  $d_a$  values could indicate the edge vertices, however, it may not generate a reliable result. Rather than using a threshold, we apply a clustering method to get two groups of vertices: *edge* and *non-edge vertices*. To do so, a  $k$ -means clustering algorithm is applied, where  $k=2$ , to determine the two groups according to the  $d_a$  values. Whichever cluster a vertex is closer to (*edge* or *non-edge*), that vertex will be added to the corresponding cluster. This procedure is iterated until the centroids of the clusters remain unchanged.

Once we have obtained these edge vertices, a rectangular bounding box of the face model can be determined by a convex-hull of the edge vertices. We can also find the vertex with

the highest  $z$  value of those edges, which is the vertex closest to the nose tip. Note that the top one-fourth of the face model is ignored to avoid noise from hair.

(2) *Component regions determination*: Within the bounding box of a facial model, we start to use four edge vertices as the initial centroids to cluster the edge vertices into four component regions, which are left eye, right eye, nose, and mouth. The initial centroids are determined simply by four edge vertices within the bounding box, which are *top-left*, *top-right*, *mid-bottom*, and *vertex close to nose tip*, respectively. The  $k$ -means clustering method ( $k=4$ ) is applied using Euclidean distances of edge vertices to the four centroids. The centroids of four regions are updated iteratively until they remain unchanged. As a result, four component regions are detected. Furthermore, the nose bridge region can be determined by the eye and nose boundaries, and the top of the convex hull. The complementary region of the five component regions within the face convex-hull forms the sixth component region. Figure 18(a) shows an example of the resulting segmentation.



**Figure 18. (a) Sample of component regions; (b) Sample of labeled surface of a sketch model, and (c) a component-based HMM based on six component regions.**



## 4.2 3D component feature representation

3D facial models of both scans and sketches can be characterized by their surface primitive features. This spatial feature can be classified by eight types: convex peak, convex cylinder, convex saddle, minimal surface, concave saddle, concave cylinder, concave pit, and planar. Such a local shape descriptor provides a robust facial surface representation [45][48]. To label the model surface, we select the vertices of the component regions, and then classify them into one of the primitive labels. The classification of surface vertices is based on the surface curvature computation [48]. After calculating the curvature values of each vertex, we use the categorization method [47] to label each vertex on the range model. As a result, each range model is represented by a group of labels, which construct a feature vector:  $G = (g_1, g_2, \dots, g_n)$ , where  $g_i$  represents one of the primitive shape labels,  $n$  equals the number of vertices in the component region. An example of the labeled surface is shown in Figure 18 (b).

Due to the high dimensionality of the feature vector  $G$ , where each of six component-regions contains vertices ranging from 300 to 700, we use a Linear Discriminant Analysis (LDA) based method to reduce the feature space of each region. The LDA transformation is to map the feature space into an optimal space that is easy to differentiate different subjects. Then, it will transform the  $n$ -dimensional feature  $G$  to the  $d$ -dimensional optimized feature  $O_G (d < n)$ .

### 4.3 Spatial HMM model classification

In each frame, the 3D facial model is subdivided into six components (sub-regions)  $C1$ ,  $C2$ ,  $C3$ ,  $C4$ ,  $C5$ , and  $C6$ , as shown in Figure 18 (c), including regions of the eyes, nose, nose bridge, mouth, and the remaining face. From  $C1$  to  $C6$ , we construct a 1-dimensional HMM which consists of the six states ( $N = 6$ ), corresponding to six regions. As aforementioned, we transform the labeled surface to the optimized feature space using LDA transformation. Given such an observation of each sub-region, we can train the HMM for each subject. Given a query sketch face model sequence of a length  $k$ , we compute the likelihood score for each frame, and use the Bayesian decision rule to decide which subject each frame is classified to. Since we obtain the  $k$  results for  $k$  frames, we take a majority voting strategy to make a final decision. As such, the query model sequence is recognized as subject  $Y$  if  $Y$  is the majority result among  $k$  frames. This method tracks spatial dynamics of 3D facial sketches, the spatial components of a face gives rise to the spatial HMM to infer the likelihood of each query model. Note that if  $k$  is equal to 1, the query sketch model sequence becomes a single sketch model for classification.

## 5. Experiments of Face Recognition

### 5.1 4DFE: 3D sketch(training) vs. 3D sketch(testing)

The 3D sketch models include 3D models created from both HD sketch images and MD sketch images. For each subject, we randomly select 50% of the model frames for training, the remaining 50% of the data for test. For subjects with HD models, we also include half of the data in the training set, and the rest are included in the test set.

For each training sequence of 4DFE, 20 sets of three consecutive frames were randomly chosen for training. Following the HMM training procedure ( $k=3$ ), we generated an HMM for each subject. The recognition procedure is then applied to classify the identity of each input sketch sequence ( $k=3$ ) as the previous section described. Based on the 10-fold cross validation approach, the correct recognition rate is about 95.5%. The ROC curve is shown in Figure 9.

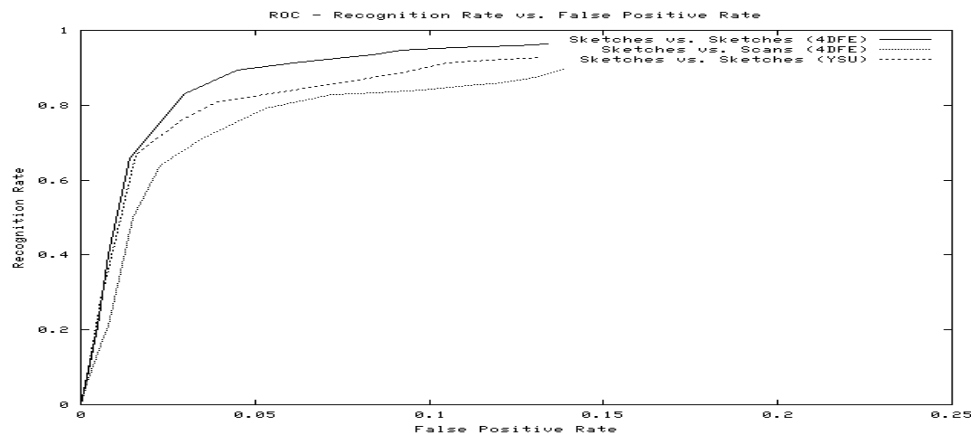
## **5.2 4DFE: 3D scans(training) vs. 3D sketches(testing)**

In order to validate the utility of the 3D sketches with respect to the 3D scans, we conducted the 3D sketch classification against the corresponding 3D scans. Similar to the above approach, for each subject, we randomly select 20 sets of three consecutive 3D scans for training. Following the HMM training procedure, we generated an HMM for each subject. The recognition procedure is then applied to classify the identity of each input 3D sketch sequence ( $k=3$ ). Based on the 10-fold cross validation approach, the correct recognition rate is about 89.4%. The ROC curve is shown in Figure 19.

## **5.3 YSU: 3D sketch(training) vs. 3D sketch(testing)**

The validation has also been conducted on the 3D sketch models created from YSU sketch database, where sketches from 50 subjects are created. Each subject has five sketches drawn by five artists separately. There are 250 sketches in total. For each subject, we randomly select 4 sketches for training, the remaining one for test. Following the HMM training procedure ( $k=1$ ), we generated an HMM for each subject. The recognition procedure is then applied to classify the identity of each input sketch model

( $k=1$ ). Based on the 10-fold cross validation approach, the correct recognition rate is about 92.6% (see ROC curve of Fig. 19).



**Figure 19. ROC curves of 3D sketch face recognition.**

Due to the sketches drawn from different artists in the YSU database, the variation of the sketch styles and the single model query plus single model training of HMM degrades the recognition performance as compared to the 4DFE case (sketches-to-sketches). However, the cross modality matching between 3D scans (training) and 3D sketches (testing) shows the challenge for classification as the 3D sketches created from 2D images may not match well to the ground true data (3D scans). A further study using a more advanced classifier will be investigated in future work.

## 6. Discussion

This chapter addressed the issues of 3D sketch modeling and its validation through 3D sketch recognition using a component based spatial HMM. The quality of 3D sketch models is evaluated by comparing to the corresponding ground-truth 3D scans. We have also shown the approximate equivalence of models between the 3D sketches from HD

and MD. Among the test data (4DFE and YSU databases), on average a 92% correct recognition rate has been achieved for 3D sketch model identification. While the results are promising, this would have more real world significance if it could be accomplished in a real-time setting. Looking into parallel algorithms for construction of the model is a future topic left to discussion.

## Chapter 5

### Facial Activity Analysis in 3D/4D Space

#### 1. Introduction

Facial activity analysis using 3D videos has become an intensified research topic in recent years [60][72][73][74]. 3D representation of real life objects allows for a more realistic behavior analysis and understanding. However, it is difficult to process the data in a 3D dynamic space. The major challenges lie in the difficulties of 3D data registration, 3D feature extraction, and 3D data description. In this chapter, we investigate approaches for effective 3D feature representations in order to characterize the dynamic geometric features across time for facial activity analysis.

Dynamic Texture (DT) is an effective method for appearance-based facial analysis from consecutive video-frames [75]. Some existing approaches to represent and extract dynamic textures were based on optical flow [79], motion history images [78], volume local binary patterns [77], and free form deformation [76]. Dynamic texture based methods have been successfully used for applications in facial expression recognition [77][78][79]. However, they are essentially 2D-based approaches with limitations of various imaging conditions (e.g., illuminations, poses, etc.).

Motivated by the dynamic texture approaches from 2D videos, we propose a new approach to describe the 3D facial activity in 3D videos, which is dynamic curvature in a

3D dynamic space for facial activity analysis. We segment the 3D facial meshes into several isolated local regions based on facial actions. Then, the histograms of shape-index from curvatures across multi-frame geometric surfaces are concatenated to form a unique descriptor - dynamic curvature for 3D facial behavior representation. Such a descriptor that represents the temporal dynamics of the facial surface will be input to a classifier (*e.g.*, SVM) for further classification of facial activities (*e.g.*, expressions, identities, etc.).

In order to segment the facial regions, it is critical to detect and track facial features across 3D geometric sequences. While research in 2D modality based tracking has produced a number of successful and widely used algorithms [57][80][81][56][58][52], research on 3D modality based analysis still faces the challenges of geometric landmark detection, mesh registration, motion tracking, and data representation. Traditionally, feature detection in 3D geometric space was performed by registration or 2D-to-3D mapping methods on static models [52][53][50][58][51][59][54][55]. In this chapter, we apply a tracking model constructed from a temporal 3D point distribution for this task.

We will evaluate the performance through an application for facial activity classification: neutral vs. non-neutral; six prototypic expressions; and the status of expression activity in low intensity vs. in high intensity.

The rest of the chapter is organized as follows: Section 2 provides a brief description of our tracking model. Section 3 describes dynamic curvature based 3D feature representation. Section 4 reports experiments and evaluations on the feature point

detection and dynamic curvature classification for facial activity recognition. Finally, discussion and conclusion are given in Section 5.

## 2. 3D Shape Tracking Model

3D range data exhibits shapes of facial surfaces explicitly. This shape representation provides a direct match with the 3D active shape model due to its inherent and explicit shape representation in 3D space. We present a 3D shape tracking model to describe the shape variation across the 3D sequences.

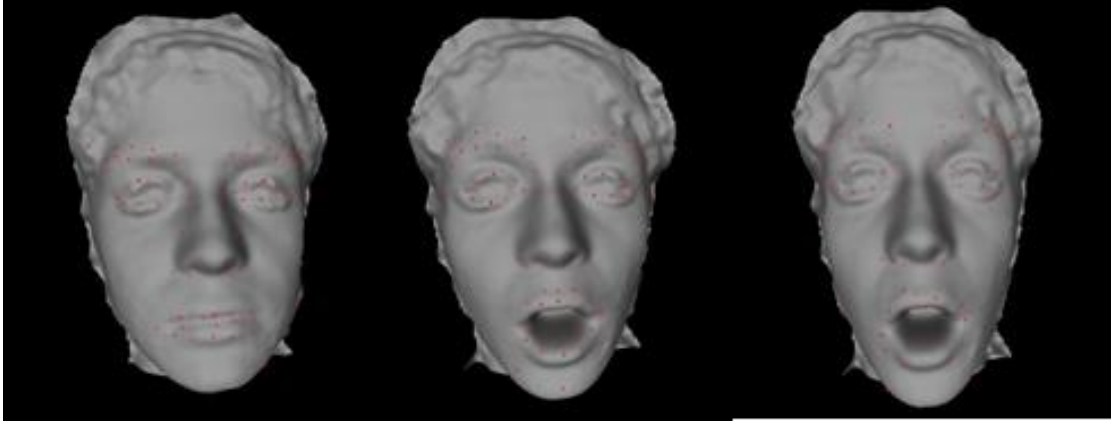
To construct a shape model, we apply a similar representation of the point distribution model to describe the 3D shape, in which a parameterized model  $S$  is constructed by 83 landmark points on each model frame. An example of landmark points is shown in Figure 1. Such a set of feature points (shape vector) is aligned by a Procrustes analysis method [56]. Then the principal component analysis (PCA) is then performed on the new aligned feature vector. This is done to estimate the different variations of all the training shape data. To do so, each shape deviation from the mean and the covariance matrix are calculated, resulting in the modes of variation,  $V$ , of the training shapes along the principal axes. Given  $V$  and a vector of weights,  $w$ , that controls the shape, we can approximate any shape from the training data by:

$$S = \bar{s} + Vw \quad (5)$$

The vector of weights,  $w$ , allows us to generate new samples by varying its parameters within certain limits.



When approximating a new shape  $S$ , the point distribution model is constrained by both the variations in shape and the shapes of neighbor frames. Figure 20 shows an example of the shape model and the tracked 83 feature points. The detailed algorithm is described in [82].



**Figure 20. Example of tracked 83 feature points on a surprise expression sequence.**

### **3. Dynamic Curvature Based Approach**

Given the detected facial features and the resulting local regions, the shape (curvature) change along the 3D model sequences can be observed in individual local regions. Inspired by the recent work on facial analysis from static curvature based approaches [51] and dynamic texture based approaches [76][77], we propose a so-called Dynamic Curvature based descriptor for facial activity classification. Visual texture of an object is the reflection of its physical surface and lighting reflectance. Physical surface of an object can be characterized by its surface descriptor, e.g., primitive curvature type, shape-index, normal, etc. Given this observation, we extend the concept of Dynamic Texture in 2D space to the concept of Dynamic Curvature in 3D space (Dynamic Shape-Index). Unlike dynamic texture based approaches, which require building a rotation/scale invariant

vector for feature representation, we use 3D shape descriptors (e.g., primitive curvature types, shape index) as our feature representation. Curvature is a good representation of local surface geometric characteristics. It is invariant to affine transformation like shift or rotation. Facial surface change reflects facial expression change. Encoding the surface changes of local facial region using dynamic curvature representation, we are able to capture the temporal dynamics of facial surface for expression classification.

After the model regions have been localized, the regional shape is described and quantified by curvature based shape-index. The dynamic curvature descriptor is then generated for classification.

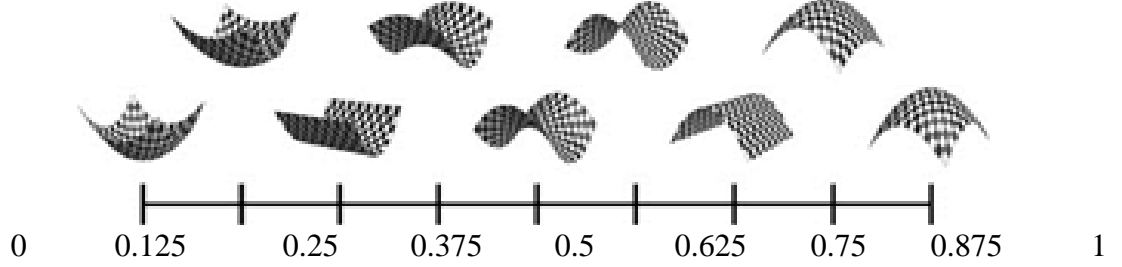
### 3.1 Shape description and quantization

Shape index is a quantitative measure of the shape of a surface at a point [60][61]. It gives a numerical value to a shape thus making it possible to mathematically compare shapes and categorize them. Shape Index is defined as follows:

$$S = \frac{2}{\pi} \times \arctan\left(\frac{k_2 + k_1}{k_2 - k_1}\right) \quad (6)$$

where  $k_1$  and  $k_2$  are the principal (minimum and maximum) curvatures of the surface, with  $k_2 \geq k_1$ . With this definition, all shapes can be mapped on the range  $[-1.0, 1.0]$ . Every distinct surface shape corresponds to a unique shape index value. The shape index is computed for each point on the model. We use a cubic polynomial fitting approach to compute the eigen-values of the Weingarten Matrix [60], resulting in the minimum and maximum curvatures  $(k_1, k_2)$ . The shape index scale is normalized to  $[0, 1]$ , and encoded as a continuous range of grey-level values between 1 and 255. To quantify the curvature

based measurement for an efficient description of a model, we transform the shape index scale to a set of nine quantization values from concave to convex, namely (1) Cup (0); (2) Trough (0.125); (3) Rut Saddle (0.25); (4) Rut (0.375); (5) Saddle (0.5); (6) Saddle Ridge (0.625); (7) Ridge (0.75); (8) Dome (0.875); and (9) Cap (1), as shown in Figure 21.



**Figure 21. Shape index quantization to nine values: Cup(0), Trough(0.125), Rut Saddle(0.25), Rut(0.375), Saddle(0.5), Ridge(0.625), Dome(0.875), and Cap(1).**

### 3.2 Dynamic Curvature Based Descriptor

Until this stage, each vertex on the 3D face model has been assigned a curvature-based label (*i.e.*, quantized shape index) based on the above shape analysis. Since each facial model is segmented into 8 sub-regions (e.g., eyes, nose, mouth, cheek, etc. as shown in Figure 22) from the set of tracked feature points, we are able to get the curvature distribution of each sub-region and combine them into a vector. To do so, we construct following histograms to form a dynamic curvature descriptor:

(1) *Regional Histogram of Intra-frame*: Given  $k$  facial frames and  $n$  regions for each individual frame, the histogram of shape-index of each region  $i$  of individual frame  $j$  is derived to form a histogram vector,  $h_i^j$ , where  $i=1, \dots, n; j=1, \dots, k$ ;

$$h_i^j = \left[ \frac{c_1}{c}, \frac{c_2}{c}, \dots, \frac{c_9}{c} \right] \quad (7)$$

where  $c$  is the total number vertices of a local region  $i$  in a single frame  $j$ , and  $c_1, \dots, c_9$  are the numbers of vertices with shape-index scale  $1, \dots, 9$  in that region, respectively.

(2) *Regional Histogram of Inter-frame*: In each region  $i$ , the statistics of shape-index is counted in all  $k$  frames as a whole to form a second histogram vector,  $\bar{h}_i^k$ , where  $i=1, \dots, n$ ;  $j=1, \dots, k$ .

$$\bar{h}_i^k = \left[ \frac{C_1}{C}, \frac{C_2}{C}, \dots, \frac{C_9}{C} \right] \quad (8)$$

where  $C$  is the total number vertices of a local region  $i$  across all  $k$  frames, and  $C_1, \dots, C_9$  are the numbers of vertices with shape-index scale  $1, \dots, 9$  in that region of all  $k$  frames, respectively.

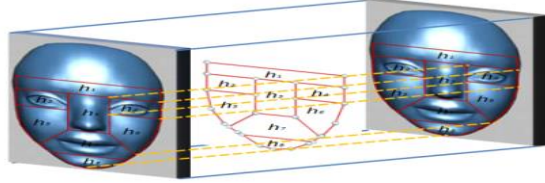
(3) *Local Temporal Histogram*: For each sub-region  $i$ , we concatenate the histogram  $h_i^j$  across  $k$  frames along the temporal axis and the histogram  $\bar{h}_i^k$  to formulate a local temporal histogram vector,

$$H_i^k = [h_i^1, h_i^2, \dots, h_i^k, \bar{h}_i^k] \quad (9)$$

(4) *Global Temporal Histogram - Dynamic Curvature Descriptor*: For the facial model across  $k$  frames, we combine all the local temporal histograms of  $n$  regions to generate a global descriptor (so-called dynamic curvature descriptor), which will be used for subsequent classification,

$$H_D^k = [H_1^k, H_2^k, \dots, H_n^k] \quad (10)$$

where  $n$  is the number of local regions and  $k$  is the number of frames ( $n=8$  and  $k=3$  in this implementation).



**Figure 22. Illustration of Dynamic Curvature descriptor based on eight local regions.**

### 3.3 Classification

After the dynamic curvature descriptor is created for 3D video sequences, we apply LDA for dimension reduction, and then use Support Vector Machine (SVM) classifiers to learn predictive power. Traditional SVM is used for binary classification. How to effectively extend it for multi-class classification problem is still an on-going research issue. One efficient way is to construct a multi-class classifier by combining several binary classifiers. The one-against-all SVM is constructed for each class by discriminating that class against the remaining  $M-1$  classes. The number of SVMs used in this approach is  $M$ . A test pattern  $x$  is classified by using the winner-takes-all decision strategy, i.e., the class with the maximum value of the discriminant function  $f(x)$  is the class that  $x$  belongs to.

Alternatively, the one-against-one SVM method is also known as one-versus-one method. An SVM is constructed for every pair of classes by training it to discriminate the two classes. Thus, the number of SVMs used in this approach is  $M(M-1)/2$ . A max-min strategy is used to determine the class that a test sample belongs to. That is to say, the class with the maximum number of votes for the test sample is assigned to the sample.

There are other existing multiclass SVM algorithms, e.g., directed acyclic graph SVM (DAGSVM) [62][63], Weston's multi-class SVM [64], and Crammer's multi-class SVM [65]. However, considering the algorithm complexity and classification performance, we chose the one-against-all SVM for the classification task.

## **4. Experiments and Evaluation**

### **4.1 Database**

A public database 4DFE [13] is used for our test. This is a 3D dynamic face model database, which contains 3D video sequences of six prototypic expressions of subjects. Each clip has neutral expressions and posed non-neutral expressions.

### **4.2 Facial Activity Classification**

Inspired by the 2D dynamic texture based approach which is capable of distinguishing different expressions, we extend the concept to dynamic curvature based approach for handling 3D dynamic range model videos. One of the advantages is that the curvature based descriptor encodes the local surface shape information explicitly, thus being relatively robust with noise and pose changes. To verify such a new descriptor, we performed experiments on facial activity on three levels. First, we distinguish the facial activity by expressive face (with non-neutral expressions) and non-expressive face (with neutral appearances). Second, given the expressive face category, we apply the SVM (one-against-all) to classify the six prototypic expressions. Third, we further identify the intensity of each prototypic expression: either low intensity or high intensity.

We used 60 subjects from 4DFE for our experiment. The experiment is subject-independent. We randomly choose 50 subjects for training and 10 subjects for testing. Based on the tenfold cross-validation approach, by which the tests are executed 10 times with different partitions to achieve a stable generalization recognition rate. The classifier used for all three-level experiments is the two-class SVM. Followings are the results for three-level facial activity classification.

***First Level: Neutral vs. Non-Neutral.***

The confusion matrix is listed as below in Table 4. The average recognition rate to separate neutral with non-neutral expression is as high as 94.7%.

**Table 4. Recognition rate for neutral/non-neutral expression.**

| True\Estimate | Neutr | Non-Neutr |
|---------------|-------|-----------|
| Neutral       | 95.1% | 4.9%      |
| Non-Neutral   | 5.7%  | 94.3%     |

***Second Level: Six prototypic expressions***

From the non-neutral group of video segments, we further classify six prototypic expressions: anger, disgust, sadness, happiness, fear, and surprise. The confusion matrix of distinguishing six universal expressions is listed in Table 5. The average recognition rate is 84.8%

**Table 5. Recognition rate for six universal expressions(%).**

| True\Estimate | Anger | Disgust | Fear | Happy | Sad | Surprise |
|---------------|-------|---------|------|-------|-----|----------|
| Anger         | 83.6  | 5.5     | 4.9  | 0     | 3.8 | 2.2      |
| Disgust       | 5.1   | 83.2    | 5.8  | 0     | 3.3 | 2.6      |
| Fear          | 1.7   | 3.2     | 81.3 | 7.5   | 4.2 | 2.1      |
| Happy         | 1.1   | 2.1     | 0    | 92.1  | 0   | 4.7      |
| Sad           | 4.2   | 8.6     | 9.2  | 0     | 78  | 0        |
| Surprise      | 1.1   | 1.9     | 3.6  | 3.9   | 0   | 89.5     |

***Third Level: Low Intensity vs. High Intensity***

For each recognized expression, their corresponding 3D video segments are further classified by the binary SVM for separating their degree of the expression: low intensity or high intensity. Below are the summary of the average rate (Table 6) and the individual confusion matrix (Table 7).

**Table 6. Average separation rate of low/high intensities.**

| Angry | Disgust | Fear  | Happy | Sad   | Surprise |
|-------|---------|-------|-------|-------|----------|
| 80.6% | 83.4%   | 79.1% | 91.2% | 78.4% | 90.7%    |

**Table 7. Confusion matrix of individual expressions for intensity(low/high) separation.**

| Expression      | True\Estimate | Low   | High  |
|-----------------|---------------|-------|-------|
| <i>Angry</i>    | Low           | 81.8% | 18.2% |
|                 | High          | 20.6% | 79.4% |
| <i>Disgust</i>  | Low           | 81%   | 19%   |
|                 | High          | 14.2% | 85.8% |
| <i>Fear</i>     | Low           | 80.1% | 19.9% |
|                 | High          | 21.9% | 78.1% |
| <i>Happy</i>    | Low           | 86.1% | 13.9% |
|                 | High          | 3.7%  | 96.3% |
| <i>Sad</i>      | Low           | 79.4% | 20.6% |
|                 | High          | 23.6% | 77.4% |
| <i>Surprise</i> | Low           | 85.5% | 14.5% |
|                 | High          | 4.1%  | 95.9% |

### 4.3 Comparison

We also conducted experiments with both our dynamic curvature based approach and other methods for recognizing expressions with both high and low intensities, respectively. We choose the recent and classic work for comparison, including 3D dynamic HMM [59][68], 3D dynamic Motion Units [68], 3D static surface primitive feature distribution [51], 2D dynamic motion units [67], 2D dynamic texture [77], and 2D static Gabor Wavelet [66]. As shown in the Table 8, the dynamic curvature based



approach outperforms other approaches in both cases of low intensity and high intensity of expressions. Its performance is close to the 3D dynamic HMM based approach where spatial-temporal features were described in the HMM framework.

**Table 8. Recognition rate from low intensity (LI) expressions and high intensity(HI) expressions using different approaches respectively.**

| Methods                                    | Low (LI) | High (HI) |
|--|----------|-----------|
| <i>3D dynamic curvature (our approach)</i> | 75.1%    | 86.3%     |
| 3D dynamic (HMM) [59][68]                  | 72.4%    | 83.7%     |
| 3D dynamic (MU based) [68]                 | 57.3%    | 72.1%     |
| 3D static (PSFD) [51]                      | 52.8%    | 71.7%     |
| 2D dynamic (MU based) [67]                 | 56.6%    | 69.2%     |
| 2D dynamic (DT based) [77]                 | 70.8%    | 81.5%     |
| 2D static (Gabor) [66]                     | 50.4%    | 68.6%     |

## 5. Discussion

This chapter presented a new 3D feature representation using a so-called dynamic curvature based approach for facial activity analysis. The experiments have shown the feasibility of such a new descriptor for 3D facial activity analysis. We have evaluated its utility for dynamic curvature based expression classification in terms of neutral vs. non-neutral, various prototypic expressions, and their high/low intensities. This type of method lends itself well to a parallel architecture. The descriptors for each of the regions can be constructed in parallel allowing for a real-time scenario for facial activity analysis.

## **Chapter 6**

### **3D/4D Feature Detection Using**

### **Action-based Statistical Shape Models**

#### **1. Introduction**

Landmark localization on 3D range data is the first step toward geometric based vision research for object modeling, recognition, visualization, and scene understanding. Applications in this area of research include 3D face recognition and expression interpretation for biometrics and human computer interaction [88] and face segmentation [89]. With the rapid and affordable [86] development of 3D imaging technologies, 3D range data is becoming one of the most popular modalities for applications in computer vision. While research in 2D modality based tracking has produced a number of successful and widely used algorithms, such as Active Shape Model [57] and Local Binary Pattern [52], research in 3D modality based analysis still faces the challenges of 3D geometric landmark localization, 3D mesh registration, and 3D motion tracking. Therefore, there is a strong demand for novel and robust algorithms for handling 3D datasets. Morphable Model [58] is a successful algorithm for these 3D problems. Another commonly used method for registering two meshes is the Iterative Closest Point algorithm (ICP) [53]. This method relies on finding the closest pairs of points between the two meshes being registered; however, it shows limitations in handling largely deformed mesh models. X. Lu et al. [59] developed an approach using ICP to detect the nose tip and mouth corner landmarks to help register the meshes for classification. Wang

et al. [47] used key facial landmarks selected semi-automatically to segment the face and perform facial expression analysis by evaluating the principal curvatures in those segmented regions.

Active shape models (ASM) have been widely used to address the problem of landmark detection and tracking, although mainly on 2D data [57][90] or volumetric data [91] for medical data segmentation. The construction and tracking of a 2D-based ASM relies on both 2D shape components and 2D texture components due to the lack of explicit 3D shape representation of texture data. The fitting process relies on a regression procedure guided by shape constraints and texture primitive constraints (e.g. edge, intensity, and color, etc.) The quality of the results, however, is limited by the accuracy of these constraints and the degree of pose variance.

Recent work has addressed the problem of fitting a deformable model to 3D range data. Fanelli et al. [92] used a random forests-based active appearance model for face alignment. While they have achieved good results with the tested data, their method currently does not handle noisy data from cameras such as the Microsoft Kinect [86]. Sun et al. [48] used active appearance models (AAM) to track features of 3D range models. However, the detection and tracking of facial features were performed on 2D videos, while the 3D features themselves were obtained by mapping the 2D features to the corresponding parts of the 3D models. Nair et al. [55] developed an approach to fit an active shape model to 3D face meshes using candidate landmarks for the inner eye corners and nose tip. Their active shape model is fit by finding a similarity transformation

between the candidate landmarks of the mesh and the corresponding landmarks within their active shape model. Perakis et al. [56][149] obtained candidate landmarks through shape index calculation, and compared them to their 3D active shape model. However, there is no single fitting or temporal fitting process for finding candidate landmarks. Zhao et al. [93] used a patch based method to probabilistically fit a statistical facial feature model to 3D range data. However, their method has a noise removal preprocessing step where spikes are detected and removed and holes are filled. Guan et. al [94] used a Bezier surface for landmark localization on 3D data. Weise et. al [95] used a statistical model to track a face and animate a virtual avatar. In all of the above approaches, the method was restricted to only face models, no temporal information was utilized, or only non-noisy data was tested on.

In this chapter, we extend our previous work [96] by proposing a method to construct action-based statistical shape models (ASSM) for landmark localization on both 3D and 4D (3D + time) range data. An ASSM can be constructed from either 3D or 4D point distribution models, without the use of textures. Our action-based models are built from different actions, and are not limited to only face models. These actions can be, but are not limited to, expressions of a face (e.g. fear, anger, happiness, sadness, surprise, and disgust), movement of an arm (e.g. bent or straight), or rotations. The basic method for model fitting relies on finding the closest points in the range mesh model that correspond to an instance of the ASSM, where an instance is defined as a sampling along the modes of variation in the ASSM. The variance of the ASSM weighted matrix determines whether a set of landmarks is considered an acceptable candidate for a good fit. For each

adaptation from the ASSM to the range mesh surface, we compute a distance score between the newly detected landmarks and the original instance of the ASSM. The lowest score is considered the best fit to the range mesh model.

Our primary contribution is the use of action-based statistical shape models for landmark localization on both static and dynamic (temporal) range data that can be noisy and/or have incomplete or missing data. Using the ASSM method we are able to fit various input modalities for range data such as faces, arms, hands, and toy models. Our ASSM method not only makes use of the local constraints imposed by statistical models, but also the inter-frame constraints imposed when modeling 4D data. This is due to the nature of each different type of action. For example a face expression can be assumed to behave in the following way: neutral, to onset, to peak, to offset, and back to neutral again. For rotation data, the sequence would have the following behavior: frontal view, partial profile view, full profile view. We have found that a small amount of modeled actions can be used for landmark localization on a variety of sequences (e.g. rotation from frontal to full profile). We apply the detected landmarks to both subject identification and expression classification on multiple public databases. We also evaluate the accuracy of the landmark detection through applications of 3D video segmentation, gesture recognition, and pose estimation.

## **2. 3D Action-Based Statistical Shape Model**

3D range data exhibits shapes of surfaces explicitly. This shape representation provides a direct match with our action-based statistical shape model due to its inherent and explicit

shape representation in 3D space. In considering this property, our landmark localization algorithm can rely solely on 3D geometric shape without assistance of any texture information, thus resulting in less sensitivity to pose and lighting variations.

To take advantage of this property, we would like to model the shape variation, as well as the implicit shape (“action”) constraints imposed between consecutive frames in a sequence of models. Given a training set of  $M$  mesh models each with  $N$  annotated landmarks, the data is separated into  $L$  groups consisting of the actions to be modeled (e.g. for a facial expression  $L=3$  for neutral, onset, and peak). To construct an action-based temporal point distribution model, a parameterized model,  $S$ , is constructed where  $S = P_1^1, \dots, P_N^1, P_1^2, \dots, P_N^2, \dots, P_1^k, \dots, P_N^k$ .  $P_i^k$  is the  $i^{th}$  landmark of the  $k^{th}$  model, where  $P_i^k = (x_i^k, y_i^k, z_i^k)$  and  $1 \leq k \leq M$  ( $M$  is the total number of training models). To construct this model, the training landmarks must be aligned. To do so a modified version of Procrustes analysis is used [57].

Procrustes analysis determines a linear transformation that aligns two sets of points (shapes). It minimizes the distance between the sets of points, which is a minimized summation of the squared errors. Once alignment has been performed, principal component analysis (PCA) is then performed on the aligned feature vector. This estimates the different variations of all the training data in the  $k \times N \times 3$  dimensional space. For PCA, each shape deviation from the mean is calculated as

$$ds_i = s_i - \bar{s} \quad (11)$$

Where  $s_i$  is the current shape and  $\bar{s}$  is the average shape. The covariance matrix  $C$  is then calculated:

$$C = \frac{1}{M} \sum_{i=1}^M ds_i ds_i^T \quad (12)$$

This equation yields the modes of variation,  $V$ , of the training shapes along the principal axes. Given  $V$  and a weight vector,  $w$ , that controls the shape, we can approximate any shape from the training data by:

$$S = \bar{s} + Vw \quad (13)$$

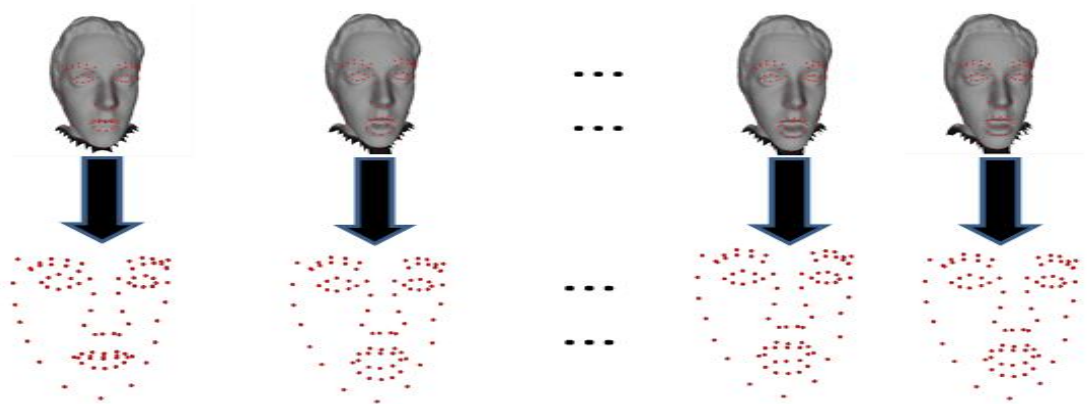
The weight vector,  $w$ , allows us to generate new samples by varying its parameters within certain limits. These limits are imposed to ensure only valid shapes are constructed (i.e. a correct facial expression). In the literature, most statistical shape models constrain the allowable shapes to be within 3 standard deviations from the mean, however, for our ASSM method we have empirically found that we can constrain the allowable shape domain to be within 2 standard deviations from the mean, giving us:

$$-2\sqrt{\lambda_i} \leq w_i \leq 2\sqrt{\lambda_i} \quad (14)$$

where  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of  $C$ . Constraining the allowable shapes domain to be within 2 standard deviations from the mean allows our ASSM method to have a smaller search space for each modeled action, which is detailed in section 3.

When approximating a new shape  $S$ , the action-based temporal point distribution model is constrained not only by the variations in shape but also by the inter-frame constraints that consecutive temporal frames impose. Given a  $k$ -frame ASSM,  $k$  consecutive input mesh models are ensured to vary in a manner that is consistent with the ASSM. For

example, we can assume that during the course of an expression, facial appearance is developed gradually. If we have a frame displaying a neutral expression at the start of the sequence, the next frame cannot display the peak of the expression, as there needs to be some form of the onset of the expression before the peak occurs. The feature vector will not allow the shape to have a neutral expression next to the peak. Therefore, during the adaptation of the ASSM, if we come across  $k$  mesh models that do not vary in a way that is consistent with our ASSM, we can attribute this to an unknown anomaly and label the  $k$  mesh models as such. Fig. 23 shows an example of a  $k$ -frame ASSM, modeling a surprise face expression, where  $N=83$ .



**Fig. 23. Example  $k$ -frame ASSM where  $N=83$ . Top row shows fit mesh models, bottom row shows visual representation of ASSM face expression vector.**

### **3. Fitting 3D and 4D Range Data**

#### **3.1 Fitting 3D Range Data Using an ASSM**

When dealing with static 3D range data we construct an ASSM where  $k = 1$ , allowing us to fit a single frame in the absence of a sequence of frames. To fit the ASSM to 3D range data, the optimal weight parameters are learned off-line by uniformly perturbing each



instance of the ASSM within the allowable shape domain of 2 standard deviations from the mean. This optimal weight vector,  $w$ , will control the allowable shapes of the ASSM. This off-line learning allows us to speed up the fitting process, as well as have more control over which shapes are constructed and to help ensure the new shapes are consistent within the allowable shape domain.

Once we have our optimal weight vector, the instances of the ASSM are then fit, without any initialization or a priori knowledge of the action class (rotation, expression, etc.), to the 3D input data. This is done by finding which vertex in the range mesh model corresponds to the closest point of each landmark in the ASSM instance. *We are able to do a simple closest point search as the final Procrustes distance will be large if desirable points have not been found.* When searching the input range model for the closest points, each model is constructed as a k-d tree, which helps to significantly speed up computation time from that of a brute force search. After we find the  $N$  closest points in the model, we then determine if the newly detected landmarks for the ASSM instance correspond to an allowable shape based on the constraint that the weight vector,  $w$ , must fall within 2 standard deviations from the mean. To do this, we must transform our detected landmarks into the model parameter space by constructing a new  $w$  vector. Since equation (13) gives us  $S = \bar{s} + Vw$ , we can then find the corresponding  $w$  vector of the detected landmarks by the following:

$$w = V^T(S - \bar{s}) \quad (15)$$

We then compare this new  $w$  vector to the allowable domain. If it is within this range, it is accepted as a candidate best fit for the 3D range data, otherwise it is discarded for the

range model we are trying to fit. For the candidates that are accepted, the ASSM instance that yielded these candidate landmarks, as well as the candidate landmarks themselves, are saved. Each candidate model then has a distance score computed between the newly detected landmarks and the ASSM instance. This distance score is the Procrustes distance, a metric used to determine the shape difference between two objects. Given the original instance of our ASSM  $m_1 = (x, y, z)$  and the detected landmarks on the range data  $m_2 = (u, v, w)$ , the Procrustes distance can be defined as:

$$D = \sum_{i=1}^N \sqrt{(u_i - x_i)^2 + (v_i - y_i)^2 + (w_i - z_i)^2}. \quad (16)$$

We find the Procrustes distance for each ASSM and its corresponding candidate landmarks on the range data for all candidate models. The smallest  $D$  value is considered the best fit. The smallest  $D$  is quickly found due to our offline vector  $w$ , as the computation is linear in terms of the number of landmarks as (16) shows. It is important to note that we have tested our approach on more than 80,000 3D/4D range models on four publicly available face databases, as well as in-house data collected from the Microsoft Kinect [19], and our range scanner. Through this testing we have empirically found our Procrustes distance-based approach to consistently show small distances when a good fit is found. Table 9 summarizes our ASSM algorithm below.

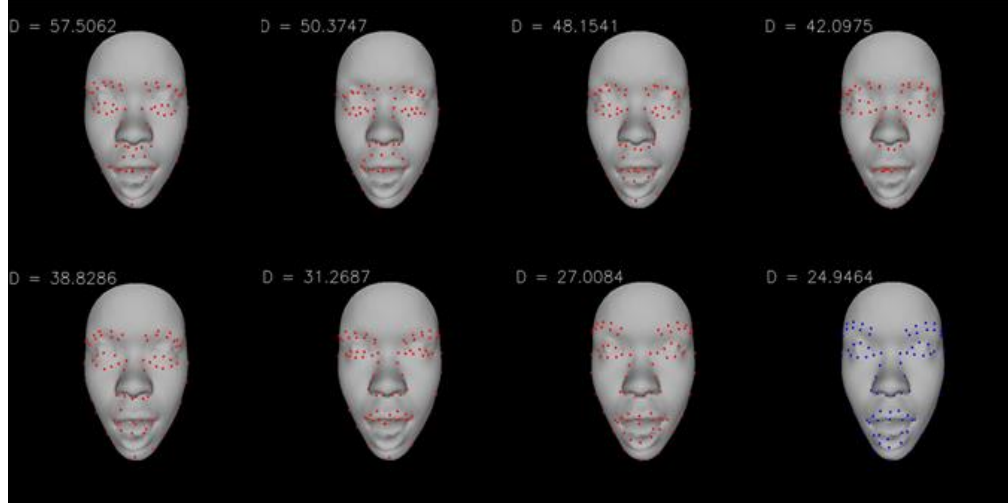
**Table 9. ASSM Fitting Algorithm.**

---

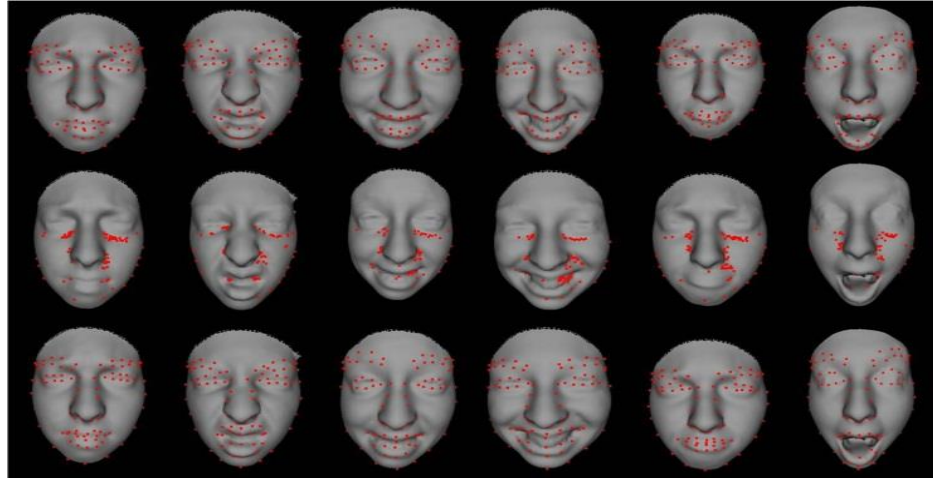
| <b>ASSM LANDMARK LOCALIZATION</b>                         |  |
|---|--|
| <hr/>   |  |
| <b>Input:</b> $k$ 3D mesh models                          |  |
| 1.  | Learn optimal weight parameters, off-line, to construct allowable instances of ASSM.     |
| 2.  | Construct $k$ k-d trees from input mesh models, to speed up computation time.            |
| 3.  | Search $k$ k-d trees for closest landmarks to each ASSM instance.                        |
| 4.  | Transform detected landmarks from step 3 into the model parameter space.                 |
| 5.  | Determine acceptable candidate fits from model parameter space.                          |
| 6.  | Find smallest distance, $D$ , for each ASSM and the candidate fits determined in step 5. |
| 7.  | Select detected landmarks that give smallest $D$ as best fit.                            |
| <b>Output:</b> Localized landmarks on $k$ 3D mesh models. |  |

---

Fig. 24 shows sample frames from the fitting process, where the smallest  $D$  value is selected as the best fit. Fig. 25 shows examples from the BU-3DFE database [31]. Included in this figure are the best and worst fits of the detected landmarks with comparison to the manually selected ground truth. For this example, the distance measure is very high for the worst fit, detailing a worst case scenario for the ASSM fitting process.



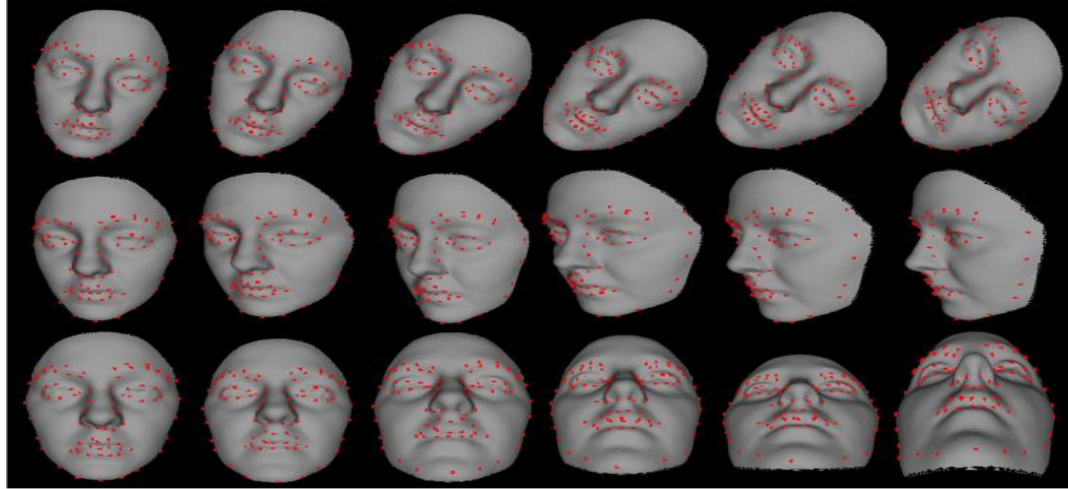
**Figure 24. Sample frames from fitting process ( $k=1$ ). Higher  $D$  values show poor fits, lowest  $D$  selected as best fit (in blue).**



**Figure 25. Top row: best fit, Middle row: worst fit, bottom row: ground truth.**

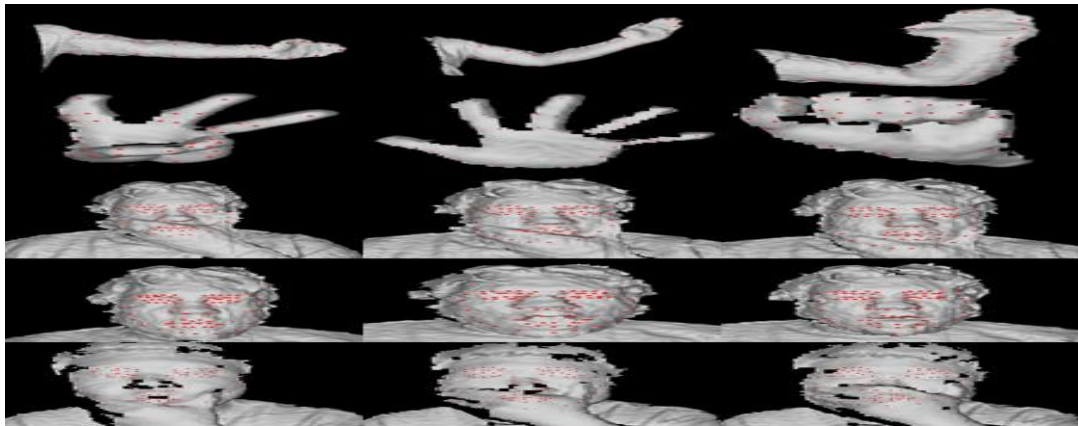
Fig. 26 shows landmark detection on models that were manually rotated to display roll, pitch, and yaw, illustrating robustness to pose variation. Fig. 5 shows example 3D mesh models captured from the Microsoft Kinect [86], including non-face objects, and partially occluded faces. As seen in Fig. 27, due to the low resolution of the Kinect scanner, there is missing data in the models including breaks in the fingers from the hand. Our ASSM algorithm can still detect the landmarks on this type of data, without the need for any pre-

processing. Fig. 6 shows sample frames captured from our in-house range scanner where the subject's face is partially occluded.

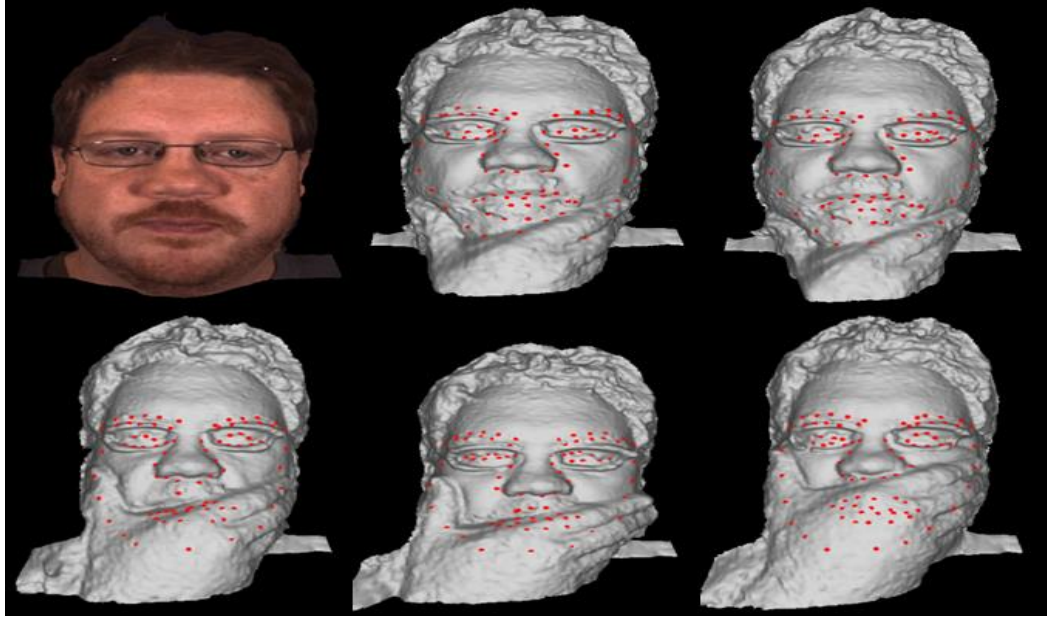


**Figure 26. ASSM fit on models displaying roll, yaw, and pitch.**

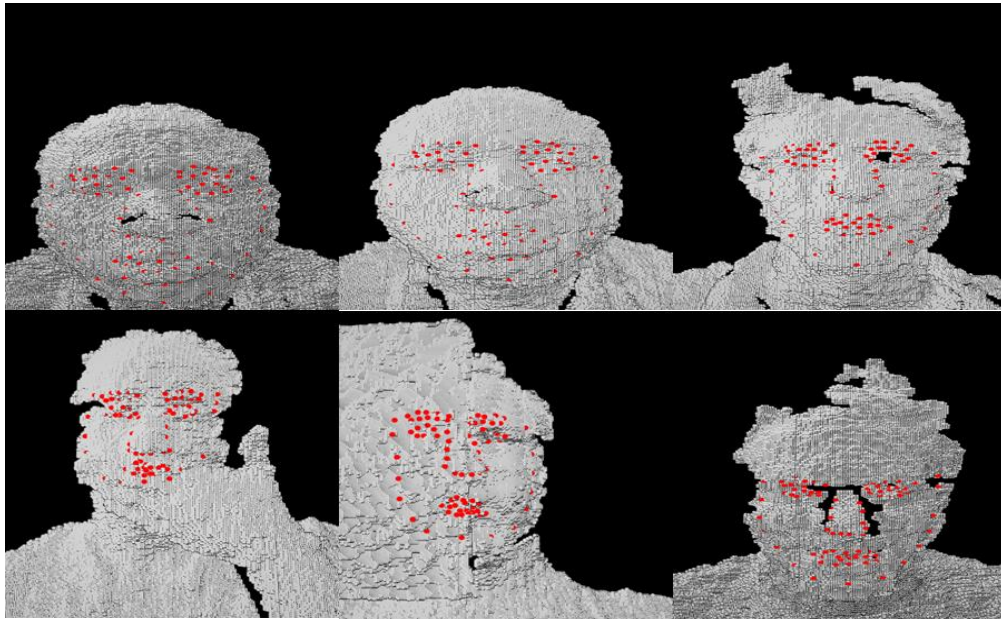
It can be seen in Fig. 27 and Fig. 28 that our ASSM method can accurately detect landmarks on face and non-face data, as well as display robustness to occlusion and incomplete data. In addition we also tested on the Eurecom Kinect Face Dataset [24], showing robustness to occlusion, noisy and missing data, as can be seen in Fig. 29.



**Figure 27. Microsoft Kinect [86] showing face data with partial occlusions and incomplete non-face data.**



**Figure 28.** Frames from our in-house range scanner showing occlusion of subject's face. *NOTE: the first frame shows texture for display purposes only, showing robustness to eye-glasses and facial hair.*



**Figure 29.** Sample frames fit with ASSM algorithm from the Eurecom Kinect Face Dataset [87], showing robustness to occlusion, noisy, and missing data.

### 3.2 Fitting 3D Range Data Using an ASSM

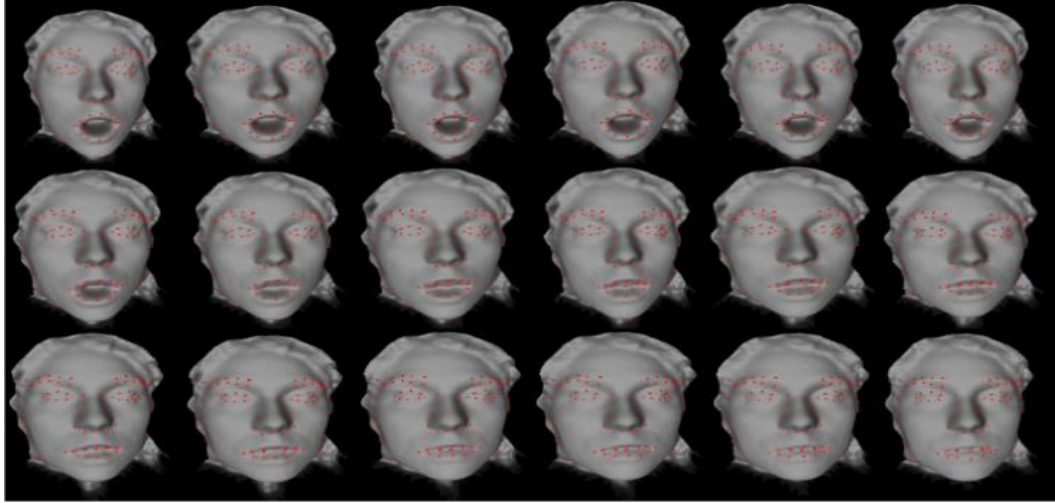
Given an input sequence of  $M$  frames, we can detect landmarks using an ASSM where  $k > 1$ , so-called multi-frame ASSM, by imposing inter-frame constraints on the fitting process. Similar to the algorithm in Table 1, we extend the  $w$  vector to the length  $k \times N$ , where  $k$  is the number of mesh models and  $N$  is the number of landmarks. Again, the reach of the  $k$  mesh models to be fit is represented as a k-d tree. The search is once again for the closest points on each of the mesh models. However, instead of searching for all  $k \times N$  landmarks in the ASSM for  $k$  mesh models, the ASSM is still searched using  $N$  landmarks for each individual model. Then, the k-frame ASSM is applied using the same criteria as in Algorithm 1 in Table 1. Since an action appears over certain durations, we can define a k-frame ASSM based on the samples of these durations. To give examples of how this method works, we will detail both expressions and rotations, however, *it should be noted that this method extends to other actions, as well as, extended durations*. While performing an expression, the following five durations would be displayed: neutral, onset, peak, offset, and neutral. Performing a rotation would display the following three durations: frontal, partial rotation, and full rotation. For our implementation when modeling rotations, we have defined frontal as 0-20 degrees, partial rotation as 20-50 degrees, and full rotation as 50-90 degrees. This can be done for both positive and negative rotations of roll, pitch, and yaw.

In our implementation we define a multi-frame ASSM with  $k = 2$ . Given an expression with durations from neutral, to onset, to peak, to offset, and back to neutral, we construct 8 ASSM for each expression, combining two frames in different durations. The 8 ASSM

where  $S_i = \{S_1, \dots, S_8\}$  are neutral to neutral, neutral to onset, onset to onset, onset to peak, peak to peak, peak to offset, offset to offset, and offset to neutral. Note that such an inter-frame relationship (or temporal constraint) makes the landmark detection across multiple frames occur simultaneously and accurately. Given a sample rotation from frontal to full profile, with durations from frontal, to partial, to full, back to partial, and finishing again at frontal, we construct 7 ASSM for the rotation sequence. These 7 ASSM where  $S_i = \{S_1, \dots, S_7\}$  are frontal to frontal, frontal to partial, partial to partial, partial to full, full to full, full to partial, and partial to frontal. This can again be applied to both positive and negative rotations for roll, pitch, and yaw. Such a relationship is applicable to any action with any speed since the multi-frame ASSM can handle variable speed actions. The temporal constraint can filter out some impossible cases (e.g. neutral-peak, frontal-full, etc.) thus resulting in a consistent fitting. Any violation of the inter-frame relationship will cause a large fitting error.

Using the 8 ASSM for the expressions, we fit each one to frames in multiple expression sequences to find the best fit. A sample surprise expression sequence from the BU-4DFE database [13] can be seen in Fig. 30. Shown in Fig. 31 are sample frames from the BU-4DFE [13] and BP4D-Spontaneous [85] in the second and bottom rows respectively. Fig. 31 also shows the examples of FRGC 2.0 [84] (top row), and BU-3DFE database [31] (third row).



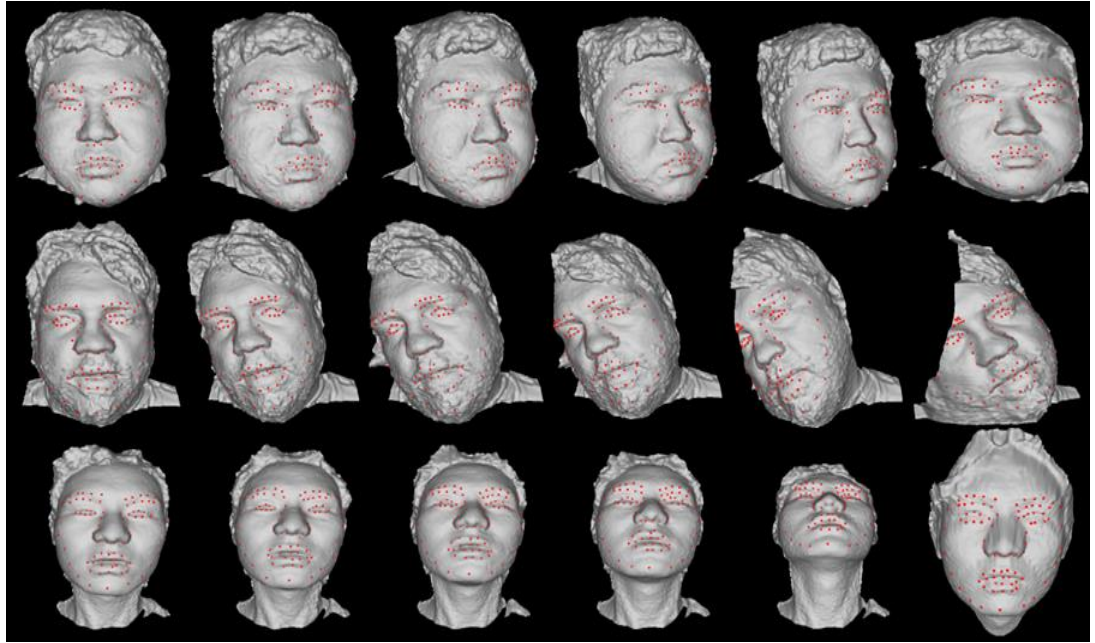


**Figure 30. 4D Sequence fit with ASSM method.**

Fig. 4 shows how our ASSM method can successfully detect landmarks on 3D rotated models, however, the data in this figure has been manually rotated for each roll, pitch, and yaw pose. While this shows robustness to pose variations, this is not an accurate representation of how rotations would occur in a real-world scenario. Given rotations of roll, pitch, and yaw, the models could exhibit large deformations in the mesh, including self-occlusions (i.e. completely missing data). Using the 7 ASSM for rotations, we are also able to model these deformations in the mesh. Example frames displaying roll, pitch, and yaw can be seen in Fig. 32, as well as Fig. 41.

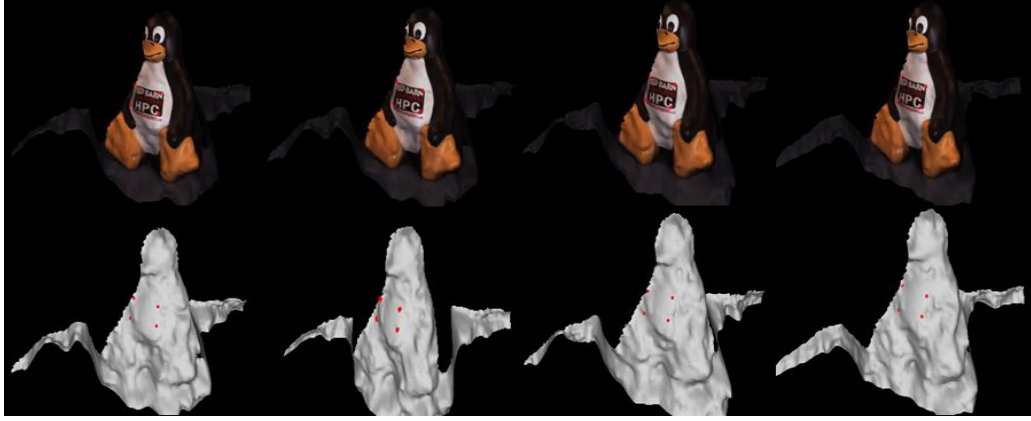


**Figure 31.** By row: FRGC 2.0, BU-4DFE, BU-3DFE, and BP4D-Spontaneous.



**Figure 32.** Sample frames from our in-house scanner showing yaw, roll, and pitch pose variations. *Note: The last column in each row is the same model from the previous column. The view has been changed to show the mesh deformations that this degree of pose shows.*

As shown in Fig. 5, our method can also model non-face objects. Fig. 33 shows another example of fitting an ASSM to non-face data, as well as the ASSM being able to fit mesh models with no real discernible features (e.g. the label on the front of the penguin is not featured on the mesh model).



**Figure 33.** Sample frames from our in-house scanner showing non-face data in the form of a rotated toy penguin. *Note: the texture is shown for display purposes only, to give a better visual representation of which features are selected.*

## 4. Experiments and Evaluation

### 4.1 Databases

Four public face databases have been used for our study including two static and two dynamic databases. BU-3DFE [31] consists of 100 subjects each displaying one neutral expression and four intensity levels of six expressions. FRGC 2.0 [84] consists of 466 subjects displaying two different expressions. BU-4DFE [13] consists of 101 subjects with sequences of six different expressions. BP4D-Spontaneous database [85] consists of 41 subjects (56% female and 44% male), each consisting of 10 different spontaneous expression sequences. The expressions are elicited activities including film watching, interviews, and experiencing cold pressor test among others. Ten different spontaneous expressions are evoked (joy, embarrassment, surprise, disgust, anxiety, fear, sadness, pain, anguish, sympathy). Each task could have multiple spontaneous expressions or mixed emotions due to the nature of the experimental setup. The database includes the 3D dynamic model sequences, texture videos, and annotated action units (AU). Fig. 31

(bottom row) shows an example of the database (details are described in [85]). Table 10 lists more details pertaining to each database.

**Table 10. Summary of databases.**

| <b>DATABASE SUMMARIES</b> |                 |             |                           |                              |                |
|---------------------------|-----------------|-------------|---------------------------|------------------------------|----------------|
| <b>Database</b>           | <b>Modality</b> | <b>Type</b> | <b>Number of Subjects</b> | <b>Number of Expressions</b> | <b>#Models</b> |
| 3DFE                      | Static          | Deliberate  | 100                       | 7                            | 2500           |
| 4DFE                      | Dynamic         | Deliberate  | 101                       | 6                            | 606 Sequences  |
| FRGC 2.0                  | Static          | Deliberate  | 466                       | 2                            | 932            |
| BP4D-Spontaneous          | Dynamic         | Spon.       | 40                        | 10                           | 240 Sequences  |

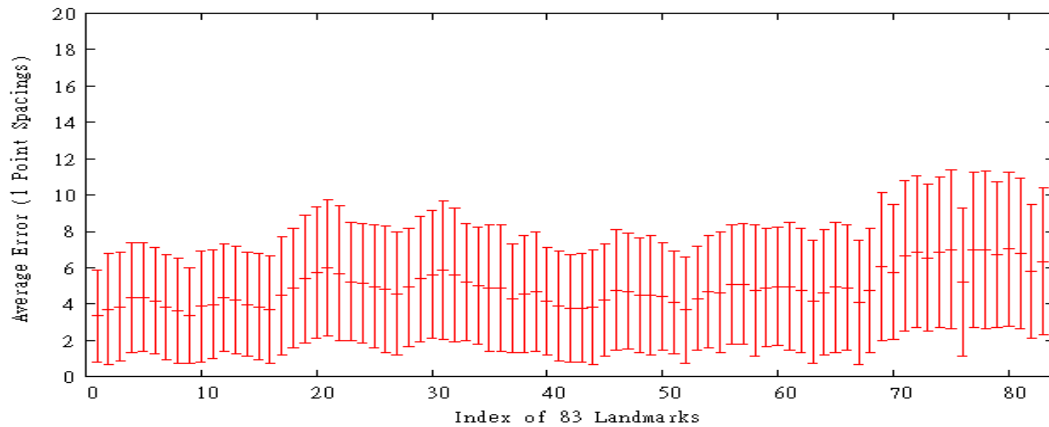
## 4.2 Database Error Rates

To construct our ASSMs, approximately 5-10% of the data was used for training and the rest was used for testing. To evaluate the accuracy of the ASSM fitting algorithm, we calculate the error between the fit landmarks and manually selected ground truth. To do so, we calculate the mean square error (MSE) between the two sets of landmarks. We define the one-point spacing as the closest pair of points on the 3D scans (0.5 mm on the geometric surface). If we treat the unit error being equivalent to 1 point-spacing, the mean error can be computed by the average of point differences between the two sets. The average errors on the four databases are listed in Table 11. As can be seen from Table 3, the algorithm can be used for multiple databases. The databases tested each had different quality and resolution of data, however, the average fitting errors are each within a consistently small range, showing our proposed algorithm has robustness to different data.

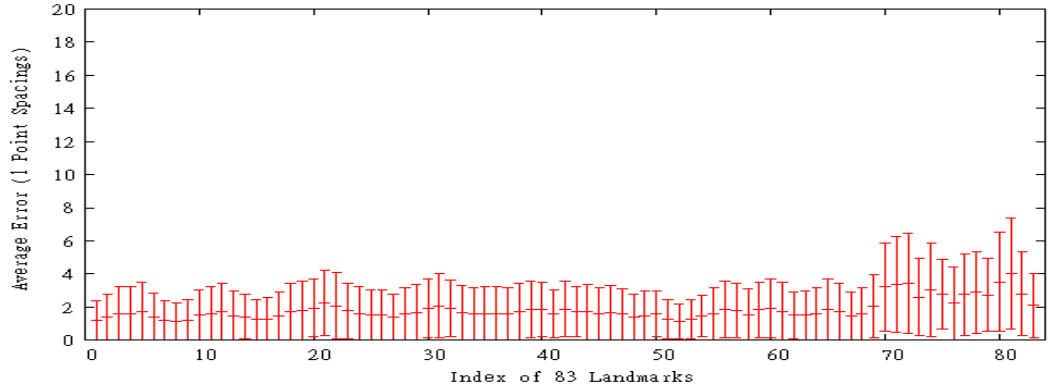
**Table 11. Average Error in point spacings for different databases and resolution in number of vertices per model.**

| AVERGAE ERROR IN POINT SPACINGS           |            |            |          |                     |
|---|------------|------------|----------|---------------------|
| Database                                  | BU<br>3DFE | BU<br>4DFE | FRGC 2.0 | BP4D<br>Spontaneous |
| Average Error                             | 5.6        | 1.5        | 6.7      | 1.6                 |
| Approximate Resolution<br>(# of Vertices) | 20000      | 30000      | 100000   | 50000               |

Fig. 34(a) shows the error statistics (average error and standard deviation for each of 83 key points) of the BU-3DFE database. Fig. 34(b) shows the error statistics for the BU-4DFE database. Fig. 34(b) shows lower error rates over each of the 83 detected landmarks compared to that of Fig. 34(a).

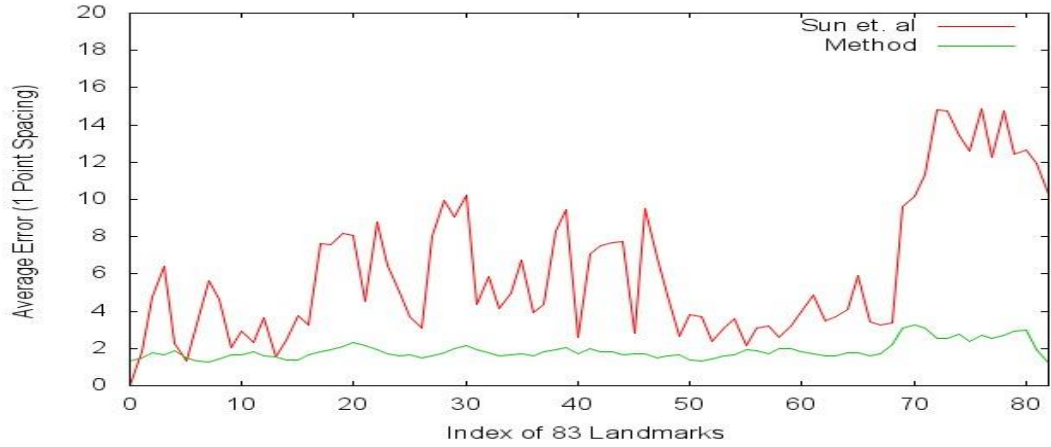


**Figure 34(a). BU-3DFE detected landmarks (83) compared to ground truth.**



**Figure 34(b). BU-4DFE detected landmarks (83) compared to ground truth.**

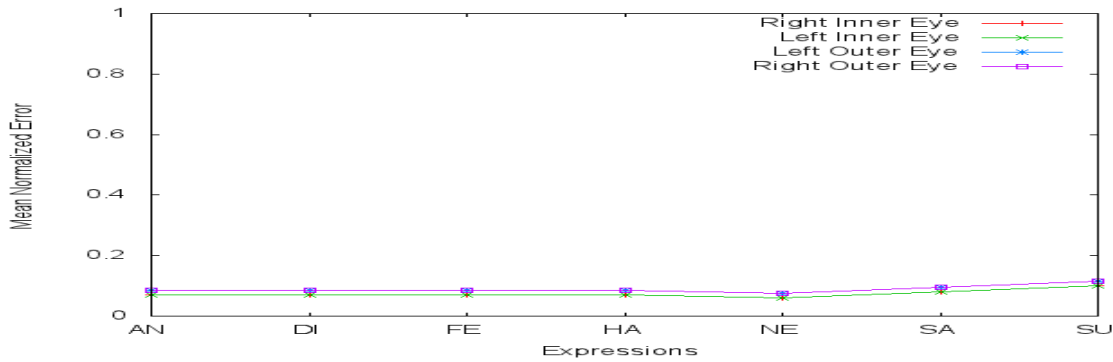
We have also compared our result of mean square error of the average point spacings to the work reported by Sun et al [48]. Our MSE for BU-4DFE is 3.7, which shows a significant improvement over the result of 6.25 as reported in [48]. Fig. 35 shows the average errors on each of 83 points using our approach and the approach in [48].



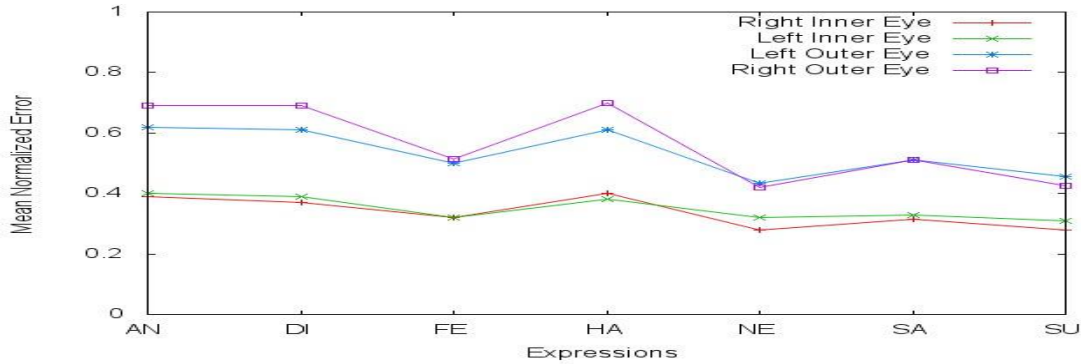
**Figure 35. Point space comparison with our ASSM method and Sun et al [13].**

In addition to the ground truth comparisons and the MSE comparison to Sun et al [48] we have also compared our results to the work reported by Nair et al [55] on the BU-3DFE database. Following their method, we selected four landmarks, the inner and outer

corners of the left and right eye, to compare to the ground truth. We achieved an approximate error rate of 0.09 as compared to their approximate error rate of 0.44. Fig. 36(a) shows our mean normalized error, and Fig. 36(b) shows the mean normalized error of [55]. The mean normalized error is the average error between the ground truth and the localized landmarks normalized to  $[0, 1]$ . The evaluation shows that our feature tracking approach outperforms [55] as our approach does not rely on candidate landmarks to guide the fitting. The evaluation also shows that our ASSM approach is robust to various expressions and has a consistent error rate across all selected features. This can be seen in the minor differences in error rate across each expression on all four of the selected features.



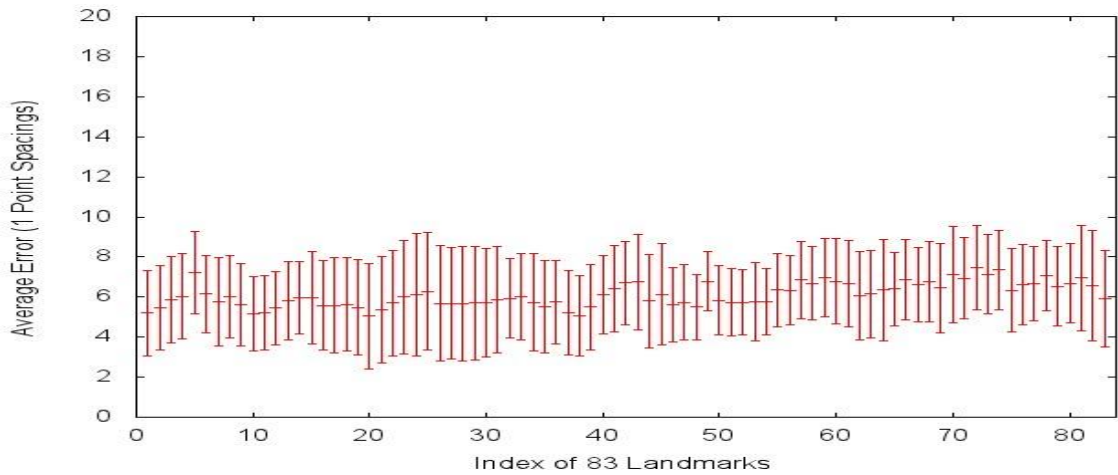
**Figure 36. (a) Mean normalized error of our ASSM method. (Key: AN=angry, DI=disgust, FE=fear, HA=happy, NE=neutral, SA=sad, SU=surprise).**



**Figure 36. (b) Mean Normalized error of Nair *et. al.* [55](Key same as Fig. 13(a)).**



We also compared tracking results from our BP4D-Spontaneous database to those obtained by using the Kinect face tracking API [5] on the same set of 3D range models. In order to do this comparison we need to modify the Kinect face tracking algorithm to work with our 3D range data instead of the depth and RGB data acquired from the Kinect. Here is a brief description of the modified Kinect face tracking (MKFT) algorithm. First we need to do multi-rendering in order to render our 3D range data in a suitable depth and RGB format to be used for the tracking. Next a position map is used to convert the 2D coordinates in the rendering space to the model space to acquire the 3D landmarks. Due to fitting error in the rendering space, there are some face contour errors that need to be adjusted by using an error minimization algorithm. Once these steps are done we are able to compare the same 83 face landmarks using the MKFT algorithm and our ASSM algorithm. Fig. 37 shows the 1 point spacing between our ASSM algorithm and the landmarks from MKFT algorithm.

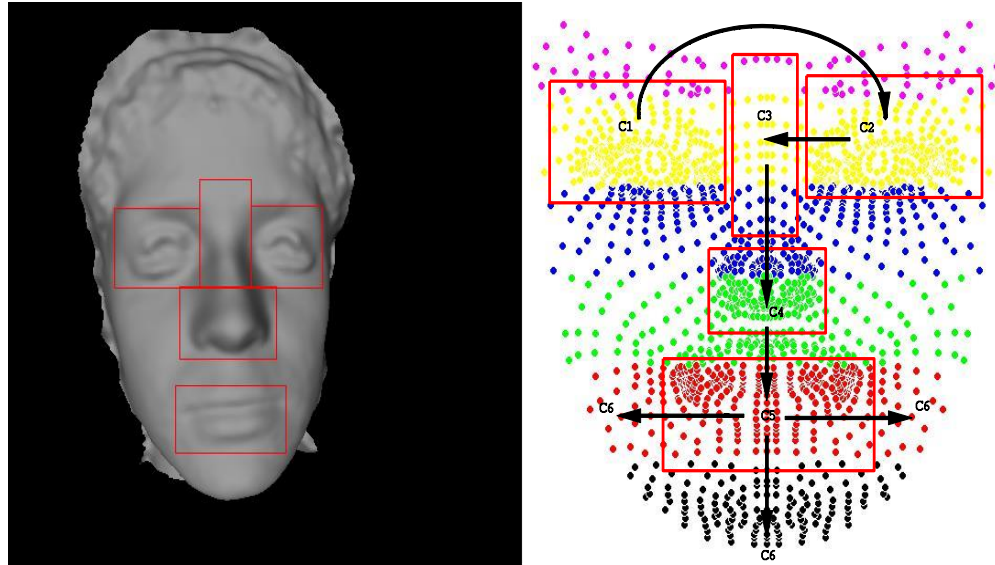


**Figure 37. Point space comparison on 83 landmarks between our ASSM algorithm and the MKFT algorithm.**



### 4.3 Subject and Expression Verification

To validate our proposed method, we apply it to subject verification and facial expression classification problems. We take the component based approach for the classification. Given the tracked feature points, we can easily segment the facial model into several component regions, such as the eyes, nose and mouth. Fig. 38(a) shows an example of the resulting segmentation.



**Figure 38. (a) Sample of component regions; (b) component-based HMM based on six component regions.**

#### 4.3.1 3D Component Feature Representation

3D facial models can be characterized by their surface primitive features. This spatial feature can be classified by eight types: convex peak, convex cylinder, convex saddle, minimal surface, concave saddle, concave cylinder, concave pit, and planar. Such a local shape descriptor provides a robust facial surface representation. To label the model surface, we select the vertices of the component regions and then classify them into one of the primitive labels. The classification of surface vertices is based on the surface

curvature computation [48]. After calculating the curvature values of each vertex, we use the categorization method [45] to label each vertex on the model. As a result, each range model is represented by a group of labels that construct a feature vector:  $G = (g_1, g_2, \dots, g_n)$ , where  $g_i$  represents one of the primitive shape labels, and  $n$  equals the number of vertices in the component region.

Due to the high dimensionality of the feature vector  $G$ , where each of six component-regions contains between 300 and 700 vertices, we use a Linear Discriminant Analysis (LDA) based method to reduce the feature space of each region. The LDA transformation maps the feature space into an optimal space where different subjects are easily differentiated. It then transforms the  $n$ -dimensional feature  $G$  to the  $d$ -dimensional optimized feature  $O_G$  ( $d < n$ ).

#### 4.3.2 Spatial HMM Model Classification

As shown in Figure 14(b), each frame of the 3D facial model is subdivided into six components (sub-regions)  $C1$ ,  $C2$ ,  $C3$ ,  $C4$ ,  $C5$ , and  $C6$ , including regions of the eyes, nose, nose bridge, mouth, and the remaining face. From  $C1$  to  $C6$ , we construct a 1-D HMM [97] which consists of the six states ( $N = 6$ ), corresponding to six regions.

We transform the labeled surface to the optimized feature space using the aforementioned LDA transformation. Given such an observation of each sub-region, we can train the HMM for each subject. Given a query face model sequence of length  $k$ , we compute the likelihood score for each frame, and use the Bayesian decision rule to decide which

subject each frame is classified to. Since we obtain  $k$  results for  $k$  frames, we use a majority voting strategy to make a final decision. As such, the query model sequence is recognized as subject  $Y$  if  $Y$  is the majority result among  $k$  frames. This method tracks spatial dynamics of 3D facial surfaces, where the spatial components of a face give rise to the spatial HMM to infer the likelihood of each query model. Note that if  $k$  is equal to 1, the query model sequence becomes a single model for classification.

#### 4.3.3 Temporal HMM Model Classification

For the 3D expression sequences, we treat 6 frames as the 6 states of the HMM model for expression classification. When observing the state change of a local region across a sequence, we are able to use the facial features of the local region to train a temporal HMM. Each local region of a facial surface learns an HMM for each distinct expression separately. Given six local regions and six prototypic facial expressions, a total of 36 T-HMMs are established for an entire facial surface.

Since the features extracted from the six local regions could generate six different classification results, we use the majority voting strategy to determine the expression type of the subsequence. If more than two regions are classified as a same expression, such expression is taken as the recognized expression for this subsequence. If there is no majority expression to be recognized among the six regions ( $R1, \dots, R6$ ), the expression with the maximum likelihood (probability) of the region will be chosen as the recognized expression of this subsequence. This procedure is formulated as the following equation:

$$R_c = \operatorname{argmax}_{R_k} \left[ \frac{P(\omega_{c*}^k | O^{R_k})}{\sum_{i=1}^C P(\omega_i | O^{R_k})} \right]_{k=1,2,\dots,6} \quad (17)$$

where  $\omega_{c*}^k$  is the expression type determined by the region  $R_k$ ,  $\omega_i$  is a trained HMM model,  $C$  is the number of trained HMM models, and  $O$  is an observation sequence. As a result, the expression of the region  $R_c$  with the maximum likelihood is selected as the recognized expression of the subsequence. In summary, the regional features of a facial surface are used to learn their temporal changes, and the classified expression is determined by either the majority voting or the maximum probability of observations of local regions. ASSM experimental results for both subject verification and expression classification follows.

#### 4.3.4 Subject Verification and Face Expression Classification

The BU-4DFE database was used for both subject verification and face expression classification purposes. For each training sequence of 4DFE, 20 sets of three consecutive frames were randomly chosen for training. Following the HMM training procedure ( $k = 3$ ), we generated an HMM for each subject. The recognition procedure is then applied to classify the identity of each input sketch sequence ( $k = 3$ ) as the previous section described. Based on the 10-fold cross validation approach, the correct recognition rate is approximately 95%. For face expression classification, the six prototypic facial expressions are classified with an accuracy of approximately 93%.

#### 4.4 Expression Segmentation (Action vs. Non-Action)

A natural extension of our proposed method is the application of expression segmentation (or facial event detection) across a sequence of facial models. Each of the ASSM instances has been labeled with a subject, an expression, and the inter-frame constraint. Using this labeling we are able to compare each set of detected landmarks to the original instances of the ASSM, which is the same method as described in section 3. Given (16), we can find the distance  $D$  from the detected landmarks to each of the ASSM instances. The smallest  $D$  value must correspond to a minimum threshold for a correct classification.

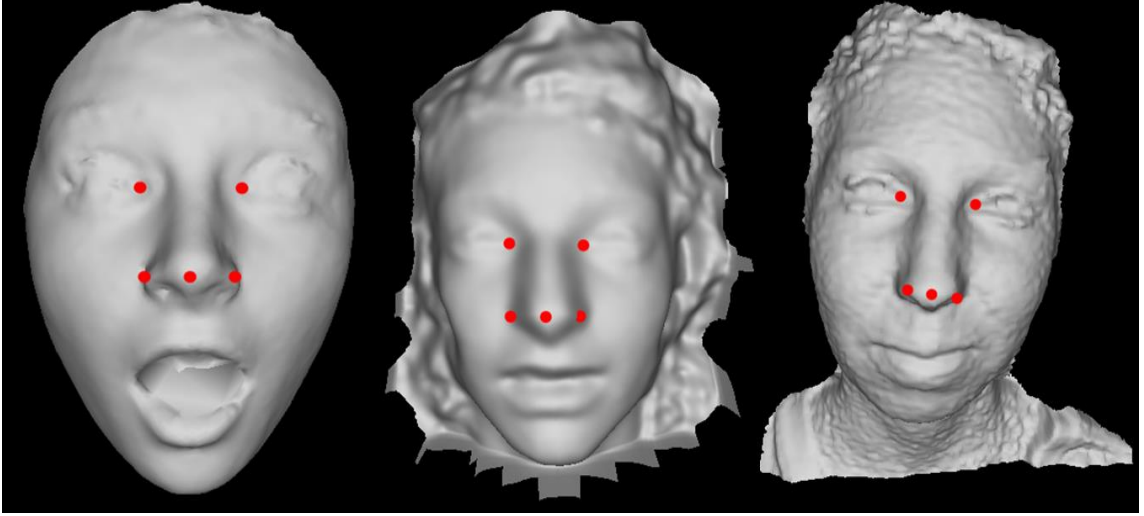
For the purposes of expression segmentation, we classify the results into one of two categories: either an action (onset/offset, and peak), or non-action (neutral expression). To analyze these results, we manually segmented sequences from the BU-4DFE [13] and BP4D-Spontaneous [85] databases and compared the automatic segmentation with this ground truth data. We achieved approximately 86% and 81% correct classification rates in terms of action vs. non-action segmentation across all expressions for the BU-4DFE and BP4D-Spontaneous databases respectively. Note that the data is very challenging for all expressions in the BP4D-Spontaneous database. Fig. 39 illustrates an example from this database of segmentation on a sequence where the subject is smiling.



**Figure 39.** An example of spontaneous expression segmentation (action/non-action) on the 4D spontaneous expression database. *Note: the texture shown in the top row is for illustration purposes only.*

#### 4.5 Pose Estimation

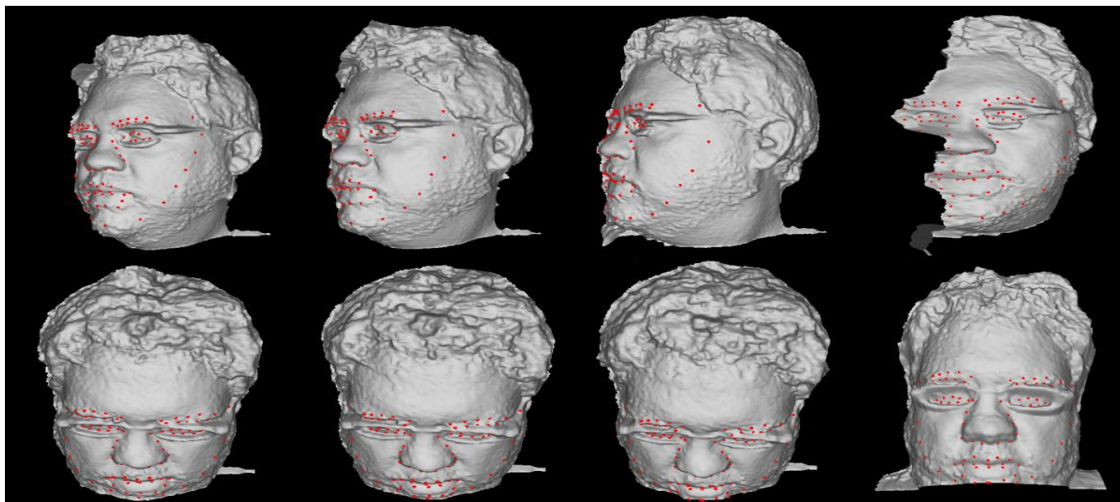
Using the ASSM method, we are able to accurately detect landmarks on sequences consisting of rotations displaying roll, pitch, and yaw as seen in Figs. 26 and 32. Using these detected landmarks, another natural extension is pose estimation. In order to perform pose estimation, we use a simple, yet effective method. To calculate the pose estimation, we use four of the feature points which are a subset of the  $N = 83$  landmarks we detected on the face models used in this chapter. From these detected landmarks we calculate a normal vector that we then use for the pose estimation. The four landmarks used to calculate the normal vector are the left and right inner eye corners, as well as the left and right corners of the nose. Given these four landmarks, a triangle is formed by each eye's inner corners and the average point of the two nose corners. The normal vector of such a triangle is relatively expression invariant, as can be seen in Fig. 40 thus representing the pose orientation of the head accordingly. We then use the relative rotation of the normal vector, compared to a model that is displaying a frontal view, to calculate the head pose angle.



**Figure 40. Sample Illustrations, on BU-3DFE, BU-4DFE, and BP4D-Spontaneous, showing the landmarks used to create the normal vector used to determine head pose. *Note: the landmarks on the nose tip regions have been translated along the z axis for illustration purposes only. This landmark, being the average of the nose corners, would normally not be visible from this view.***

To test the accuracy of this method, we selected one model per subject from the BU-3DFE database [31] giving us a total of 100 different face models. We then manually rotated each model, as can be seen in Fig. 26, from a full frontal view to a full profile view (only yaw rotation was used for the pose estimation calculations). We saved the models every 10 degrees  $[0, 90]$ , giving a total of 1000 rotated frames (including the frontal views). We used this data as the ground truth angle to compare our automatic method to. We then calculated the MSE of our estimated pose to the ground truth. Across all degrees  $[0, 90]$ , our resulting MSE is 0.00041 degrees, with most degrees having an individual MSE of 0. We also compared our pose estimation results using 1700 models from the BP4D-Spontaneous database to the pose obtained from the modified Kinect [5] face tracking algorithm as detailed in section 4.2. The comparisons show 2.53, 1.35, and

2.44 differences in degree across pitch, roll, and yaw respectively. Fig. 41 shows models from our in-house scanner, displaying yaw and pitch, with estimated pose.



**Figure 41.** Sample frames from our in-house scanner displaying pitch and yaw pose estimations. Top Row (Yaw): -37, -49, -51; Bottom Row (Pitch): -20, -23, -27. *Note: The last column is the same model from the previous column. The view is changed to show the deformations that this degree of pose shows.*

## 4.6 Gesture Recognition

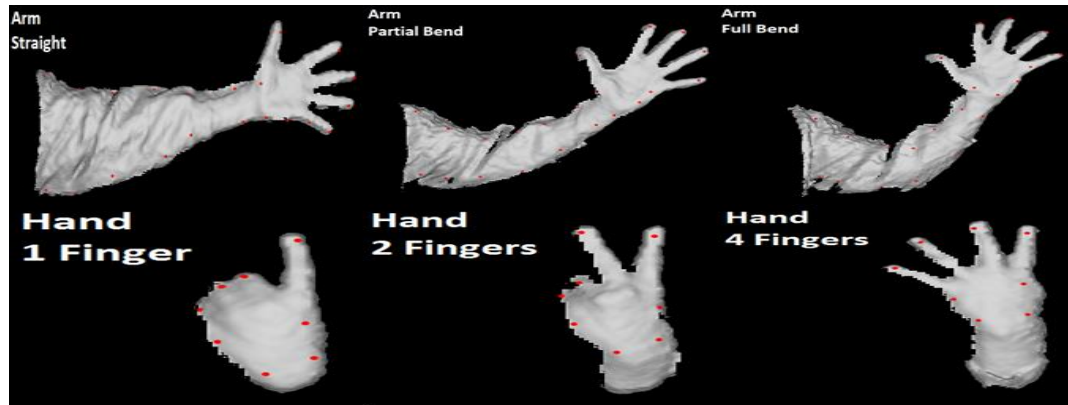
Similar to the experiments detailed in sub-section C, where each ASSM has been labeled with a corresponding subject, expression, and constraint, each ASSM for the arm and hand models have been labeled with both an object and action type. As in the experimental setup of sub-section C, we again compare the detected landmarks to the ASSM instances. Given Equation (16), we can find the distance  $D$  from the detected landmarks to each of the ASSM instances. The smallest  $D$  value must again correspond to a minimum threshold for a correct classification. Table 12 details the object and action types for both the hand and arm models. For 6 hand actions and 3 arm actions, we achieved 100% and 97% recognition for the type of model and action present



respectively. Fig. 42 shows sample hand and arm models along with the automatically labeled type and action. This method is also applicable to other object types, as well as extendible to more actions for each object type.

**Table 12. Object and action type for hand and arm ASSM.**

| <b>Object and Action Type for Hand and Arm ASSM</b> |                   |                   |                     |                    |                    |                   |
|---|-------------------|-------------------|---------------------|--------------------|--------------------|-------------------|
| <b>Object Type</b>                                  | <b>Action One</b> | <b>Action Two</b> | <b>Action Three</b> | <b>Action Four</b> | <b>Action Five</b> | <b>Action Six</b> |
| <b>Hand</b>   | 1 Finger          | 2 Fingers         | 3 Fingers           | 4 Fingers          | 5 Finger           | Closed Hand       |
| <b>Arm</b>  | Straight          | Partial Bend      | Full Bend           | N/A                | N/A                | N/A               |



**Figure 42. Sample frames from Microsoft Kinect [86], showing correct automatic labeling of object and action type for an arm and hand.**

## 5. Discussion

In this chapter, we have presented a new 3D/4D action-based statistical shape model for detecting key landmarks on both 3D and 4D range mesh models. The ASSM method has been tested on 4 public face databases, as well as non-face data collected from our in-house scanner. The method is able to accurately model large rotations, with deformations of the mesh, and range data of occluded faces. We have evaluated the accuracy of the

feature detection and validated its utility for subject verification and segmentation, pose estimation. While the algorithm itself is relatively fast, the off-line selection of landmarks for training data is a bottleneck. Assuming we have a large selection of data that has been trained, an ASSM lends itself well to parallel architecture. Each of the frames in the ASSM can be simultaneously fit to the input range data. This has the potential to allow real-time 3D detection and tracking of facial landmarks for high resolution range data.

## **Chapter 7**

### **3D/4D Feature Detection Using Shape Index-based**

#### **Statistical Shape Models**

##### **1. Introduction**

Applications such as face recognition, expression analysis, human-computer interaction, and face video segmentation are increasingly being developed based on 3D, and 4D (3D+time) range data [137][138][139][142][47][144], given the rapid technological advancement of 3D imaging systems [74][141][86] [84]. Landmark localization on 3D/4D range data is the first step toward geometric based vision research for object modeling, recognition, visualization, and scene understanding [89][140][100] [101][108].

While 2D based tracking methods have been successfully developed, such as Active Shape Models [57], Active Appearance Models [131], using a consensus of exemplars [124], Constrained Local Models (CLM) [127], regularized landmark mean-shift [134], generative shape regularization model [128], explicit shape regression [125], supervised descent method for face alignment [135], and shape-constrained linear predictors [98], there is a need for novel and robust algorithms to handle 3D/4D range data. Morphable Model [58] is one of the successful algorithms for handling 3D range data.

There has been recent work to address the problem of detecting feature landmarks on range data. Zhao et al. [93] had success with detecting 3D landmarks using a statistical

facial feature model; however there is an upper bound on the number of landmarks. Fanelli et al. [92] used an active appearance model that is based on random forests; however this method used depth and intensity data rather than the 3D/4D range data. Sun et al. [48] used a so-called vertex flow approach, which used an active appearance model (AAM) to track features of 3D range models. However, the tracking of facial features was not truly in the 3D space, rather it was tracked in the 2D space and the 3D features themselves were obtained by mapping the 2D features to the corresponding parts of the 3D models, tending to cause inaccurate projections. Nair et al. [55] fit a 3D active shape model to facial data using candidate landmarks to deform the model, however the resulting error rate for fitting is relatively large, and problems occur when holes exist around the nose. Zhou et al. [104] created a 3D active shape model which was trained using a 3DMM, although the fitting for this method was done in 2D. Perakis et al. [133] used a 3D active shape model which was fit from previously determined candidate landmarks. A draw-back to this method is the need for preprocessing. Guan et al. [94] performed landmark localization on facial data by utilizing a Bezier surface. This method was tested on a small dataset consisting of 100 3D models. Jeni et al. [129] used a 3D constrained local model method (estimated from 2D shape) to track landmarks for action unit intensity estimation. Baltrusaitis [105] used a 3D CLM (a.k.a. CLM-Z) trained with depth data rather than 3D/4D range data for rigid and non-rigid feature tracking. A statistical model (blend-shape) was utilized by Weise et al. [95] to track facial data and animate a virtual avatar; however, this has the limitation that the blend-shape may not have a unique set of needed weights for an expression. Chen et al. [140][100] applied a coarse-to-fine approach via curvature and active normal model for landmarking.

Recently, we have developed a so-called 3D temporal deformable shape model (TDSM) for feature tracking through 3D range sequences [96]. However, such a multi-frame based shape model may not work well for different expressions within a very short duration when dramatic motions or sudden expression changes occur in the 3D videos, thus the performance on 3D geometric tracking still needs to be improved. Motivated by the previous work [96], we continue to address the issue of feature detection and tracking on 3D/4D range data with a more reliable way.

In this chapter, we propose to construct a *shape index-based statistical shape model (SI-SSM)* with both global and local constraints. The SI-SSM is constructed from both the global shape of 3D feature landmarks and local features from patches around each landmark. In order to construct the patches we find 3D features from the  $(u, v)$  coordinates around each landmark. From these new features we construct a  $n \times n$  patch, where each vertex is represented by a unique shape index value. Using both the global shape and the local features around each landmark enables us to reliably detect and track features on the range mesh data. The feature detection and tracking are based on finding the correlation between the local shape index patches on the input range data and the trained SI-SSM model (as illustrated in Figure 3).

The main contribution of this chapter is the construction of a statistical model that makes use of both the global shape of 3D face surface, as well as the local shape around individual features by way of shape index representation. This model can be used to detect and track features on range data. By using the shape index representation we are

able to make the local fitting invariant to both lighting and pose changes. We are able to model and fit data that includes various emotions, rotations, occlusions, and missing data by training on each of these data types. Following is the summary of the main contribution of this work:

- (1) We proposed and developed a novel approach for 3D/4D facial feature detection and tracking. This approach has extended the global statistical shape model to an integrated global and local shape model to improve the tracking performance with respect to various imaging data conditions. In particular, we have presented a shape-index based local shape model and combined this model with the global shape model as a new statistical shape descriptor (so-called *shape-index based statistical shape model (SI-SSM)*).
- (2) We have tested the new SI-SSM model on five public 3D/4D face databases (i.e., BU-3DFE [31], BU-4DFE [13], BP4D-Spontaneous [136], FRGC 2.0 [84], and Eurecom Kinect Face Database [132]) which cover a variety of data types, including static vs. dynamic, posed vs. spontaneous, high-resolution vs low-resolution, etc.
- (3) We show the merit of the new SI-SSM based detection and tracking through performance evaluations with respect to various authentic facial behaviors, dramatic head rotations, data conditions with noise, occlusion, and incompleteness, as well as comparison with four state of the art approaches.
- (4) We have validated the usability of our new approach through its application to facial expression recognition and head pose estimation. Especially, we applied a *spatial-*

*temporal HHM model* to classify six posed expressions on 4DFE and eight spontaneous expressions on BP4D-Spontaneous database successfully.

The chapter is organized as follows: Section 2 presents the new statistical model and its construction. Section 3 describes the feature detection and tracking algorithm in detail. The experiments and evaluations are reported in Section 4, followed by application study for 3D/4D face analysis. Finally, the conclusion and future work are discussed in Section 6.

## **2. Shape Index-based Statistical Shape Model (SI-SSM)**

Our proposed method models both the global shape of 3D facial landmarks, as well as the local curvatures from patches around the landmarks. In order to construct the SI-SSM, we annotate the training data with  $L$  landmarks. From these annotated landmarks we are able to model both the global and local shapes of a face. An example of an annotated mesh can be seen in Figure 43, where  $L=83$ . The resulting global shape, local curvature patches, and the final construction of the SI-SSM are detailed in the following subsections.

### **2.1 Global Face Shape**

To model the global face shape, we first create a  $n \times n$  patch around each of the  $L$  annotated landmarks for each training mesh. To construct these patches we use the corresponding (u, v) coordinates for each of the training data. An example of a 3D mesh with patches can be seen in Figure 1.

Given a set of  $M$  training data, each with  $L$  patches, a parameterized model,  $S_G$ , is constructed. This parameterized model contains the global shape of all of the training data, where  $S_G = (x_1, y_1, z_1, \dots, x_N, y_N, z_N)$ , where  $N = L \times n \times n$ . The first step to create this model is aligning the  $N$  landmarks, on each of the  $M$  training data, by using a modified version of Procrustes analysis [57]. PCA is then applied to learn the modes of variation from the training data. For our experiments we keep approximately 95% of the variance. We can then approximate any shape by

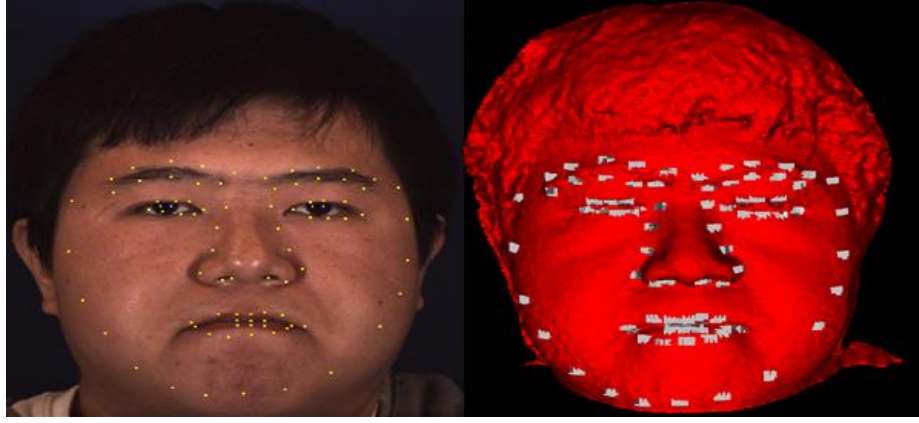
$$S_G = \bar{s} + Vw \quad (18)$$

where  $\bar{s}$  is the mean shape,  $V$  is the eigenvectors of the covariance matrix  $C$ , which describes the modes of variation learned from the training data, and  $w$  is a weight vector used to generate new shapes (referred to as an instance of the SI-SSM) by varying its parameters within certain limits. We impose these limits to ensure only valid shapes are constructed. For our model we constrain those valid shapes to be within two standard deviations from the mean (which is the allowable shape domain)

$$-2\sqrt{\lambda_i} \leq w_i \leq 2\sqrt{\lambda_i} \quad (19)$$

Where  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of  $C$ . We have empirically found, from the training data,  $\pm 2$  standard deviations from the mean to be a suitable constraint for our model as this range gives us a good balance between speed and accuracy of model fit. A smaller constraint would shrink the search space and possibly miss the best fit to the input model. A larger domain would create an unnecessarily large search space that would have instances of the model that do not look like a face.





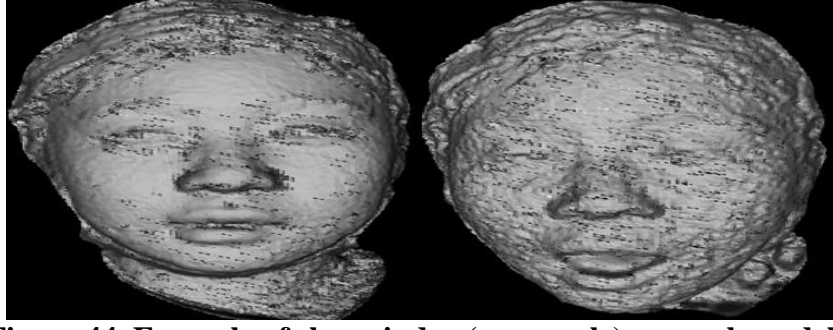
**Figure 43. Left: 83 landmarks defined on a face; Right: corresponding 3D patches with grey-scale shape index values.**

## 2.2 Local Face Shape

To model the local face shape we apply the shape index values to represent the local patches. To do so, we calculate the shape index values for each of the  $L$  patches in the global face shape. Calculating the shape index gives us a quantitative measure of the shape of each patch around the  $L$  annotated landmarks. Shape index is defined as follows:

$$SI = \frac{2}{\pi} * \arctan\left(\frac{k_2 + k_1}{k_2 - k_1}\right) \quad (20)$$

where  $k_1$  and  $k_2$  are the min and max principal curvatures of the surface, with  $k_2 \geq k_1$ . All shapes can be mapped to the range  $[-1.0, 1.0]$ , where each unique shape corresponds to a specific shape index value. A cubic polynomial fitting approach is used to compute the eigen-values of the Weingarten Matrix [61] giving us  $k_1$  and  $k_2$ . We normalize the shape index scale to  $[0, 1]$  and encode them as a continuous range of grey-level values between 1 and 255. To give us an efficient description of the data, we transform the shape index scale to a set of nine quantization values from concave to convex. Figure 44 shows example range meshes with the shape index values normalized to  $[1, 255]$  for illustration.



**Figure 44. Example of shape index (grey-scale) on mesh models.**

Given the set of  $M$  training data with  $L$  patches where each contains the calculated shape index values, we construct a second parameterized model  $S_L = (SI_1, \dots, SI_N)$ . PCA is then applied to this local shape vector in the same manner as the global shape vector does. We construct a new vector,  $V_{SI}$ , which yields of the modes of variation along the principal axes for the local shape index values. Similar to the global shape, we can approximate any local patch shape using the vector,  $V_{SI}$ , and a weight vector  $w_{SI}$  by

$$S_L = \bar{s}l + V_{SI}w_{SI}. \quad (21)$$

### 2.3 Combined Global and Local Feature Model

To take both global and local shape constraints, we integrate the two features into a combined feature model. To do so, we concatenate both the global and local shape feature vectors into one feature vector  $S_{GL}$ , where  $S_{GL} = (x_1, y_1, z_1, \dots, x_N, y_N, z_N, SI_1, \dots, SI_N)$ . Using the combined feature vector allows us to move the local patches, on the face data, to a more representative surface on the model while maintaining the constraint that we approximate a valid face shape in the allowable shape domain. Other methods that use statistical

models such as [126][127] have been successful in using statistical models to create a combined feature vector that incorporates both the shape and “appearance” of the face. The “appearance” portion (e.g. textures) of the model helps to guide the model and fit to new data, however, these approaches suffer from the problem of global lighting variation, as well as skin tone of the modeled face. The grey-level appearance information in these models must be normalized in order to handle this lighting variation. Our SI-SSM uses shape index values to model our local features, which guide our model and fit to new range data. Shape index values are invariant to global lighting variation and skin tone. As described in the previous section, shape index is a quantitative measure of shape, so using these features our model does not encounter the same issues that similar “appearance” based solutions do.

### **3. 3D/4D Landmark Detection and Tracking**

Given an SI-SSM we are able to detect and track landmarks on 3D/4D sequences of range data. In order to perform the detection and tracking, we must first calculate the shape index values for the vertices of the input range mesh. This is done in the same manner as described in Section 2.2. Once we have these values calculated we can then apply the SI-SSM fitting algorithm to the input range mesh data.

First, an initialization phase is performed to give us a sufficient starting point to perform a local patch-based correlation search. During the initialization phase, to fit our model to the range data we learn the weight parameters  $w$  of the global shape by uniformly varying the weight vector to generate new instances of the SI-SSM. By performing this learning

offline, for the initialization, we are able to have precise control over which shapes are constructed, ensuring that the new shapes constructed are valid (within the allowable shape domain). Iterative closest point (ICP) [48] is used to minimize the distance between each SI-SSM instance and the input range data. The patches from the instance of the SI-SSM with the lowest ICP matching score are used as the initialized starting landmarks for the SI-SSM. Given this global fit, we then calculate the local patch-based correlation score with the SI-SSM and the input range mesh. This correlation score is computed using a cross correlation template matching scheme [130]. The correlation score,  $CS_p$  is computed for each patch as:

$$CS_p = \frac{\sum_{i',j'} (P(i',j') \cdot R(i+i',j+j'))}{\sqrt{\sum_{i',j'} P(i',j')^2 \cdot \sum_{i',j'} R(i+i',j+j')^2}} \quad (22)$$

where  $P(i',j')$  is the computed shape index value at index  $(i',j')$  of the SI-SSM patch, and  $R(i+i',j+j')$  is the summation between the shape index value at index  $(i',j')$  of the SI-SSM patch and the corresponding shape index value on the range mesh. The final correlation score,  $CS$ , is computed as

$$CS = \sum_{p=1}^L CS_p \quad (23)$$

This initial correlation score allows us to have a base line comparison for the local patch-based correlation search, as well as define tighter convergence criteria.

Once we have the initialized patches and initial correlation score we then perform a local search around each of the patches of the SI-SSM. For each patch in our model we construct a new patch of the same size around each of the  $n \times n$  points of the original patch. For example, when  $n=3$ , we construct a patch centered on each point of the

original  $3 \times 3$  patch, resulting in 9 new patches (as illustrated in Figure 3). The shape index values for each of these patches correspond to the shape index values of the vertices of the new patches. Using Equation 22 we compute a new  $CS_p$  for each of the new patches we created. The patch that gives us the highest correlation score is marked as the new patch of the SI-SSM. It is important to make sure that when all of the patches have been moved the new global shape of the face is within the allowable shape domain of  $\pm 2$  standard deviations from the mean. From Equation 21, we can derive the corresponding  $w_{SI}$  vector of the newly transformed SI-SSM by the following:

$$w_{SI} = V_{SI}^T (S_L - \bar{S}_i) \quad (24)$$

This new weight vector is constrained to be within the allowable shape domain, and we approximate a new shape by again utilizing Equation 21 with this weight vector.

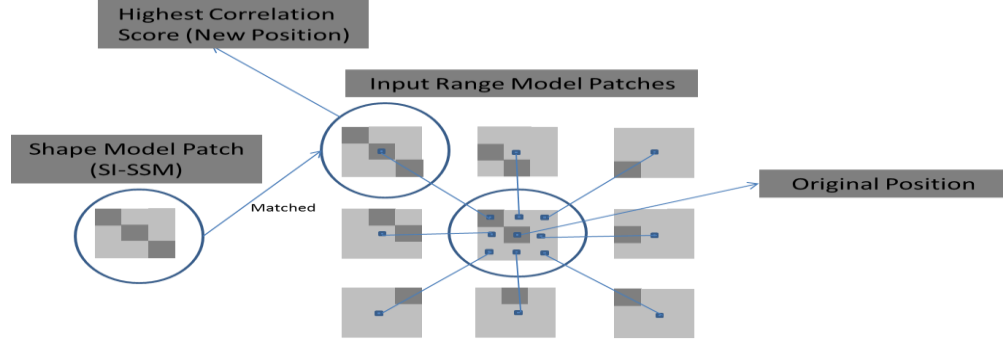
Once we have the new approximated global shape of the face, iterative closest point is then used to again minimize the distance between the new SI-SSM instance and the range mesh. This process continues until convergence is reached. Convergence is defined by two main criteria:

- (1) The computed correlation score,  $CS$ , for the transformed SI-SSM is higher than the computed score in the previous iteration (for the first iteration we make use of the correlation score computed in the initialization phase).
- (2) The computed correlation score,  $CS$ , exhibits little to no change from the  $CS$  computed in the previous iteration.

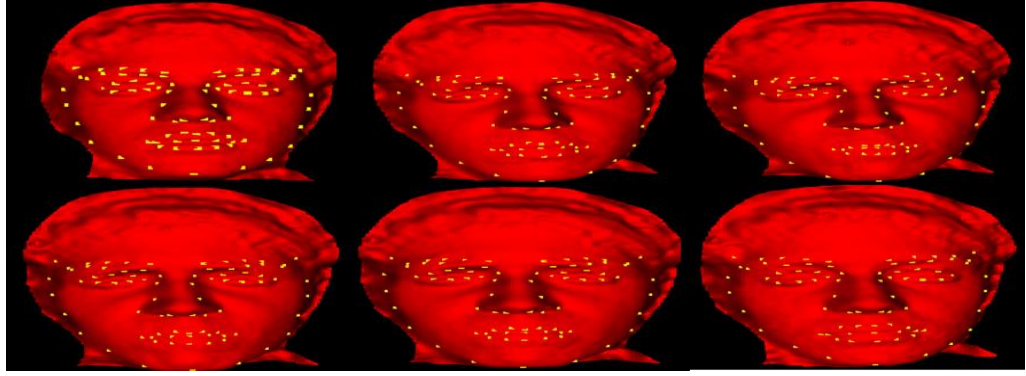
If the first convergence criterion is satisfied after the first iteration after initialization, the transformed patches are discarded and the previously computed global patch shape is used. Due to this, we need to compute the initialization correlation score as it is possible in our initialization phase that our SI-SSM will find the best fit to the range mesh, and additional transformation(s) of the model are not required. Once we have the detected features for the current 4D mesh in the sequence, we then use ICP to move the landmarks to the next mesh in the sequence and continue the tracking of the sequence. The fitting process is then repeated with the previously detected landmarks used as the initial model fit. Table 13 outlines the algorithm, Figure 45 shows an example illustration outlining the fitting process, and Figure 46 shows several sample 4D range models with detected patches using the SI-SSM algorithm.

**Table 13. SI-SSM fitting algorithm.**

| <b>SI-SSM FITTING ALGORITHM</b>                              |   |
|--|---|
| <b>Input:</b> Range mesh model                               |   |
| 1.   | Learn weight parameters for SI-SSM instances.   |
| 2.   | Initialize SI-SSM by using ICP to minimize distance between instances and input range mesh model. |
| 3.   | Calculate correlation score, $CS$ , for initialized SI-SSM.                                       |
| 4.   | Perform local patch-based correlation search.   |
| 5.   | Constrain transformed patches from step 4 to be within allowable shape domain.                    |
| 6.   | Calculate new correlation score for newly transformed patches                                     |
| 7.   | Compare new correlation score to score of previous iteration.                                     |
| 8.   | Repeat steps 4-7 until convergence.   |
| <b>Output:</b> Detected patch landmarks on input range mesh. |   |



**Figure 45.** Example of correlation search between a SI-SSM patch and input range model patch at size of  $n \times n$ , (where  $n=3$  for instance).



**Figure 46.** Tracked frames from BU-4DFE displaying an angry expression.

## 4. Experiments and Evaluation

### 4.1 Databases

Five public face databases have been used for our study including three static and two dynamic databases (as shown in Table 14, and Figures 47 and 48 for examples).

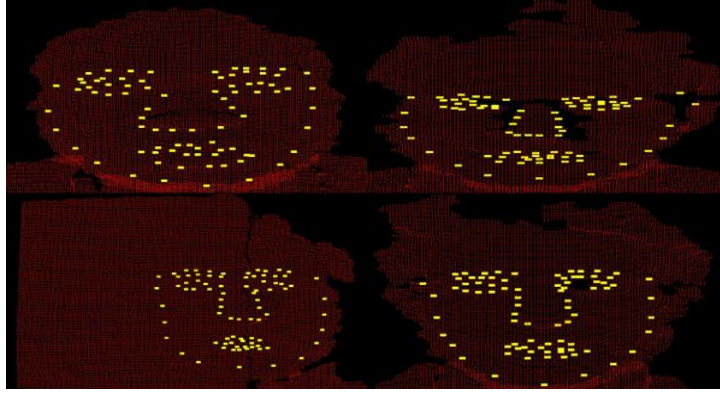
- (1) BU-3DFE [31] consists of 100 subjects each displaying one neutral expression and four intensity levels of six deliberate expressions.
- (2) Eurecom Kinect Face Database [132] consists of 52 subjects, displaying 9 deliberate expressions, obtained through the Microsoft Kinect [86].

- (3) FRGC 2.0 [84] consists of 466 subjects displaying two different deliberate expressions.
- (4) BU-4DFE [13] consists of 101 subjects with sequences of six different deliberate expressions.
- (5) BP4D-Spontaneous database [136] consists of 41 subjects, each consisting of 8 different spontaneous expression sequences (e.g., joy, embarrassment, surprise, disgust, fear, sadness, pain, and anger). The expressions were elicited through activities including film watching, interviews, and experiencing cold pressor test, etc. The database includes the 3D dynamic model sequences, texture videos, and annotated action units (AU). Table 2 lists more details pertaining to each database.

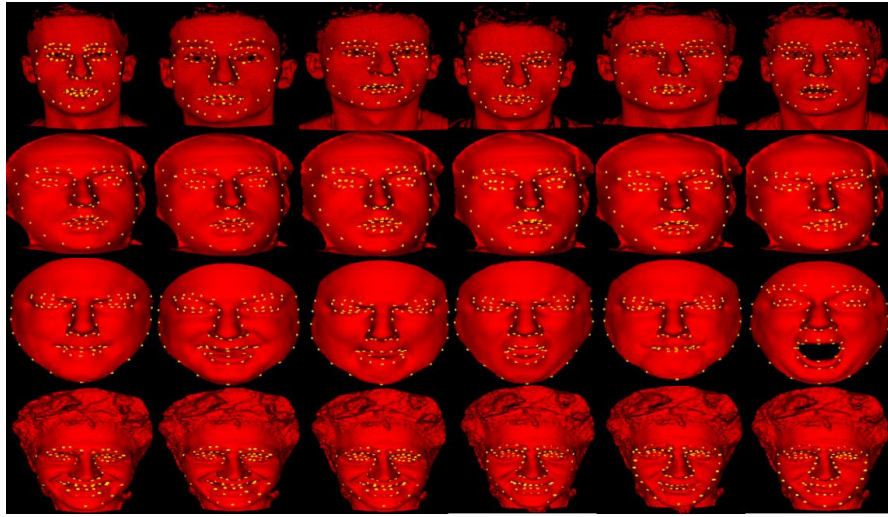
**Table 14. Summary of 3D/4D Databases.**

| <b>3D/4D DATABASE SUMMARIES</b> |                 |             |                           |                                   |                              |                                   |
|---------------------------------|-----------------|-------------|---------------------------|-----------------------------------|------------------------------|-----------------------------------|
| <b>Database</b>                 | <b>Modality</b> | <b>Type</b> | <b>Number of Subjects</b> | <b>Resolution (# of vertices)</b> | <b>Number of Expressions</b> | <b>Number of Models</b>           |
| 3DFE                            | Static          | Deliberate  | 100                       | 8,000                             | 7                            | 2,500                             |
| 4DFE                            | Dynamic         | Deliberate  | 101                       | 30,000                            | 6                            | 606 Sequences (100 frames/seq.)   |
| FRGC 2.0                        | Static          | Deliberate  | 466                       | 100,000                           | 2                            | 932                               |
| BP4D                            | Dynamic         | Spon.       | 41                        | 50,000                            | 8                            | 328 Sequences (1,500 frames/seq.) |
| Eurecom                         | Static          | Deliberate  | 52                        | 65,000                            | 9                            | 936                               |





**Figure 47.** Sample frames fit with SI-SSM algorithm from the Eurecom Kinect Face Database, showing robustness to occlusion, noise, and missing data.



**Figure 48.** By row: FRGC 2.0, BU-4DFE, BU-3DFE, and BP4D-Spontaneous

## 4.2 Feature Detection and Tracking on Five Databases

To evaluate the accuracy of detecting and tracking landmarks using our SI-SSM method, we calculate the mean squared error between the ground truth and our detected/tracked landmarks (centroids of patches). We do this by calculating the one-point spacing between each landmark. The one-point spacing is defined as the closest pair of points on the 3D scans (0.5mm on the geometric surface). We treat the unit error as equal to 1 point

spacing, so we can compute the average of the point distances between the sets. Table 15 details the error rates, mean squared error (MSE), for all five tested databases.

**Table 15. Error rates for all 5 databases.**

| <b>Database</b>             | <b>3DFE<br/>[31]</b> | <b>4DFE<br/>[13]</b> | <b>FRGC 2.0<br/>[84]</b> | <b>BP4D<br/>[136]</b> | <b>Eurecom<br/>[132]</b> |
|-----------------------------|----------------------|----------------------|--------------------------|-----------------------|--------------------------|
| <b>Error Rate<br/>(MSE)</b> | 9.6                  | 3.2                  | 11.8                     | 2.9                   | 4.4                      |

Note that the ground truth feature points that have been used for comparison in each database are obtained as follows:

- (1) For 3DFE and 4DFE databases, we used the associated feature points (N=83) (released from the databases) as ground truth;
- (2) For FRGC 2.0 and Eurecom databases, the ground truth feature points (N=83) were obtained through our manual annotation;
- (3) For BP4D-Spontaneous database, the ground truth feature points (N=83) were obtained by a semi-automatic method: First, we utilized the Kinect face tracking API [86] and modified it for 3D range data tracking. To modify the Kinect face tracking algorithm a multi-rendering is done to render the 3D range data in a suitable depth and RGB format. Second, the 2D coordinates are converted into model space to acquire the 3D landmarks. Finally, we manually correct the feature points that were erroneously detected or mapped.

As can be seen from Table 3 our proposed algorithm performs well on the 4DFE and BP4D databases. The relatively higher error rate on the 3DFE can be attributed to the low resolution of this database. Also, the relatively higher error rates on the FRGC and

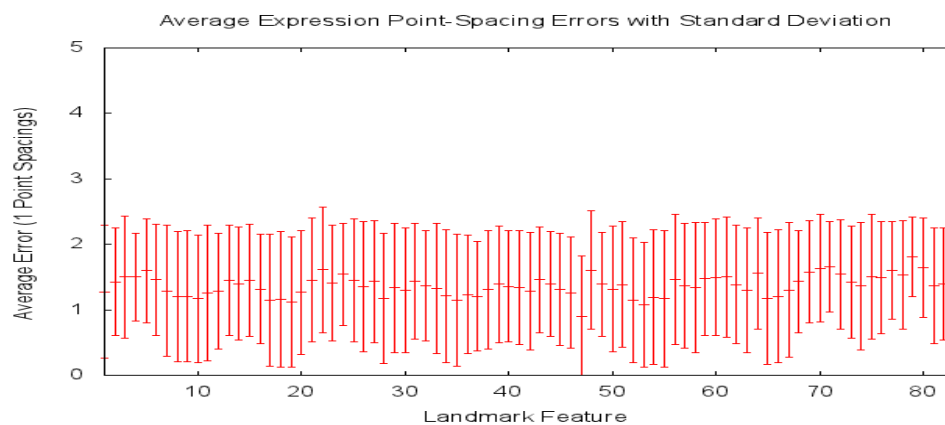
Eurecom databases can be attributed to the greater level of noise and holes in these datasets.

### **4.3 Performance Evaluation**

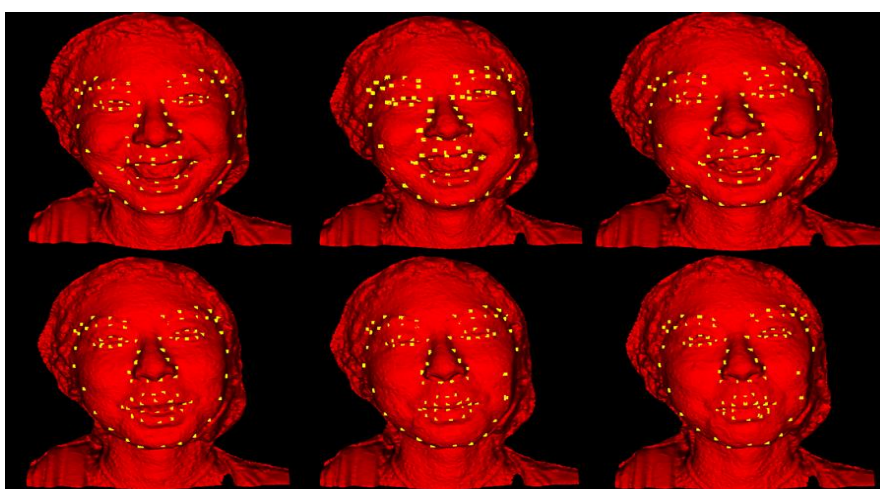
We also conducted three separate experiments on the BP4D database [136], which were split into the following categories: (1) expression segments, (2) rotations, and (3) occlusions/incomplete data. The details and errors statistics are detailed in the following sub-sections.

#### *4.3.1 Spontaneous Expression Segments*

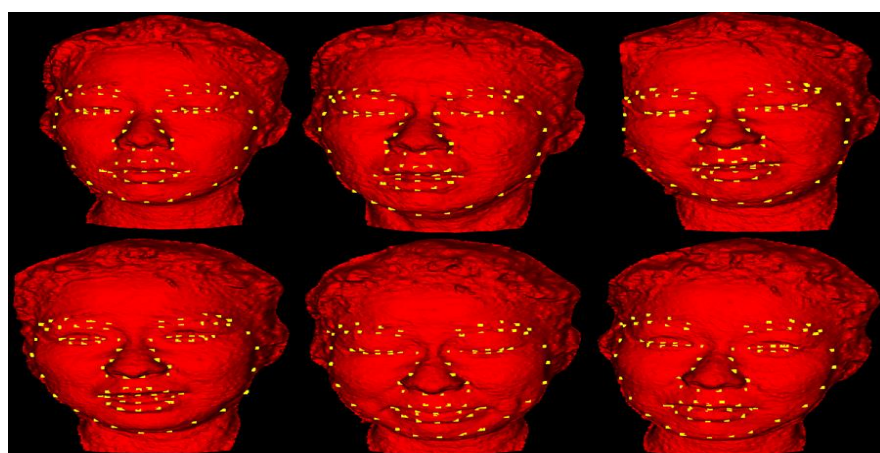
The BP4D [136] includes 8 tasks that are meant to elicit an emotional response. Those emotions include happiness, sadness, surprise, embarrassment, fear, pain, anger, and disgust. We test the accuracy of our algorithm on segments containing 8 explicit expressions and plotted the average error in point spacings. For all of the tested expression segments there is a MSE of 3.1, the average point spacing error for each landmark (where  $L=83$ ) along with the standard deviation can be seen in Figure 49. As can be seen from this figure, there is a small amount of variance between each of the 83 landmarks, with respect to their average error, thus showing it's robustness to different expressions. Two examples of spontaneous expressions can be seen in Figures 50 and 51.



**Figure 49.** Average error in point spacings of spontaneous expression sequences.



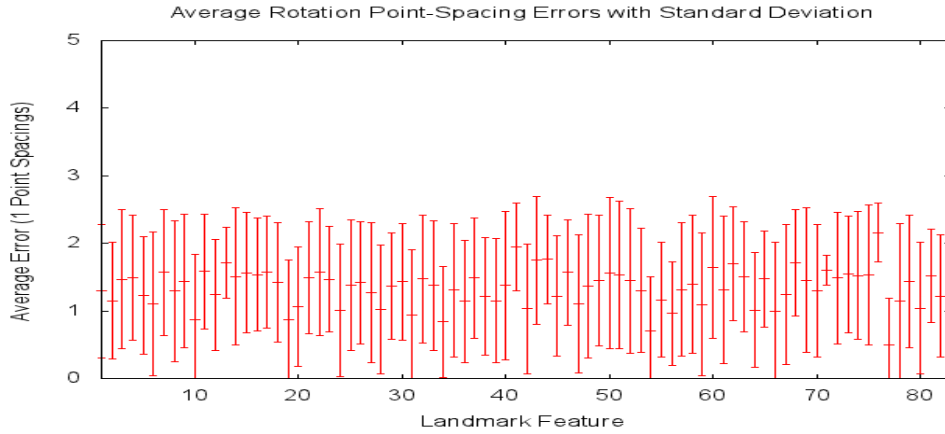
**Figure 50.** Example of a tracked sequence of a subject in a joyful condition when watching a film



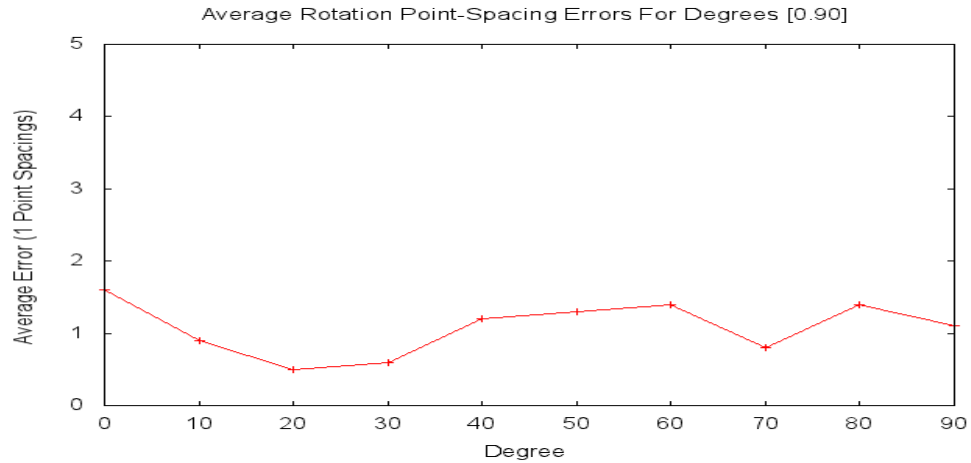
**Figure 51.** Example of a tracked sequence showing a startled emotion

#### 4.3.2 Rotation Sequences

We also tested our algorithm on sequences that contain rotations only. We are able to successfully fit rotations in the range of  $[-90, 90]$ . Figure 12 shows an example of rotations between  $[0, 90]$ . For all tested rotation sequences there is a MSE of 3.2. Figure 10 shows the average point spacings error along with the standard deviation for each of the,  $L=83$ , landmarks. As can be seen in Figure 10 the average error for rotations is fairly stable across all for all landmarks degrees, showing robustness to large rotations. Figure 52 show the average error, for all landmarks, across each of the degrees in the range  $[0, 90]$ .

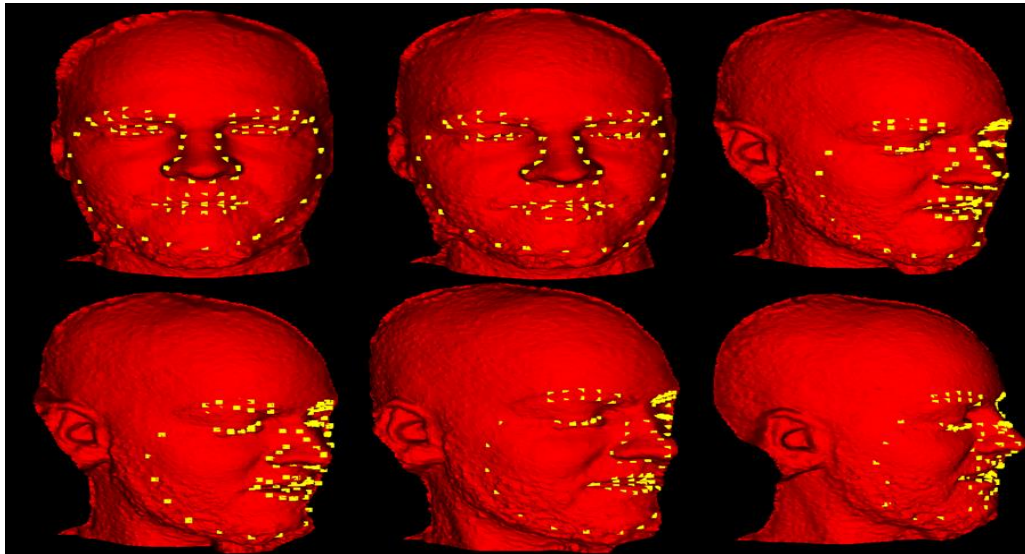


**Figure 52. Average error in point spacings for sequences displaying occlusions from rotations.**



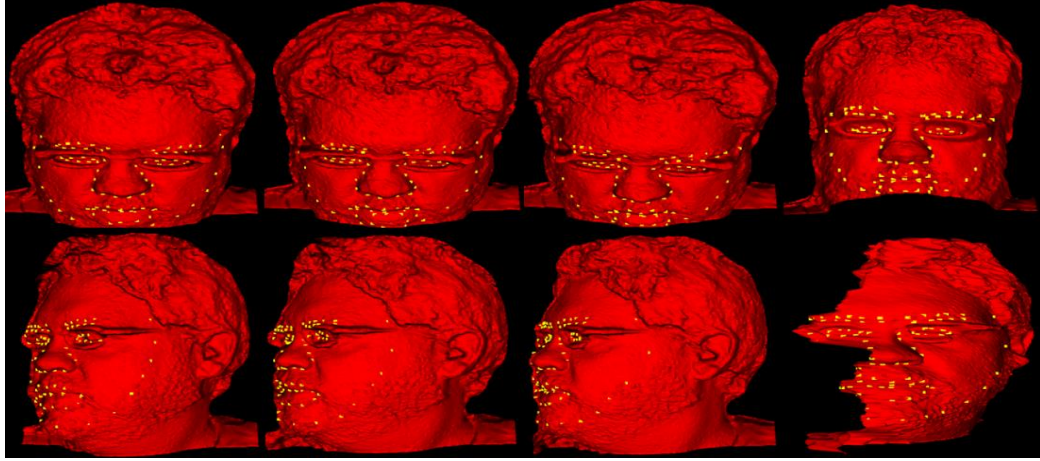
**Figure 53. Average point spacing error in relation to rotation degree.**

The results displayed in Figure 53 demonstrate the SI-SSM is robust to large rotations. For all rotations the average point spacing error is fairly consistent remaining under 2, with the largest error being 1.6 and the smallest error being 0.5. Two examples of sequences with large head rotations can be seen in Figures 54 and 55.



**Figure 54. Example of 4D data showing rotations from 0 degree to 90 degree**



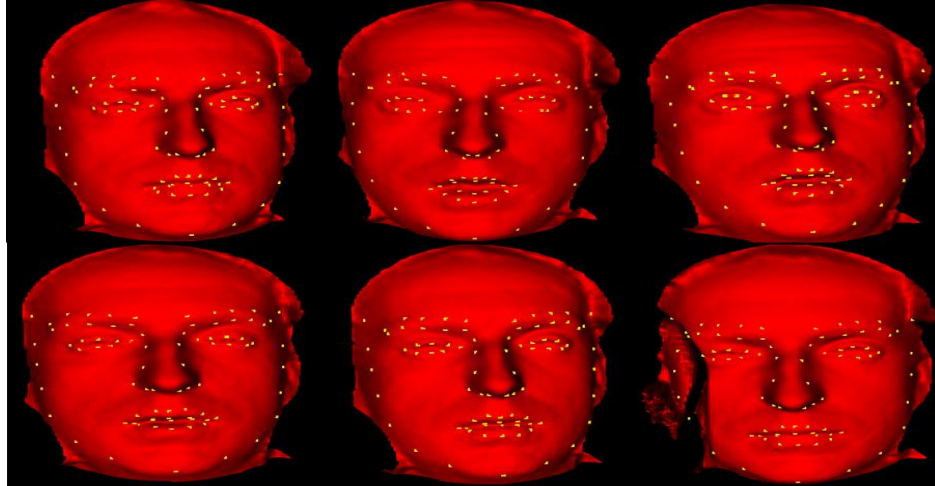


**Figure 55.** Sample frames displaying pitch and yaw pose estimations. Top Row (Pitch): -20, -23, -27; Bottom Row (Yaw): -37, -49, -51;. *Note: The last column is the same model from the previous column. The models (with eye glasses) are rotated to the frontal view so that they show the mesh deformations (or missing pieces) that this degree of pose causes.*

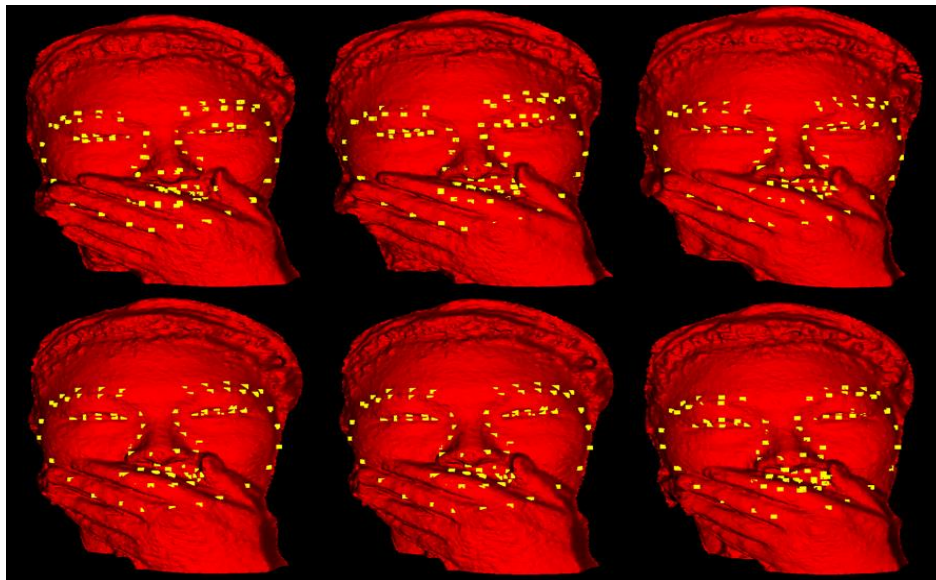
#### 4.3.3 Low Quality Sequences

We also tested the accuracy of our algorithm on sequences that contain low quality data.

We define low quality data as missing data from self-occlusion (eye glasses, hand in front of face, etc.), noisy data (beards, distorted patches caused by 3D data capture, etc.), and incomplete scans containing holes and isolated patches. Figures 56-58 illustrate these types of low quality data sequences.

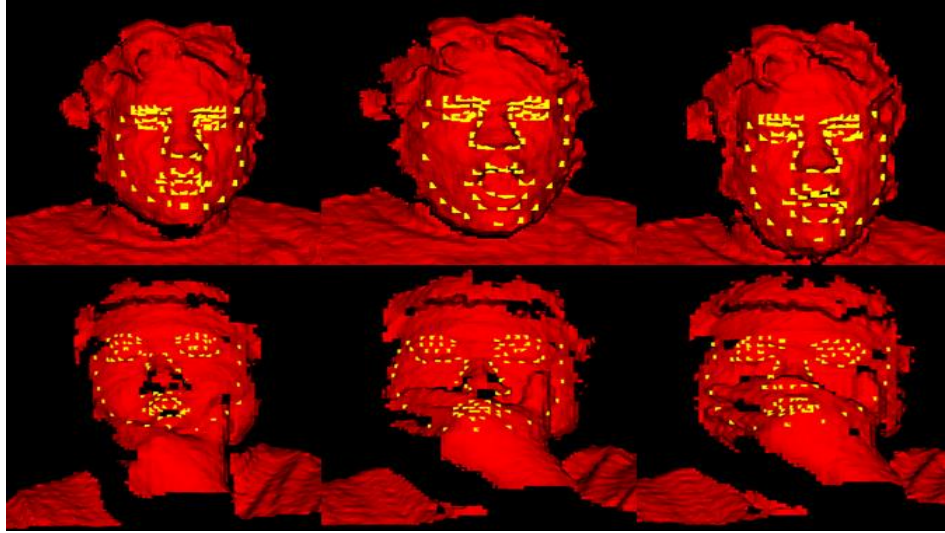


**Figure 56.** Tracked frames displaying a surprised expression. NOTE: the bottom right frame in the sequence is missing data on the side of the face, and the SI-SSM still fits to the missing data showing robustness to missing data.



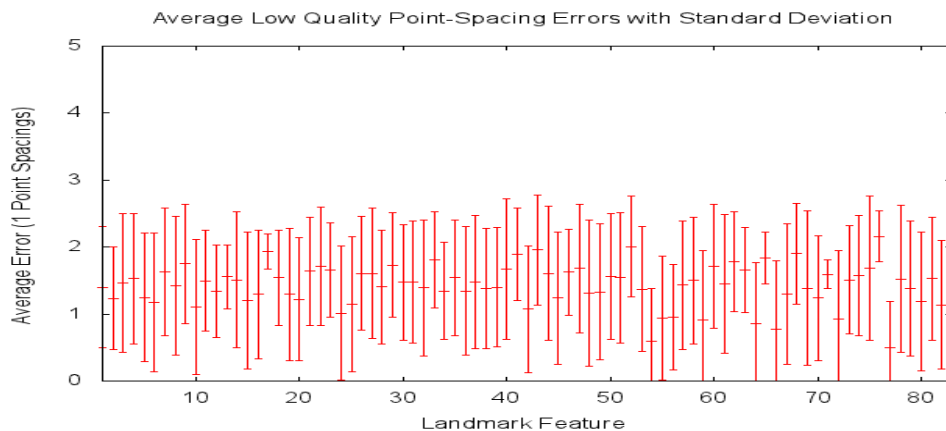
**Figure 57.** Tracked data showing robustness to occlusion.





**Figure 58. Tracked data with partial occlusions (robustness to noise and missing data).**

Our test results on low quality data shows the MSE error is 3.6. Figure 59 shows the average error in point spacings, along with the standard deviation for low quality data. While the MSE is slightly higher and the average errors show more variance than other tested data, the SI-SSM is still able to successfully fit to this data with a generally low error rate, showing robustness to low quality data.



**Figure 59. Average error in point spacings of low quality data.**

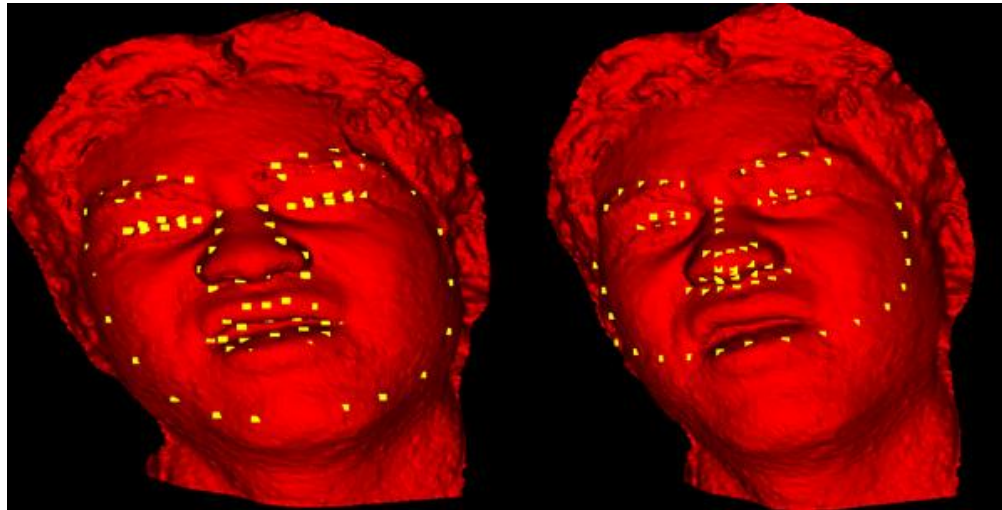
#### 4.4 Comparison to the State-of-the-Art

We also compared our SI-SSM algorithm, using 10% of the data for training and the rest for testing, against other state of the art algorithms. We compared our results against a 2D CLM mapped to 3D [134], TDSM [83] and Sun et al. [48] on the BU-4DFE and BP4D databases. For all comparisons we use the centroid landmark in each of the patches for comparisons, and report the MSE of the average point spacings. Note that the data tracked with the 2D CLM [134] only used 66 landmarks while we used 83 landmarks. In order to perform these experiments we selected the common sub-set of the two sets of landmarks, resulting in 49 landmarks for comparison. These landmarks comprise of the left and right eyes, nose, mouth and landmarks on the contour of the face. The 3D features mapped from 2D CLM have an error rate of 13.2 as compared to ours of 2.9. The high error rate of the 2D CLM based method can be attributed to frames where the tracking was lost and the method was unable to find a correct fit, as well as the mapping error from 2D to 3D. Figure 60 shows an example where the 2D CLM based method was unable to detect the correct landmarks while our SI-SSM was successful in detecting them. As can be seen from Table 16, which shows the results from these comparisons, our SI-SSM method outperforms the compared state of the art methods.

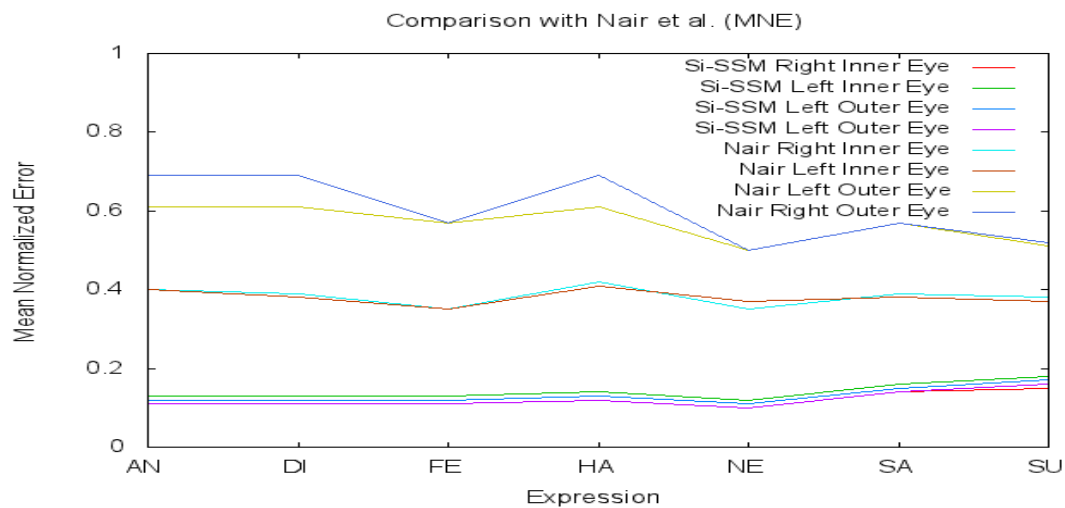
We also compare our work to Nair et al. [55] on the BU-3DFE database. For this experiment, we followed their detailed procedure. We selected four landmarks, the inner and outer corners of the left and right eyes, to compare to the ground truth. Figure 61 shows the mean normalized errors (MNE) for each of the four selected landmarks for all seven expressions in the database.

**Table 16. Comparison of SI-SSM, TDSM, Sun et al., and 2D CLM Mapped to 3D.**

| Comparison With State-of-the-Art |                                   |              |                   |            |
|----------------------------------|-----------------------------------|--------------|-------------------|------------|
|                                  | 3D Mapped From<br>2D CLM<br>[134] | TDSM<br>[83] | Sun et al<br>[48] | SI-SSM     |
| BU-4DFE                          | N/A                               | 3.7          | 6.3               | <b>3.2</b> |
| BP4D                             | 13.2                              | 4.0          | 7.2               | <b>2.9</b> |



**Figure 60. SI-SSM (left side), 3D mapped from 2D CLM (right side). NOTE: In the sequence that contains this frame showing a rotated head pose with a painful expression from BP4D-Spontaneous database, there are approximately 100 frames that the 2D CLM fails to correctly fit (similar to this figure) whereas the SI-SSM is successful.**



**Figure 61. MNE comparison to Nair et al. [55] for expressions in BU-3DFE.**

## 5. Applications

### 5.1 Posed and Spontaneous Facial Expression Classification

#### 5.1.1. Approach

To validate our proposed method, we apply it to facial expression classification problems for both posed expressions and spontaneous expressions, respectively. We take the component based approach for the classification. Given the tracked feature points, we can easily segment the facial model into several component regions, such as the eyes, nose and mouth. Fig. 38(a) shows an example of the resulting segmentation.

#### (i) 3D Component Feature Representation

To represent the 3D features, the same method was utilized as described in chapter 6. The details of this method can be seen (chapter 6) in sub-sections 4.3.1, 4.3.2, and 4.3.3. The next sub-section details a component-based spatial-temporal HMM Model.

#### (ii) Component-based Spatial-Temporal HMM Model

In order to determine a class of a certain expression, results from S-HMM and T-HMM are integrated as follows:

- (a) The expression class follows one of the results of S-HMM and T-HMM if both are the same.
- (b) The expression class follows the result of T-HMM if both are not the same, but the T-HMM has the more votes for a certain expression among six components than the votes of the other expressions from S-HMM.

(c) Vice versa, the expression class follows the result of S-HMM if both are not the same, but the S-HMM has the more votes for a certain expression among six frames than the votes of the other expressions from T-HMM.

(d) If none of above, the likelihoods (maximum probability) each individual expression from S-HMM and T-HMM are added, resulting six likelihoods (if six expressions). The one with highest likelihood is chosen as the recognized expression.

#### *5.1.2 Experiment results on face expression classification using spatial-temporal HHM*

The posed facial expression database (BU-4DFE) and spontaneous facial expression database (BP4D-Spontaneous) are used for experiments on face expression classification.

(i) Posed expressions: For training sequences of 4DFE, 1,200 sets of six consecutive frames were randomly chosen for training. Following the HMM training procedure ( $k = 6$ ), we generated the spatial HMM and temporal HMM for each expression. The recognition procedure is then applied to classify the expression of each input sequence ( $k = 6$ ). Based on the 10-fold cross validation approach, the six prototypic facial expressions are classified with an accuracy of approximately 92.3%.

(ii) Spontaneous expressions: For training sequences of BP4D-Spontaneous, 2,560 sets of six consecutive frames were randomly chosen for training. Similar to the above procedure, we generated the spatial HMM and temporal HMM for each expression. The recognition procedure is then applied to classify the expression of each input sequence ( $k$

= 6). Based on the 10-fold cross validation approach, the eight spontaneous facial expressions are classified with an accuracy of approximately 83.7%.

## **5.2 Pose Estimation**

Using the same approach as detailed in chapter 6, section 4.5, pose estimation is performed using the results from the SI-SSM detection. The comparisons show 2.53, 1.35, and 2.44 differences in degree across pitch, roll, and yaw respectively. Fig. 32 shows sample models displaying yaw, roll, and pitch with estimated poses.

## **6. Discussion**

In this chapter we have presented a novel method of detecting and tracking landmarks on 3D and 4D data using a shape index-based statistical shape model. The SI-SSM has been tested on five public 3D/4D face databases. The SI-SSM has shown robustness to rotations, occlusions, and low quality data, as well as superiority to the compared state of the art methods, given only the geometric information used. In a similar fashion to the ASSM algorithm, the SI-SSM also lends itself well to parallelism. Instead of fitting temporal data in the sequence, each of the patches in the model could be analyzed in parallel. This would give a massive increase in speed for the fitting process, as well as enabling a further study into finding the best patch size. This is due to allowing larger patches to be quickly and accurately fit to the input data.

## **Chapter 8**

### **Conclusion**

#### **1. Findings**

The research presented in this dissertation is aimed at studying the benefits of using 3D information for eye gaze estimation, face and sketch recognition, facial activity analysis, and feature tracking. Utilizing tracked 2D facial features, a new scale-space topographic modeling approach for modeling 3D facial appearance and eye sight directions has been presented giving promising results. Extending the idea of creating 3D data from 2D, a fusion-based face recognition method was detailed. Fusing multiple frames of a subject rotating their head, almost doubled the recognition rates under strong shadow, which is a challenging task. Next, using 2D sketch data and detected facial landmarks, a 3D sketch model was created. This 3D sketch model was then used for 3D face sketch recognition achieving a recognition rate of approximately 92%.

While the first half of the dissertation involved creating 3D data from 2D, the second half involves research into directly utilizing 3D range data. The first study into using explicit 3D data involves the construction of a new dynamic curvature based descriptor for facial activity analysis. The descriptor was validated in terms of determining neutral vs. non-neutral, multiple prototypic expressions, and their intensity levels. The last part of this dissertation studies 3D/4D feature tracking by proposing two new statistical-based models. The first uses the explicit shape of a 3D model to detect and track features in 3D

and 4D. The second proposed method makes use of both the global and local shapes of the 3D data to detect and track the 3D features. Both methods have been shown to outperform current state-of-the-art methods.

## **2. Applications**

The studies in this dissertation are applicable to a wide range of fields. The construction of 3D data from 2D presented in Chapter 2 has been validated in the use of eye gaze estimation. This method is applicable to others areas as well including expression analysis, 3D face recognition, and entertainment. This is also detailed in Chapter 4, where 3D sketch data is constructed from 2D sketch data for face sketch recognition. In Chapter 3, the presented fusion-based face recognition method has a wide range of uses from law-enforcement, government, and security. The methods detailed in this dissertation are also applicable to track features on non-face data such as hands [150].

In Chapters 5 and 6 the presented 3D feature detection and landmark methods are important to many fields. Feature detection is an important first step for many applications. These include face expression analysis, face recognition, video segmentation, subject/object verification and identification, and entertainment.

## **3. Limitations**

While the presented methods and evaluations show promising and exciting results, there are some limitations that need further research to alleviate. 3D data is widely known to be invariant to pose changes, however, the proposed method of constructing 2D to 3D data,



currently, only covers a small range of poses. This is, in part, due to some of the limitations of detecting and tracking features in 2D to construct the 3D models. This also extends to the 3D face sketch recognition presented in this dissertation. This method also requires off-line work which makes this method currently unfeasible for real-time model creation.

The results presented for the fusion-based face recognition method are very encouraging and show a strong case for the using fusion-based methods to increase recognition rates, however, the type of data used can impose a limitation. A sequence of rotated heads is not always going to be readily available, so this method may not be applicable in all scenarios. Further study into alternative, effective ways to gather pseudo-3D data from images is required.

The work presented in this dissertation on 3D facial activity analysis, while promising, also suffers from some limitations. Only one public database is tested on that includes 6 prototypic expressions, along with a neutral expression. This method needs to be tested and evaluated on the largest and most challenging 3D face databases available [136].

Chapters 6 and 7 present novel methods for detecting and tracking 3D features, and show an improvement over current state-of-the-art methods. Although the results are very encouraging there are also some limitations to these methods. Both methods that are presented are person-dependent requires a large number of statistical models to accurately detect and track landmarks on multiple subjects. Further study into a 3D

person-independent, statistical model is needed. Also, each of the models is linear in nature. This limits the number of “dynamic” expressions that can ultimately be modeled, causing some sequences to have sections that are unsuccessfully detected and tracked.

When using a shape index-based statistical shape model, the size of the patch is an important topic and one that needs to be studied further. If the patch size is too small the “true” shape of the local regions around each landmark will not be correct. However, too large of a patch and you will have “fake” features (e.g. not the real shape, or possibly the shape around a different feature to be modeled). The size of the patches also effects the time it takes to find the correct fit. When thinking in terms of real-time applications the optimal patch size would be one that balances accuracy of fit vs. speed at which the model is fit. This is a difficult problem to solve, as it seems natural that finding the correct patch size would be task/person dependent.

#### **4. Discussion and Future Work**

Each of the topics covered in this dissertation are open questions within a vibrant community of scholars. The results presented compare favorably or better to the current-state-of-the-art. The evaluations presented in this dissertation validate the efficacy of each of the proposed methods. With the wide range of applicable fields, as discussed in section 2 of this chapter, the methods are important topics that require further study and evaluation.

The work presented in this dissertation lends itself well to future work by allowing a combination of multiple topics and algorithms. Face sketch recognition is a very important field that requires further investigation into. While the 3D face sketch presented shows encouraging results, there are many open questions that still need to be answered including:

- (1) Can 3D face sketch recognition out-perform 2D?
- (2) Will a fusion-based, or multi-modal-based approach increase recognition rates?
- (3) What are the most important features for face sketch recognition? Are they similar to image-based face recognition?
- (4) Are similar methods used for image-based face recognition sufficient enough for face sketch recognition, or are new, innovative solutions required for optimal recognition rates?

These questions help drive the motivation for the future work of this dissertation. I will investigate the above questions, and more, by combining statistical model-based learning (e.g. ASSM, SI-SSM), 3D face sketch data, and a fusion/multi-modal based approach to face sketch recognition. The use of a generic reference model to create the 3D face sketches has the advantage of all 3D mesh models are aligned to the same reference frame. This allows for easy registration of multiple meshes, which is a great fit for statistical model-based approaches. Utilizing this approach, some statistical combinations, of the following features, can be used for face sketch recognition; (1) explicit 3D shape of the mesh model; (2) sketch “texture” information; and (3) shape-

index values of deformed reference mesh. Due to the novelty of the presented 3D face sketch approach; this type of work would be the first of its kind in 3D face sketch recognition.

Using a statistical model of variance for 3D face sketch data would also allow for the creation of a real-time system that would allow a forensic artist to create/match 3D face sketches from eye witness statements. This could be an invaluable tool in helping law enforcement apprehend possible suspects. Given eye-witness statements regarding a suspect, the forensic artist would be able to use the system to create (deform reference mesh) 3D characteristics of the subject using a supplied user interface tool. Once the 3D representation of the suspect is created, the software would allow the user to search a database of possible 2D and/or 3D image and sketch mugshots.

There are various types of research problems when dealing with sketch recognition. From text to sketch, image to sketch, and sketch to image to name a few. These are all challenging fields that present their own unique set of problems. In creating a real-time system that can create 3D sketch models these types need to be factored in. Each of them are going to product different resulting sketches, as well as their own biases towards the data (i.e. different sketches can look very different). An important question to ask is “How can I infer data from on type to another?” Answering this question will require new and innovative algorithms to handle 3D data. A possible solution, and future work for this dissertation, involves the use of different shape descriptors to model the data. While shape index is an intuitive to describe shape, it may not be the best/only approach

to this problem. 3D edge information and temporal shape information can be used separately or combined with the available shape index data. This can give us a more accurate representation of the data that would allow us to infer from multiple data types.

Another major issue when constructing this type of system is the length of time it takes to successfully model the 3D sketch data. Incorporating a parallel algorithms could help alleviate many of the speed issues. This begs the question: “How can we leverage these parallel algorithms to efficiently and accurately create 3D sketch data?” One possibility is the segmenting of the 3D data, and “splicing” the model together. Each of the slices of the 3D sketch data can be analyzed in parallel, and then combined to create a final 3D sketch model. Parallel algorithms are a natural and intuitive solution to creating real-time 3D sketch data, and one of the major problems to be studied from this dissertation.

There are some challenges in the creation of this type of system. First, a more automatic/intuitive way of generating the initial features for the 3D sketch model would be required. Secondly, a sufficiently large sketch database would be needed for creation of the 3D statistical model. These challenges, as well as the questions that still remain in both 2D and 3D face sketch recognition will be the main focus of my future research.

## References

- [1] S. Amarnag, R. S. Kumaran, and J. N. Gowdy, "Real time eye tracking for human computer interfaces", *Proc. of IEEE International Conference on Multimedia and Expo*, 2003.
- [2] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23 no. 6. pp. 681-685, June 2001.
- [3] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models", *Image and Vision Computing*, vol. 23, no. 11, pp. 1080-1093, 2005.
- [4] T. Ishikawa, S. Baker, I. Matthews, T. Kanade, "Passive driver gaze tracking with active appearance models", *Proceedings of the 11<sup>th</sup> World Congress on Intelligent Transportation Systems*, 2004.
- [5] Q. Ji, H. Wechsler, A. Duchowski, and M. Flickner, "Eye detection and tracking", *Computer Vision and Image Understanding*, 98(1), 2005.
- [6] Di3D Inc. [www.di3d.com](http://www.di3d.com)
- [7] J. Magee, M. Scott, B. Waber, and M. Betke, "EyeKeys: a real-time vision interface on gaze detection from a low-grade video camera", *Computer Vision and Pattern Recognition Workshop*, 2004.
- [8] T. Takegami, T. Gotoh, S. Kagei, and R. Minamikawa, "A Hough based eye direction detection algorithm without on- cite calibration", *IEEE Trans. on PAMI*, 20(10): 2001.
- [9] D. Terzopoulos and K. waters. "Analysis-synthesis of face image seq. using physical-anatomical model", *IEEE Transactions on. Pattern Analysis and Machine Intelligence*, 1993.
- [10] O. Trier, T. Text, and A.K. Jain. "Recognition of digits in hydrographic maps: binary vs topographic analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 1997.
- [11] J. Wang, L. Yin, and J. Moore, "Using geometric property of topographic manifold to detect and track eyes for human computer interaction", *ACM Transactions on Multimedia Computing Commuication Applications*, 3(4): 1-19, 2007.

- [12] L. Yin and K. Weiss. "Generating 3d views of facial expressions from frontal face video based on topographic analysis", In *ACM Multimedia 2004*, p360-363.
- [13] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A high-resolution 3D dynamic facial expression database", *IEEE International Conference on Face and Gesture Recognition*, 2008.
- [14] P. Phillips, W. Scruggs, A. O'Toole, P. Flynn, K. Bowyer, C. Schott, and M. Sharpe. "FRVT 2006 and ICE 2006 large-scale results", *National Institute of Standards and Technology, Internal Report 7408*, 2007.
- [15] K. W. Bowyer, K. Chang, and P. J. Flynn. "A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition," *Computer Vision and Image Understanding*, 101(1):1-15, 2006.
- [16] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video", *Computer Vision and Image Understanding*, 91, pp. 214-245, 2003.
- [17] R. Chellappa and S. Zhou, "Face tracking and recognition from video", *Handbook of Face Recognition*, S. Li and A. K. Jain (Eds.), Springer, 2004.
- [18] K. Lee, J. Ho, M. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," *IEEE Computer Vision and Pattern Recognition*, pp. 313-320, 2003.
- [19] R. Singh, M. Vatsa, A. Ross, and A. Noore, "A mosaicing scheme for pose invariant face recognition", *IEEE Transactions on Systems, Man, and Cybernetics Part B*, 37(5):1212-1225, 2007.
- [20] R. Gross, S. Baker, I. Matthews, and T. Kanade, "Face recognition across pose and illumination", *Handbook of Face Recognition*, S. Li and A. K. Jain (Eds.), Springer, 2004.
- [21] T. C. Faltemier, K. W. Bowyer, and P. J. Flynn, "Using multi-instance enrollment to improve performance 3D face recognition", *Computer Vision and Image Understanding*, 112(2):114-125, 2008.
- [22] T. Kim and J. Kittler, "Design and fusion of pose-invariant face-identification experts", *IEEE Transactions. on Circuits & Systems for Video Technology*, 16(9), pp. 1096-1106, 2006.
- [23] Y. Zhang and A. Martinez, "A weighted probabilistic approach to face recognition from multiple images and video sequences", *Image & Vision Computing* 24(6):626-638, 2006.

- [24] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker, "Multi-PIE", *IEEE Face and Gesture Recognition*, 2008.
- [25] X. Liu and T. Chen, "Video-based face recognition using adaptive hidden Markov models", *IEEE Computer Vision and Pattern Recognition*, 2003.
- [26] D. Thomas, K. W. Bowyer, and P. J. Flynn, "Strategies for improving face recognition from video", *Advances in Biometrics: Sensors, Systems and Algorithms*, N. Ratha and V. Govindaraju, editors, Springer, 2007.
- [27] S. Canavan, M. Kozak, Y. Zhang, S. Sullins, M. Shreve, and D. Goldgof, "Face recognition by multi-frame fusion of rotating heads in videos", *IEEE Conference on Biometrics: Theory, Applications, and Systems 2007*.
- [28] M. Turk, and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, (3)1, pp. 71-86, 1991.
- [29] [www.cs.colostate.edu/evalfacerec/](http://www.cs.colostate.edu/evalfacerec/)
- [30] M. Savvides, B. Vijaya Kumar, and P. Khosla, "'Corefaces'- Robust Shift Invariant PCA based correlation filter for illumination tolerant face recognition", In *IEEE Computer Vision and Pattern Recognition 2004* pp. 834-841.
- [31] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato, "A 3D facial expression database for facial behavior research", *IEEE Face and Gesture Recognition*, 2006.
- [32] M. Reale, "Head pose determination, tracking, and gaze estimation for HCI", *Master thesis*, Binghamton Univ., 2009.
- [33] L. Gibson, "*Forensic Art Essentials: A Manual for Law Enforcement Artists*", Academic Press, 2007.
- [34] C. D. Frowd, V. Bruce, A. McIntyre, D. Ross, and S. Fields, Y. Plenderleith, and P. Hancock, "Implementing holistic dimensions for a facial composite system", *Journal of Multimedia*, 1(3), pp. 42-51, 2006.
- [35] P. Yuen, and C. Man, "Human face image searching system using sketches", *IEEE Transactions on Systems, Man, and Cybernetics Part A*, 37(4), 2007.
- [36] C. Frowd, D. McQuiston-Surrett, S. Anandaciva, and C. Ireland, and P. Hancock, "An evaluation of US systems for facial composite production", *Ergonomics*, 50:562-585, 2007.
- [37] B. Flare, Z. Li, and A. Jain, Matching forensic sketches to mug shot photos, *IEEE Transactions on Pattern analysis and Machine Intelligence*, 33(3):639-646, March, 2011



- [38] X. Wang and X. Tang, "Face Photo-Sketch Synthesis and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11), Nov. 2009.
- [39] X. Gao, J. Zhong, J. Li, and C. Tian, "Face Sketch Synthesis Algorithm Based on E-HMM and Selective Ensemble," *IEEE Transactions Circuits and Systems for Video Technology*, 18(4):487-496, 2008.
- [40] B. Xiao, X. Gao, D. Tao, and X. Li, "A New Approach for Face Recognition by Sketches in Photos," *Signal Processing*, 89(8):1576-1588, 2009.
- [41] Z. Xu, H. Chen, S. Zhu, and J. Luo, "A hierarchical compositional model for face representation and sketching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):955-969, June, 2008
- [42] L. Clarke, M. Chen, and B. Mora, "Automatic generation of 3D caricatures based on artistic deformation styles", *IEEE Transactions on Visualization & Computer Graphics*, 17(6), 2011.
- [43] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey", *ACM Computing Surveys*, 35(4), pp. 399-458, 2003.
- [44] S. Z. Li and A. K. Jain (editors), "*Handbook of Face Recognition*", Springer, 2005.
- [45] P. Flynn and A. Jain, "Surface classification: Hypothesis testing and parameter estimation", *IEEE Computer Vision and Pattern Recognition* 1988.
- [46] H. Nizami, J. Adkins-Hill, Y. Zhang, J. Sullins, C. McCullough, S. Canavan, and L. Yin, "A biometric database with rotating head videos and hand-drawn face sketches", *IEEE 3<sup>rd</sup> International Conference on Biometrics: Theory, Applications, and Systems*, 2009.
- [47] J. Wang, X. Wei, L. Yin, and Y. Sun, "3D facial expression recognition based on primitive surface feature distribution", In *IEEE Computer Vision and Pattern Recognition* 2006.
- [48] Y. Sun, X. Chen, M. Rosato, and L. Yin, "Tracking vertex flow and model adaptation for 3D spatio-temporal face analysis", *IEEE Transactions on Systems, Man, and Cybernetics Part A*. 40(3), 2010.
- [49] J. Kittler, A. Hilton, M. Hamouz, and J. Illingworth, "3D assisted face recognition: A survey of 3D imaging, modeling and recognition approaches", In

*Computer Vision and Pattern Recognition Workshop on Advanced 3D Imaging for Safety and Security*, 2005.

[50] X. Lu, A. Jain, and D. Colbry. “Matching 2.5D face scans to 3D models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):31–43, Jan. 2006.

[51] F. Steinke, B. Scholkopf, and V. Blanz, “Learning dense 3d Correspondence”, *Proceedings of 20th Annual Conference on Neural Information Processing Systems*, 2006.

[52] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 2002, p971-987.

[53] P. Besl and N. McKay, “A method for registration of 3D shapes”, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[54] P. Dalal, B. C. Munsell, S. Wang, J. Tang, and K. Oliver, “A fast 3d correspondence method for statistical shape modeling”, *IEEE Computer Vision and Pattern Recognition*, 2007.

[55] P. Nair, and A. Cavallaro, “3-D face detection, landmark localization, and registration using a point distribution model”, *IEEE Transactions on Multimedia*, 11(4):611-623, 2009.

[56] P. Perakis, G. Passalis, T. Theoharis, G. Toderici, and I.A. Kakadiaris, “Partial matching of interpose 3D facial data for face recognition”, *Proceedings of 3rd IEEE Biometrics: Theory, Applications, and Systems*, pp. 439-466.

[57] T. Cootes, C. Taylor, D. Cooper, and J. Graham. “Active shape model-their training and application”, *Computer Vision and Image Understanding*, 61:18-23, 1995.

[58] V. Blanz and T. Vetter. “A Morphable model for the synthesis of 3D faces”, *Computer Graphics Proceedings of Special Interest Group on Graphics and Interactive Techniques*, 1999.

[59] X. Lu and A. K. Jain, “Automatic feature extraction for multiview 3D face recognition”, *Proceedings of IEEE Conference on Automatic Face and Gesture Recognition*, Southampton, UK, 2006, pp. 585-590.

[60] Z. Zeng, M. Pantic, G. Roisman, T. Huang, “A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39-58, 2009.

- [61] C. Dorai, A. Jain, “Cosmosa representation scheme for 3D free-form objects”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, No. 10, 1997
- [62] J. Koenderink and A. van Doorn, “Surface shape and curvature scales”, *Image and Vision Computing*, Vol. 10, No. 8, Oct. 1992, p557-564
- [63] U. H.-G. Krebel. “Pairwise classification and support vector machines”, *Advances in Kernel Methods: Support Vector Learning*, pages 255–268. The MIT Press, Cambridge, MA, 1999
- [64] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, “Large margin DAGs for multiclass classification”, *Advances in Neural Information Processing Systems 12*, pages 547–553. The MIT Press, Cambridge, MA, 2000.
- [65] J. Weston and C. Watkins, “Multi-class support vector machines”, *Technical Report CSD-TR-98-04*, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.
- [66] K. Crammer and Y. Singer, “On the Algorithmic Implementation of Multi-class SVMs”, *Journal of Machine Learning Research*, 2:265-292, 2001.
- [67] M. Lyons, J. Budynek, and S. Akamatsu, “Automatic classification of single facial images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:1357–1362, 1999.
- [68] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang., “Facial expression recognition from video sequences: temporal and static modeling”, *Computer Vision and Image Understanding*, 91(1), 2003.
- [69] Y. Sun and L. Yin, “Facial expression recognition based on 3d dynamic range model sequences”, In *10th European Conference on Computer Vision*, Marseille, France, October 2008.
- [70] D. Cosker, E. Krumhuber, A. Hilton, “A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling”, *IEEE International Conference on Computer Vision*, (2011) 2296-2303.
- [71] G. Stratou, A. Ghosh, P. Debevec, L.-P. Morency, “Effect of Illumination on Automatic Expression Recognition: A Novel 3D Relightable Facial Database”, *9th International Conference on Automatic Face and Gesture Recognition*, Santa Barbara, California, 2011.
- [72] A. Savran, N. Alyuz, H. Dibeklioglu, O. Celiktutan, B. Gokberk, B. Sankur, L. Akarun, “Bosphorus database for 3D face analysis”, *Proceedings First Cooperation in Science and Technology 2101 Workshop on Biometrics and Identity Management*, Roskilde University, Denmark, 2008, pp. 47–56.

- [73] Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic and Daniel Rueckert, “A Dynamic Approach to the Recognition of 3D Facial Expressions and Their Temporal Models, Special Session: 3D facial behaviour analysis and understanding”, *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.
- [74] T. Fang, X. Zhao, O. Ocegueda, S.K. Shah and I.A. Kakadiaris, “3D Facial Expression Recognition: A Perspective on Promises and Challenges”, *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.
- [75] V. Le, H. Tang and T. Huang, “Expression Recognition from 3D Dynamic Faces using Robust Spatio-temporal Shape Features”, *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.
- [76] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, “Dynamic textures”, *International Journal of Computer Vision*, 51(2):91–109, 2003.
- [77] S. Koelstra, M. Pantic, and I. Patras, “A dynamic texture-based approach to recognition of facial actions and their temporal models” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11) 1940–1954, 2010.
- [78] G. Zhao and M. Pietikainen, “Dynamic texture recognition using local binary patterns with an application to facial expressions”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(29), 915-928, 2007.
- [79] M. Valstar, M. Pantic, and I. Patras, “Motion history for facial action detection in video”, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 635–640, 2004.
- [80] D. Chetverikov and R. Peteri, “A brief survey of dynamic texture description and recognition”, *4th Conference on Computer Recognition Systems*, pages 17–26, 2005.
- [81] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F. De la Torre Frade, and J. Cohn, “AAM derived face representations for robust facial action recognition”, *IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.
- [82] K. Ramnath, S. Koterba, J. Xiao, C. Hu, I. Matthews, S. Baker, J. Cohn, and T. Kanade, “Multi-view AAM fitting and construction”. *International Journal of Computer Vision*, 2007.
- [83] S. Canavan and L. Yin, “3D feature tracking using a deformable shape model”, Technical Report, Binghamton University, Feb., 2012.

- [84] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, J. Worek, "Overview of the face recognition grand challenge," *IEEE Computer Vision and Pattern Recognition*, 2005.
- [85] X. Zhang, L. Yin, J. Cohn et al, "A high resolution spontaneous 3D dynamic facial expression database," *Face and Gesture Recognition*, 2013.
- [86] Kinect for Windows. Microsoft Corporation, Redmond WA.
- [87] T. Hunynh, R. Min, J-L. Dugelay, "An Efficient LBP-Based Descriptor for Facial Depth Images applied to Gender Recognition using RGB-D Face Data", *Asian Conference on Computer Vision Workshop on Computer Vision with Local Binary Pattern Variants*, 2012.
- [88] I. A. Kakadiaris, G. Passalis, G. Toderick, M. N. Murtuza, Y. Lu, N. Karampatzikas, T. Theohari, "Three-dimensional face rec. in the presence of facial expressions: An annotated deformable model approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
- [89] M. Segundo, C. Queirolo, O.R.P. Bellon, L.Silva, "Automatic 3D facial segmentation and landmark detection", *International Conference on Image Analysis and Processing* 2007.
- [90] T. Cootes, et al. "Active shape models: Evaluation of a multi-res. approach method for improving the image search," *British Machine Vision Conference*, 1994.
- [91] M. De Bruijne et al., "Adapting active shape models for 3D segmentation of tubular structures in medical images," *Information Processing in Medical Imaging*, 2003.
- [92] G. Fanelli, M. Dantone, and L.V. Gool, "Real time 3D face alignment with random forests-based active appearance models," *Face and Gesture Recognition*, 2013.
- [93] X. Zhao et al, "Accurate landmarking of three-dimensional facial data in the presence of facial expressions and occlusions using a three-dimensional statistical facial feature model," *IEEE Transactions on Systems, Man, and Cybernetics Part B*, 41(5): 1417-1428, 2011.
- [94] P.Guan, Y.Yu, and L. Zhang, "A novel facial feature point localization method on 3D faces," *International Conference on Image Processing* 2007.
- [95] T. Weise, S. Busaziz, H. Li, and M. Pauly, "Real-time performance-based facial animation," *ACM Transactions on Graphics*, 2011.

- [96] S. Canavan, X. Zhang, L. Yin, "Fitting and tracking 3D/4D facial data using a temporal deformable shape model," *International Conference on Multimedia and Expo 2013*.
- [97] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of IEEE*, 77(2), 1989.
- [98] E. Ong, and R. Bowden, "Robust facial feature Tracking using shape- constrained multiresolution-selected linear predictors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1844-1859, 2011.
- [99] T.F.Cootes, K. Walker, and C.J. Taylor, "View-based active appearance models," *Image and Vision Computing* 20(9): 657-664, 2002
- [100] P. Szeptycki, M. Ardabilian, and L. Chen, "A coarse-to-fine curvature analysis-based rotation invariant 3D face Landmarking," *Biometrics: Theory, Applications and Systems* 2009.
- [101] H. Dibeklioglu, A.A.Salah, and L. Akarun, "3D facial landmarking under expression, pose, and occlusion variations," *Biometrics: Theory, Applications and Systems* 2008.
- [102] G.J.Edwards, T.F.Cootes, and C.J.Taylor, "Advances in active appearance models," *International Conference on Computer Vision*, 1999.
- [103] E. Munoz, J.M. Buenaposada, and L. Baumela, "A direct approach for efficiently tracking with 3D Morphable Models," *International Conference on Computer Vision*, 2009.
- [104] D. Zhou, D. Petrovska-Delacretaz, and B. Dorizzi, "3D active shape model for automatic facial landmark location trained with automatically generated landmark points," *International Conference on Pattern Recognition*, 2010.
- [105] T. Baltrusaitis, P.Robinson, and L. Morency, "3D constrained local model for rigid and non-rigid facial tracking," *Computer Vision and Pattern Recognition*, 2012.
- [106] J. Shi, and C. Tomasi, "Good feature to track," *Computer Vision and Pattern Recognition*, 1994.
- [107] X. Zhu, and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," *Computer Vision and Pattern Recognition*, 2012.
- [108] A. Salazar, S. Wuhler, C. Shu, and F. Prieto, "Fully automatic expression-invariant face correspondence," *Technical report*, 2012.

- [109] H. Li, T. Weise, and M. Pauly, "Example-based facial rigging," *ACM Transactions on Graphics*, 2010.
- [110] A. Patel, and W. Smith, "3D morphable models revisited," *International Conference on Computer Vision*, 2009.
- [111] A. Balan and M. Black, "The naked truth: estimating body shape under clothing," *European Conference on Computer Vision*, 2008.
- [112] A. Johnson and M. Hebert, "Recognizing objects by matching oriented points," *International Conference on Computer Vision*, 1997.
- [113] A. Brunton, A. Salazar, T. Bolkart, and S. Wuhler, "Comparative analysis of statistical shape spaces," *Technical Report*, 2012.
- [114] K. Chang, K.W.Bowyer, and P.Flynn, "Multiple nose region matching for 3D face recognition under varying facial expression," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2006.
- [115] L.Dekker, I. Douros, B.F. Bruxton, P.Treleaven, "Building symbolic information for 3D human body modeling from range data," *3D Digital Imaging and Modeling*, 1999.
- [116] J.D'Hose, J.Colineau, C. Bichon, and B. Dorizzi, "Precise localization of landmarks on 3D faces using gabor wavelets," *Biometrics: Theory, Applications and Systems*, 2007.
- [117] T. Whitmarsh, R.C. Veltkamp, M. Spagnuolo, S. Marini, and F. Haar, "Landmark detection on 3D faces by facial model registration," *Symposium on Shapes and Semantics*, 2006.
- [118] A.S. Mian, M. Bennamoun, R. Owens, "Keypoint detection and local feature matching for textured 3D face recognition," *International Journal of Computer Vision*, Vol. 79, pp. 1-12, 2008.
- [119] A. Salah, and L. Akarun, "3D facial feature localization for registration," *Multimedia Content, Classification and Security*, 2006.
- [120] N. Alyuz, B. Gokberk, H. Dibeklioglu, A. Savran, A. Salah, L. Akarun, and B. Sankur, "3D face recognition benchmarks on the bosporus database with focus on facial expressions," *Biometrics and Identity Management*, 2008.
- [121] S. Gupta, M. Markey, A. Bovik, "Anthropometric 3D face recognition," *International Journal of Computer Vision*, Vol. 90, pp. 331-349, 2010

- [122] P. Liu, M. Reale, and L. Yin, "3D head pose estimation on scene flow and generic head model," *International Conference and Multimedia Expo*, 2012.
- [123] C. Creusot, N. Pears, and J. Austin, "A machine-learning approach to keypoint detection and Landmarking on 3D meshes," *International Journal of Computer Vision*, 102(1-3): 146-179, 2013.
- [124] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar, "Localizing Parts of Faces Using a Consensus of Exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (35)12:2930-2940, 2013.
- [125] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," *International Journal of Computer Vision*, 107:177-190, 2014.
- [126] T.F.Cootes, K. Walker, and C.J. Taylor, "View-based active appearance models," *Image and Vision Computing*, 20(9): 657-664, 2002.
- [127] D. Cristinacce and T. Cootes, "Feature Detection and Tracking with Constrained Local Models," *British Machine Vision Conference*, 2(5), 2006.
- [128] L. Gu and T. Kanade, "A generative shape regularization model for robust face alignment," *European Conference on Computer Vision*, 2008.
- [129] L. Jeni, A. Lorinca, T. Nagy, Z. Plotai, J. Sebok, Z. Szabo, D. Takacs, "3D shape estimation in video sequences provides high precision evaluation of facial expressions," *Image and Vision Computing*, (30)10: 785-795, 2012.
- [130] J. Lewis, "Fast Template Matching," *Vision Interface*, 1995.
- [131] I. Matthews and S. Baker, "Active Appearance Models Revisited," *International Journal Computer Vision*, (60):135-164, 204.
- [132] R. Min, N. Kose, and J Dugelay, "KinectFaceDB: A Kinect Database for Face Recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, 44(11): 1534-1548, 2014.
- [133] P. Perakis, G. Passalis, T. Theoharis, and I.A. Kakadiaris, "3D Facial Landmark Detection Under Large Yaw and Expression Variations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1552-1564, 2013.
- [134] J.M. Saragih, S. Lucey, J.F. Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal Computer Vision*, 91(2), 2011.
- [135] X. Xiong and F. De la Torre, "Supervised Descent Method and its Applications to Face Alignment," *Computer Vision and Pattern Recognition*, 2013.



- [136] X. Zhang, L. Yin, J. Cohn S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. Girard, "BP4D-Spontaneous: A high resolution 3D dynamic facial expression database," *Image and Vision Computing*, 10, 2014.
- [137] I. A. Kakadiaris, G. Passalis, G. Toderick, M. N. Murtuza, Y. Lu, N. Karampatzikas, T. Theohari, "Three-dimensional face rec. in the presence of facial expressions: An annotated deformable model approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4): 640-649, 2007.
- [138] T. Fang, X. Zhao, O. Ocequeda, S. Shan, and I. Kakadiaris, "3D/4D facial expression analysis: an advanced annotated face model approach," *Image and Vision Computing*, 30(10):738-749, 2012
- [139] H. Li, D. Huang, JM. Morvan, Y. Wang, and L. Chen, "Towards 3D Face Recognition in the Real: A Registration-Free Approach using Fine-Grained Matching of 3D Keypoints," *International Journal of Computer Vision*, pp. 1-14, 2014.
- [140] J. Sun, D. Huang, Y. Wang, and L. Chen, "A Coarse-to-Fine Approach to Robust 3D Facial Landmarking via Curvature Analysis and Active Normal Model," *IEEE International Joint Conference on Biometrics*, 2014.
- [141] G. Sandbach, S. Zafeiriou, M. Pantic, L. Yin, "Static and dynamic 3D facial expression recognition: A comprehensive survey," *Image and Vision Computing*, 30 (10) (2012) 683-697.
- [142] H. Soyel, H. Demirel, "Facial expression recognition using 3D facial feature distances," *Image Analysis and Recognition* , p831-838, 2007.
- [143] H. Tang, T. S. Huang, "3d facial expression recognition based on properties of line segments connecting facial feature points," *8th IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, 2008.
- [144] S. Berretti, A. D. Bimbo, P. Pala, B. B. Amor, M. Daoudi, "A set of selected sift features for 3D facial expression recognition," *20th International Conference on Pattern Recognition*, pp. 4125-4128, 2010.
- [145] S. Canavan and L. Yin, "Dynamic Face Appearance Modeling and Sight Direction Estimation Based on Local Region Tracking and Scale-Space Topo-Representation," *International Conference on Multimedia and Expo*, 2009.
- [146] S. Canavan, B. Johnson, M. Reale, Y. Zhang, L. Yin, and J. R. Sullins, "Evaluation of Multi-Frame Fusion Based Face Classification Under Shadow," *International Conference on Pattern Recognition*, 2010.

- [147] S. Canavan, X. Zhang, L. Yin, and Y. Zhang, "3D Face Sketch Modeling and Assessment for Component Based Face Recognition," *International Joint Conference on Biometrics*, 2011.
- [148] S. Canavan, Y. Sun, X. Zhang, and L. Yin, "A Dynamic Curvature Based Approach for Facial Activity Analysis in 3D Space," *Computer Vision and Pattern Recognition Workshop on Socially Intelligent Surveillance and Monitoring*, 2012.
- [149] P. Perakis, T. Theoharis, G. Passalis, and I. Kakadiaris, "3D Facial Landmark Detection & Face Registration: A 3D Facial Landmark Model & 3D Local Shape Descriptors Approach," *Technical Report*, Computer Graphics Laboratory, University of Athens, 2011.
- [150] M. Reale, P. Liu, L. Yin, and S. Canavan, "Art Critic: Multisignal Vision and Speech Interaction System in a Gaming Context," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Special Issue on Modern Control for Computer Games*, July 2013.
- [151] M. Reale, X. Zhang, and L. Yin, "Nebula feature: a space-time feature for posed and spontaneous 4D facial behavior analysis," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.
- [152] X. Tang, and X. Wange, "Face Sketch Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 2004.
- [153] Q. Liu, X. Tang, H. Jin, H. Lu and S. Ma, "A nonlinear approach for face sketch synthesis and recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [154] X. Tang and X. Wang, "Face sketch synthesis and recognition," *IEEE Conference on Computer Vision*, 2005.
- [155] Y. Li, M. Savvides, and V. Bhagavatula, "Illumination tolerant face recognition using a novel face from sketch synthesis approach and advanced correlation filters," *IEEE Conference on Acoustics, Speech and Signal Processing*, 2006.
- [156] Y. Zhang, C. McCullough, J. R. Sullins, and C. R. Ross, "Hand-Drawn Face Sketch Recognition by Humans and a PCA-Based Algorithm for Forensic Applications," *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 40(3), 2010.