# Reconstruction of Complete Head Models with Consistent Parameterization

**Niloofar Aghayan**

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the degree

**Master of Computer Science**

Ottawa-Carleton Institute for Computer Science

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ontario, Canada

January 2014

# Abstract

This thesis introduces an efficient and robust approach for 3D reconstruction of complete head models with consistent parameterization and personalized shapes from several possible inputs. The system input consists of Cyberware laser-scanned data where we perform scanning task as well as publically available face data where (i) facial expression may or may not exist and (ii) only partial information of head may exist, for instance only front face part without back part of the head.

Our method starts with a surface reconstruction approach to either transfer point clouds to a mesh structure or to fill missing points on a triangular mesh. Then, it is followed by a registration process which unifies the representation of all meshes. Afterward, a photo-cloning method is used to extract an adequate set of features in a semi-automatic way on snapshots taken from front and left views of provided range data. We modify Radial Basis Functions (RBFs) deformation so that it would be based on not only distance, but also regional information. Using feature point sets and modified RBFs deformation, a generic mesh can be manipulated in a way that closed eyes and mouth movements like separating upper lip and lower lip can be properly handled. In other word, such mesh modification method makes construction of various facial expressions possible. Moreover, new functions are added where a generic model can be manipulated based on feature point sets to consequently recover missing parts such as ears, back of the head and neck in the input face. After feature-based deformation using modified radial basis functions, a fine mesh modification method based on model points follows to extract the fine details from the available range data. Then, some post refinement procedures employing RBFs deformation and

averaging neighboring points are carried out to make the surface of reconstructed 3D head smoother and uniform. Due to existence of flaws and defects on the mesh surface such as flipped triangles, self-intersections or degenerate faces, an automatic repairing approach is leveraged to clean up the entire surface of the mesh. The experiments which are performed on various models show that our method is robust and efficient in terms of accurate full head reconstruction from input data and execution time, respectively. In our method, it is also aimed to use minimum user interaction as much as possible.

# Acknowledgments

First and foremost, I would like to express my deepest appreciations to my supervisor, Prof. WonSook Lee for her patience and constant support during my thesis research. Her critical comments and insightful feedbacks were essential for any steps toward completion of this research.

My thanks and appreciations also go to my colleague at CG++ Research Laboratory, Nima and Adriana, who have helped me much to complete my master studies through their valuable collaboration and guidance.

Last by not least, I thank almighty, my parents, sisters and my friend, Mahbubeh, for their constant encouragement without which this research would not be possible.

# Dedication

To my

Mother and Father

for their endless love, support and encouragement

# Table of Contents

Abstract ..... ii

Acknowledgments ..... iv

Table of Contents ..... vi

List of Figures ..... x

List of Tables ..... xvi

Abbreviations ..... xvii

Chapter 1. Introduction ..... 1

1.1 Overview of the System ..... 3

1.2 Overview of the Thesis ..... 7

Chapter 2. Literature Review ..... 8

2.1 Sculpture and Plaster Models ..... 8

2.2 Digital Sculpting using Arbitrary Photograph as Reference ..... 9

2.3 Morphing Techniques ..... 11

2.3.1 Morphable Model ..... 11

2.3.2 Interactive Morphing System ..... 13

2.4 Range Data ..... 13

2.4.1 Laser Scanner ..... 14

2.4.2 Stripe Generator ..... 17

2.4.3 Kinect ..... 17

2.5 Feature Point-based Adaptation with Generic Model ..... 20

2.5.1 Facial Feature Points ..... 20

2.5.2 Model Adaptation using Feature Points ..... 22

2.6 Control Point-based Deformation in 3D ..... 25

2.6.1 Free Form Deformation ..... 25

2.6.2 Extended Free Form Deformation ..... 26

2.6.3 Dirichlet Free Form Deformation ..... 26

2.6.4 Radial Basis Functions ..... 27

2.7 Non-Feature based Shape Manipulation ..... 28

2.7.1 Deformation using Eigen-based Approach ..... 28

2.8 Mesh Parameterization ..... 29

2.9 Multi-resolution Modeling ..... 31

2.10 Face Animation ..... 33

2.10.1 Performance Capturing for Modeling and Animation ..... 33

2.10.2 Physically-based Animation ..... 39

2.11 Mesh Repair ..... 40

2.11.1 Global Repairing Methods ..... 40

2.11.2 Local Repairing Methods ..... 41

Chapter 3. Methodology ..... 44

3.1 Generic Model ..... 45

3.2 Input Model ..... 46

3.2.1 Laser-scanned Model ..... 46

3.2.2 Publically Available 3D Face Model ..... 47

3.3 Surface Construction from Points and Filling Holes ..... 49

3.4 Registration of Input Face in Generic Face Space ..... 50

3.5 Feature Points Extraction ..... 51

3.6 Feature-based Generic Head Adaptation to Input Model ..... 55

3.6.1 Lips Contour Detection ..... 57

3.6.2 Regional Mask Creation ..... 60

3.6.3 Point Partition based on Region ..... 61

3.6.4 RBF-based Mesh Modification ..... 62

3.7 Mesh Refinement ..... 65

3.7.1 Resolution Increase using Subdivision ..... 65

3.7.2 Surface Point-based Adaptation using Cylindrical Projection ..... 66

3.8 Post Mesh Refinement for Smoothing ..... 68

3.8.1 Smoothing using RBFs ..... 69

3.8.2 Smoothing using Average Points Calculation ..... 70

3.9 Mesh Defect Repairing ..... 71

Chapter 4. Result and Validation ..... 75

4.1 Experimental Results ..... 75

4.1.1 Person A with Expression I ..... 76

4.1.2 Person A with Expression II ..... 80

4.1.3 Person A with Expression III ..... 83

4.1.4 Person B with Expression I and Closed Eyes ..... 86

4.2 Analysis of Meshes, System Performance and Validation ..... 89

4.3 Discussion ..... 95

Chapter 5. Conclusion ..... 100

5.1 Summary ..... 100

5.2 Contribution ..... 101

5.3 Future Research ..... 102

References ..... 103

# List of Figures

Figure 1.1: Proposed system pipeline. ..... 6

Figure 2.1: 3D surface construction from a reference photograph. a) Reference photograph used by sculptor [3] and TextureFit tools [4], b) texture map Composed of the photograph, c) 3D mesh result from different views. ..... 11

Figure 2.2: Cyberware laser scanner. ..... 15

Figure 2.3 : Microsoft's Xbox Kinect. ..... 18

Figure 2.4: Defined feature points distribution by MPEG-4 standard. ..... 21

Figure 2.5: The flow diagram of generic model adaptation based on feature points. ..... 22

Figure 3.1: Differences between meshes due to a wide variety in a) expression, b) resolution, c) mesh defect, and d) mesh structure. ..... 44

Figure 3.2: Generic model (12600 triangles). ..... 45

Figure 3.3: Scanned face of an individual (723614 triangles). ..... 47

Figure 3.4: Various facial expressions are performed by a female subject. ..... 48

Figure 3.5: Surface reconstruction using Poisson reconstruction method. a-b) Different views of original point cloud, c-d) different views of reconstructed surface. ..... 49

Figure 3.6: Facial feature points on a) the generic mesh, b) the original input face. ..... 50

Figure 3.7: Registration process using Procrustes Analysis. a) The generic mesh (shown in white) and the original input face (shown in pink) before registration, b) the input face and the generic mesh after registration. ..... 51

Figure 3.8: 2D distributions of 128 feature points on a) generic model and b) input model. ..... 52

Figure 3.9: 3D feature points at the back of the generic model. ..... 54

Figure 3.10: A sample of using RBF functions without applying any regional information. a) Generic model before deformation plus illustrated direction of feature point displacements, b) deformed generic model representing unpleasant deformation result around the mouth. ..... 55

Figure 3.11: Sample representation of a) lower-opened vertex (shown in blue), b) upper-opened vertex (shown in green). ..... 57

Figure 3.12: Contour vertices on a) upper lip and b) lower lip. ..... 59

Figure 3.13: Representation of 2D frames of points. Green (upper frame) and blue (lower frame) frames are associated with upper lip and lower lip, respectively. ..... 61

Figure 3.14: Feature-based adaptation using RBFs deformation. a) Generic model, b) input model, c) deformed generic model. ..... 64

Figure 3.15: Subdivided model after feature-based adaptation. ..... 65

Figure 3.16 : Surface point-based adaptation of subdivided model to the input face using cylindrical projection-based method. a) Transformed point to cylindrical coordinates, b) detected intersecting triangle using barycentric coordinates, c) computed 3D point on the input face. ..... 67

Figure 3.17: Surface point-based adaptation process. a) Input face, b-c) subdivided model from two views, d-e) refined model after cylindrical projection from two views. ..... 68

Figure 3.18: Demonstration of a) the refined region (shown in pink) and the border points (shown in red) from three different views and b) the correspondences of the border points (shown in blue). ..... 69

Figure 3.19: An illustration of the border points (shown in red) and their corresponding neighboring points (shown in green). ..... 70

Figure 3.20: Top row demonstrates resultant mesh after surface point-based adaptation and second row shows natural looking and smoothed mesh after two post refinement approaches based on RBFs deformation and averaging neighboring points from three different views. ..... 71

Figure 3.21: Created defects such as self-intersections around nose alae. ..... 72

Figure 3.22: The flow diagram of mesh model repairing using the technique proposed by Attene [70]. ..... 72

Figure 3.23: Head model after a) repairing method, b) post refinement method with contour points shown in red, c) repairing method with extra points (shown in green) surrounded by contour points (shown in red), d) undesired points removal. ..... 73

Figure 3.24: Fixed flaws such as self-intersections around nose alae. ..... 74

Figure 4.1: Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views. ..... 76

Figure 4.2: Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model. ..... 77

Figure 4.3: Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views. ..... 77

Figure 4.4: Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from three views. ..... 78

Figure 4.5: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method..... 79

Figure 4.6 : Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views. ..... 80

Figure 4.7 : Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model. ..... 80

Figure 4.8 : Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views. ..... 81

Figure 4.9 : Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from three views. ..... 81

Figure 4.10: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method..... 82

Figure 4.11 : Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views. ..... 83

Figure 4.12: Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model. ..... 83

Figure 4.13: Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views. ..... 84

Figure 4.14: Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from three views. ..... 84

Figure 4.15: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method..... 85

Figure 4.16: Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views. ..... 86

Figure 4.17: Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model. ..... 86

Figure 4.18 : Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views. ..... 87

Figure 4.19: Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from two views. ..... 87

Figure 4.20: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method..... 88

Figure 4.21: a) Having the sampled surface S, the closest point to the point A is the point A' on the surface S', b) having the sampled surface S', the closest point to the point A' is the point A on the surface S. ..... 89

Figure 4.22 : Color scale of the difference from the input surface to the output surface in the range [-0.2 0.2] for person A with expression a) I, b) II, c) III, and d) person B with expression I. In this color scale, pure red represents zero distance between two surfaces and colors from red to blue represent increase in distance. ..... 94

Figure 4.23: Smoothing results on a-b) surface point-based adapted model after c-d) HC Laplacian smooth method [82], e-f) our proposed smoothing method. ..... 96

Figure 4.24: Skin representations from a) cheek area and b) back of the head area. ..... 97

Figure 4.25: Mesh representation after surface repairing methods. a-c) obtained surface after Poisson reconstruction method, d-f) reconstructed surface after employing our method. ..... 99

# List of Tables

Table 4.1: Performance analysis for person A with expression I. ..... 90

Table 4.2: Performance analysis for person A with expression II. ..... 90

Table 4.3: Performance analysis for person A with expression III. ..... 91

Table 4.4: Performance analysis for person B with expression I and closed eyes ..... 91

Table 4.5: Reconstruction error calculation between input mesh and output mesh for person A with expression I. ..... 92

Table 4.6: Reconstruction error calculation between input mesh and output mesh for person A with expression II. ..... 92

Table 4.7: Reconstruction error calculation between input mesh and output mesh for person A with expression III. ..... 92

Table 4.8: Reconstruction error calculation between input mesh and output mesh for person B with expression I. ..... 93

Table 4.9: Average mesh processing times of all processes in our system for all four models. .. 93

# Abbreviations

| | |
|---|---|
| **2D** | Two Dimensional |
| **3D** | Three Dimensional |
| **FFD** | Free Form Deformation |
| **EFFD** | Extended Free Form Deformation |
| **DFFD** | Dirichlet Free Form Deformation |
| **RBFs** | Radial Basis Functions |
| **IR** | Infrared |
| **PCA** | Principal Component Analysis |

# Chapter 1. Introduction

Over the past decades, development of capturing tools and resources has provided us the ability to acquire individuals' 3D head meshes which can be used in computerized facial animation. Captured 3D models usually vary in mesh structure, resolution and expression and might also suffer from not only missing parts such as ears and back of the head, but also defects like holes and self-intersections. Thus, a level of manual repair and modification is commonly required for these models. Due to existing preceding irregularities on the original meshes, direct processing of such models is erroneous and difficult in practical applications including morphing, detail transfer and texture mapping. Therefore, it is highly recommended to process parameterized meshes which establish correspondences between points and features of the head models since the head models have no region information without parameterization. Parameterized meshes can also be useful for facial animation eventually. Moreover, the definition of consistent parameterizations is a set of parameterizations which rely on only one generic mesh with a desired structure (or one abstract simplical complex) and also the correspondences between the features of the models which may have different resolutions.

Besides, creation of animated human head toward individualization is a tedious and challenging task even for skilled 3D artists due to complexity of human facial anatomy. The harmony of details and shape which makes realistic and unique characteristics of the face is directly associated with elements such as gender, age and ethnicity of the individuals and should remain intact through entire reconstruction process. Even a small anomaly in the reconstructed 3D model is easy to be recognized by audience because of the well-known structure and form of the human head. Moreover, producing plausible results from a variety of input types is still non-

trivial problem. We should keep in mind that vital circumstances such as execution time, storage requirements and model suitability are always taken into consideration in order to enhance the process efficiency which consequently affect model realism. The preceding issues and concerns have made reconstruction of human being an extensively studied subject in computer vision and computer graphics.

Some of the related works described in the literature propose solutions relying on establishing consistent parameterization on only neutral expression of individuals and then animating it artificially either using design tool or motion capture data which is a lot easier to be accomplished. Even though they are capable of reproduction of various expressions (not necessarily extreme ones), the expressions after the shape reconstruction are usually limited to a specified range or most of time different from natural expressions performed by individual. Moreover, reconstruction of individual's front face is the main focus of a majority of existing works and information for the rest of the head like ear, head back and neck is eventually excluded. Therefore, full head reconstruction based on the existing front part of the face is still a challenging issue.

This work is motivated by the drawbacks and bottlenecks encountered in the related researches in the literature. Our ultimate goal is to reconstruct highly accurate and consistently parameterized head of expressed model from different inputs with a high level of autonomy. Hence, we propose a robust and practical system which can have not only our own scanned faces under control, but also included publically available face data under control while faces might be in form of either point clouds or triangular mesh and might cover various face shapes, level of details, expressions, missing parts or triangles and surface defects including self-intersections. In other words, the system converts all input models into one mesh structure using a given generic

mesh and establishing a consistent parameterization among them while preserving the individual shapes and expressions. Based on available frontal face of the input model, the system robustly and efficiently reconstructs a smooth surface of whole head from input model and recovers missing parts without requiring a great deal of user interaction from various input models which extend the generality of the system.

## 1.1 Overview of the System

We introduce a framework for creation of a personalized head model while establishing consistent parameterization. Our framework begins with the creation of input model pool by capturing complete 360 degree models of the heads using a Cyberware laser scanner where we perform scanning task and collecting 3D faces from publically available face database. It is noted that a variety of facial expressions may exist in such input model pool. After collecting data, a surface reconstruction process is accomplished for the input face either in the form of point cloud, which would be transferred to a mesh structure or triangular mesh which its missing points would be properly filled.

Before proceeding with mesh manipulation processes, a 3D geometrical fitting stage should be employed to bring input faces into the generic model space and to do so, nine 2D markers are manually selected on model surface and their corresponding 3D values are computed by leveraging a back-projecting approach. These values are used by Procrustes analysis method in order to solve registration problem. Then, dense surface correspondences are achieved by a two-step mesh adaptation: feature point-based adaptation (rough matching) and surface point-based adaptation (fine matching) and subdivision method. Feature point-based adaptation starts with taking snapshots of both generic model and input face from front and side views. Afterward,

predefined number of 2D feature points is extracted in a semi-automatic way on generic model and input face. This adequate set of feature points guarantees to capture appropriate shape of input face from a very different generic head. 3D positions of feature points are calculated using back-projecting approach. To properly reproduce facial expressions which require separating upper lip and lower lip, a mouth movement simulation is performed by integrating regional information into radial basis functions so that they no longer depend on only distance information. According to two sets of 3D feature points extracted from input and generic meshes, a generic model is adapted to input face using modified RBF-based deformation. As a result, our method suitably recreates not only individual's neutral expression, but also a variety of facial expressions. Due to the absence of head back in input model, the system provides users an interactive interface on the snapshot from side view of model to change proportion of the head back to the facial region. By default, the mesh inherits head back, neck and ears from generic model after feature-based adaptation step.

To increase resolution of the mesh, Loop subdivision method is used which makes the reconstruction of a multi-resolution model possible in the system by choosing different number of loop iterations. Next, second adaptation method relying on surface points is carried out to deform the generic model to fit all of its vertices to the input face. The preceding method is constituted of cylindrical projection and barycentric coordinates to provide a fine match between the input face and the subdivided mesh.

Furthermore, a post refinement procedure is leveraged based on RBFs deformation and averaging neighboring points to create a smooth and natural looking surface through entire model. Finally, an automatic repairing method is employed to remove mesh defects like self-intersections which are mostly introduced during surface point-based adaptation that lowers the

quality of the mesh. Since such approach results in manifold and watertight mesh, some additional steps are added to repairing method to recover the desired mesh shape around mouth if required. As a result, the system output is obtained as a complete, cleaned and smooth head and adopted to individual's face. Figure 1.1 depicts an overall flow diagram of the aforementioned methods leveraged for head reconstruction from two types of input.

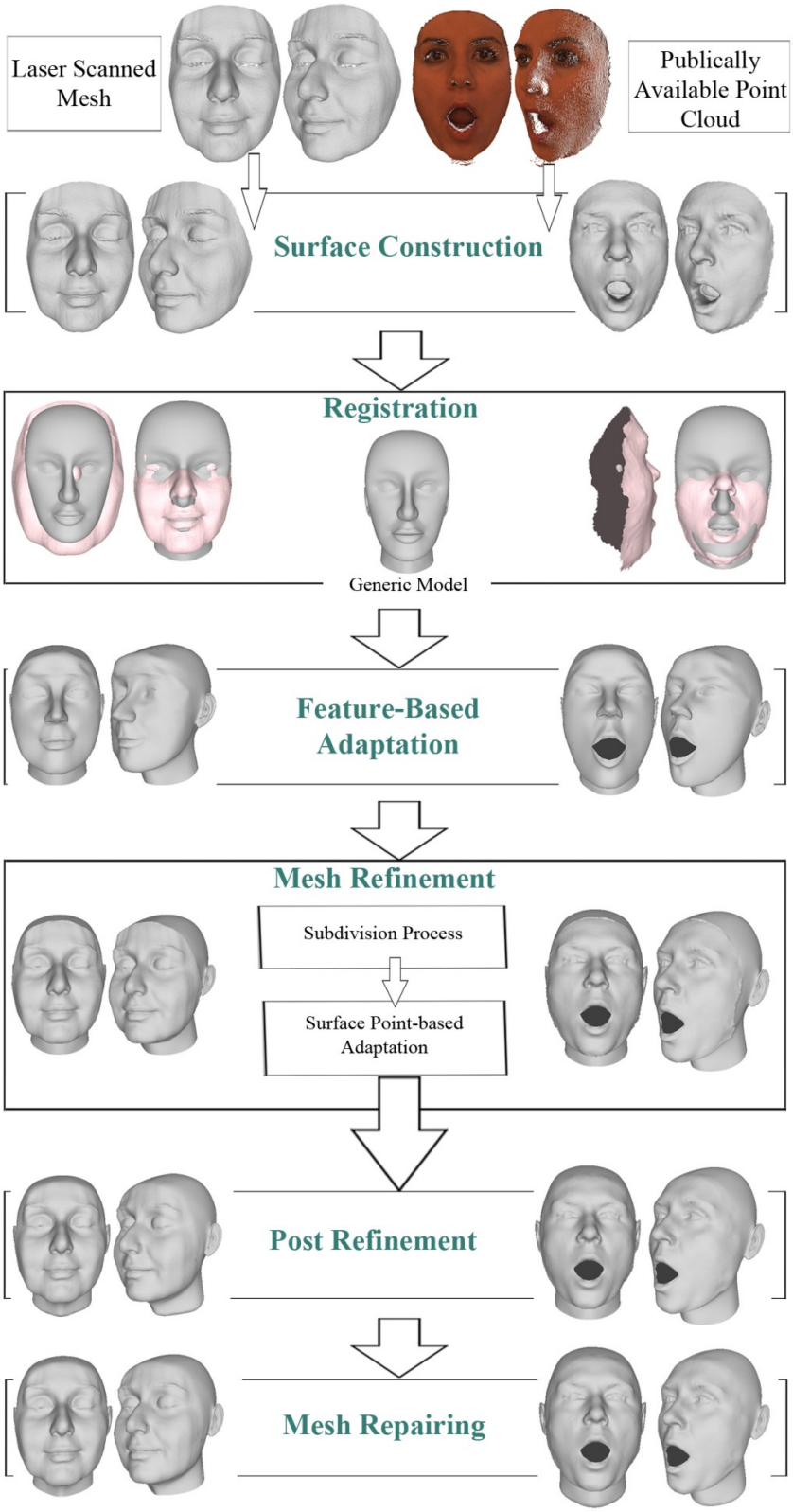


Figure 1.1: Proposed system pipeline.

## 1.2 Overview of the Thesis

The rest of this thesis is organized in the following manner:

- Chapter 2 elaborates a background study of related works for facial modeling and animation based on different capturing devices and input types. Since we are dealing with mesh defects such as self-intersecting faces, some existing solutions for remeshing and cleaning up the models are also covered in this chapter.
- Chapter 3 outlines the framework for this research and details the methodology used for this thesis study to create a personalized animation head model. It covers methods used for data gathering, reconstruction, registration, manipulation (including a novel mouth movement simulation), refinement (including a novel surface smoothing method) and repairing of surfaces.
- Chapter 4 presents experimental results carried out to demonstrate the efficiency and robustness of the proposed scheme. Evaluation of the proposed scheme including its benefits, general issues and limitations is also covered in this chapter.
- Chapter 5 concludes the thesis and summarizes the methods used for this research. At the end, it outlines the potential and future research direction of this work.

# Chapter 2. Literature Review

In the last two decades, a huge number of methods have been proposed for reconstruction of a realistic and plausible human face in a virtual environment. The following subsections delineate some of these reconstruction methods which can mainly be classified into categories including plaster model-based methods, morphing methods, interactive deformations and performance-based methods. In such methods, some sorts of user intervention are required most of the time even as minimal as it can possibly be. These manual interventions might be setting up the equipment, making fine adjustments to parameter or processing data. The user needs to carry out a trial-and-error procedure in some cases such as for setting equipment parameters or for putting markers in interactive systems which dynamically capture data. In our method, it is aimed to use minimum user interaction as much as possible and all the processes except registration and feature points selection are done automatically.

This chapter ends with brief descriptions of some related research in the context of polygonal model repairing.

## 2.1 Sculpture and Plaster Models

One of the early techniques in face modeling is based on using plaster and sculpture models which are built in real world. However, creating sculptures of the characters' head is a skillful task and without proficiency the result would not be satisfactory enough to be used as the reference for generating 3D head models in a semi-automatic way.

In 1987, Magnenat-Thalmann et al. [1] built 3D face models of Marilyn Monroe and Humphrey Bogart for the short film called "Rendez-vous à Montréal" in which they utilize the digitized photographs taken from different angles of chosen facet and vertex markings on plaster models. The extracted 2D data from those photographs is used to obtain 3D coordinates of the model data. Also, higher resolution data can be obtained in any area of interest (such as lips and eyelid) in order to animate the facial models more accurately. However, one of the tasks involved in such technique is drawing a triangulated mesh on each face and doing so is neither fast nor easy.

Since then, plaster models have been successively used in movie and game industries. One example is the creation of character Geri [2] in a short animated film named "Geri's Game" produced by Pixar company in 1997. Geri's construction process starts with making the head and hands as sculpture and then digitizing them into a 3D mesh model.

## 2.2 Digital Sculpting using Arbitrary Photograph as Reference

Creation of 3D head models of the famous celebrities who are not available at the moment for filming and photographing and specially those who are not alive has become so popular in the last decades. The aforementioned work by Magnenat-Thalmann et al. [1] is an outstanding example of 3D construction of the memorable actress Marilyn Monroe. This approach is based on the subject's pictures which are one of the most available resources of all time. Since there might exist only one or some pictures of a specific subject in various locations, positions and time intervals, the construction task becomes tough and tricky. There are two feasible solutions in order to handle such problem; the first solution is that the individual's pictures should be matched all together by applying some user interactions. The other solution is to perform

comparisons between pixels in the pictures which definitely needs manual set ups of photographing environment. One of the early digital sculpting software called Sculptor [3] is introduced to construct 3D objects from several photographs based on local and global geometric deformations. Different actions involved in the sculpting process consist of adding, deleting, modifying and assembling triangle meshes. User interactions are provided into such software first by holding an interactive input device called the Spaceball in one hand in a way that the 3D object can be moved and examined from several angles. Then, a mouse held in the other hand of the user is used to perform picking and deformation tasks in 3D space from different views provided by Spaceball. Simultaneous usage of the mouse and Spaceball through the process leads to a close resemblance between digital and traditional sculpting results. The quality of resultant 3D models can be enhanced further by employing techniques such as texture mapping which applies real images onto the model surface in 3D. A texture mapping tool called TextureFit [4] is utilized to improve final realism of the objects using interactive fitting of the texture by the user. As initialization of the fitting process, designers select main feature points (see section 2.5.1) on 3D model surface and the corresponding values of such feature points in 2D image is automatically computed. It also allows users to adjust 2D point locations and the rest of 3D object vertices are projected into 2D image according to the location of main feature points.

The preceding tools (Sculptor and TextureFit) were used by MIRALab research group at the University of Geneva to build 3D objects. The construction process is based on a reference photograph of a soldier from the Terracotta Army as shown in Figure 2.1. The texture map obtained from one photograph is created using Photoshop software. Rest of the images illustrates the 3D soldier model of Terracotta army in different views.

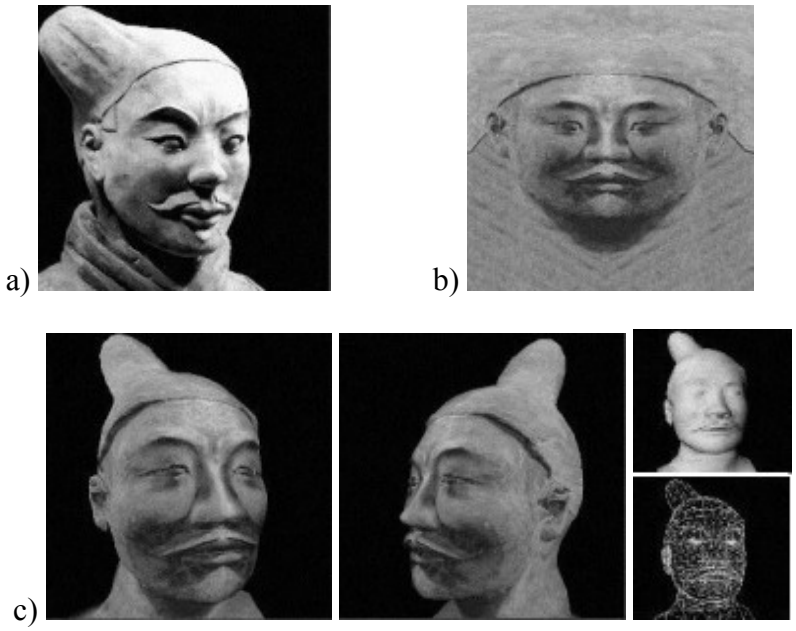


Figure 2.1: 3D surface construction from a reference photograph. a) Reference photograph used by sculptor [3] and TextureFit tools [4], b) texture map Composed of the photograph, c) 3D mesh result from different views.

## 2.3 Morphing Techniques

Morphing techniques either in 2D or 3D are extensively used by applications in computer graphics. In the following subsections, some of the outstanding works based on morphing techniques are described.

### 2.3.1 Morphable Model

In 1999, Blanz and Vetter [5] introduced a morphable model for synthesis of 3D faces while preserving naturalness of the constructed face. Their process begins with building a large database of 200 subjects' face (100 males and 100 females) aged from 18 to 45 years and acquired by a 3D Cyberware$^{TM}$ laser scanner while removing shoulders and behind of the ears

regions by an automatic pre-processing method. Every face in the database consists of approximately 70,000 vertices and is based on vector space representation including shapes and texture vectors ($S_i$(X, Y, Z) and $T_i$(R, G, B) respectively). Morphing between faces in the database requires dense 3D point-to-point correspondences which are achieved in a semi-automatic way using an optical flow algorithm [6]. After providing full correspondences between all faces, new realistic 3D faces can be expressed as linear combinations of models available in the database using the following formulas.

$$S_n = \sum_{i=1}^{m}(a_i S_i), \quad T_n = \sum_{i=1}^{m}(b_i T_i), \quad \sum_{i=1}^{m}(a_i) = \sum_{i=1}^{m}(b_i) = 1 \qquad (2.1)$$

Where a= ($a_1$, $a_2$,…,$a_m$) and b= ($b_1$, $b_2$,…,$b_m$) are parameterization coefficients and a new face can be generated by changing the values of such parameters. Moreover, the probability distribution for $a_i$ and $b_i$ coefficients are calculated with the purpose of controlling the coefficient likelihood and consequently regulating the appearance likelihood of the constructed faces.

In their work, a new 3D face can be derived from two input types; first, one or more 2D images can be used as the system input. However, it is noted that reconstruction of 3D shape and texture of a face from a single image suffers from ill-posed issue. A novel new face can also be derived from manipulating the vector spaces (shape and texture) of 3D scans in a "natural way".

In the case of matching morphable model to a 2D image, it is crucial to adjust 3D model coefficients according to many rendering parameters which can be a time consuming task; otherwise the constructed image wouldn't be as similar as possible to the input image. Rendering parameters which should be provided by users consist of camera position, intensity of ambient and directed light and surface shininess. Also, estimation of some other parameters such as object scale, image plane rotation and translation is necessary.

On the other hand, computation of derivatives for each iteration step is a time consuming task and since all the procedure is performed in 105 iterations on an SGI R10000 processor, total computation time is noticeably slow (about 50 minutes). Moreover, it is noted that 3D scanned faces in the dataset contain a limited set of facial expressions and variations attained during speech.

In overall, while leveraging morphable models as prior models decreases the ambiguity of facial deformation using images, the reconstructed geometry and motion don't have the quality of individual-specific captured data.

### 2.3.2 Interactive Morphing System

The other examples of using morphing techniques either in 2D or 3D are the works proposed by Lee and Magnenat-Thalmann[7] [8]. In their method, human faces (which are consistently parameterized) are initially reconstructed from either two orthogonal pictures or range data using the method introduced by Lee and Magnenat-Thalmann [9] [10]. Then shape and texture parameters are separately treated to achieve intuitive face modeling using a user-friendly interface. The main techniques utilized are linear-interpolation among the given face models for the shape and image metamorphosis based on triangulation for the texture. While this method is a good candidate for face reconstruction from pictures in terms of speed and requirement cost, the results still suffer from lack of details and smoothness (not realistic enough) required for high-level applications including computer-generated films. In the case of the range data (from laser scanner) as input [10], smooth results can be obtained but the resultant faces are animated with given expression parameters in this method. Therefore, the reconstructed expressions may

be different from the expressions performed by the person. Morphing techniques can be also used for animation purposes which are covered in Section 2.10.

## 2.4 Range Data

Acquisition of 3D range data has received considerable attention in the last decades since the resultant range data can be used for many higher level vision tasks such as 3D object modeling and recognition in game and movie industries and also health-science projects. Based on that, hardware and software systems are developed in order to digitize the shape and texture/color of the object and typically produce output as a point set or a triangulated mesh. The factors such as acquisition speed and constructed object accuracy while reducing noise play an important role in the complexity and cost of the acquisition systems. Laser scanner, stripe generator and Kinect are three scanning technologies used extensively for obtaining 3D range data especially from human subject.

### 2.4.1 Laser Scanner

3D scanners (specifically 3D non-contact scanners) are repeatedly employed surface acquisition tools by the most popular commercially available systems. These scanners can be classified into active and passive categories in term of radiation or light emission.

Laser scanners are non-contact active scanners which radiate light to the object and the light reflection of the object surface can be detected and used by the scanners to identify the surface. An example of 3D active laser scanner is triangulation scanner. It is so called because the laser emitter, the camera and the laser dot resulted by emitting a laser on the subject create a triangle. This scanner uses the known distance between the laser emitter and the camera, the angles of the

laser emitter and camera corners to determine the laser dot corner on the object and eventually detects most of the surface points.

Another example of 3D active laser scanner is time-of-flight which determines the object distance by calculating the time between light radiation from a laser and its reflection captured by a detector. Such scanner is capable of handling very long distances.

Moreover, a number of commercial scanners are offered by Cyberware production for capturing external structure and appearance (color information) of the objects (Figure 2.2 (a)). Based on the physical structures and usage applications, those scanners capture range data with different speeds which consequently affect the quality of the outputs. As the aforementioned work by Blanz and Vetter [5] uses Cyberware scanner to generate a large face database for creating a 3D morphable face database. Cyberware scanners and even their complicated and enhanced versions have been extensively used for collecting input data [11] [12] [13] [14].

Figure 2.2: Cyberware laser scanner.

While such scanners make capturing the range data with very high-quality facial details such as pores, wrinkles and folds possible, their acquisition speed is fairly slow compared with motion capture systems (section 2.10).

#### 2.4.1.1 Data Post-Processing

As mentioned earlier, 3D acquisition systems such as laser scanners capture raw measurements of face data (shape, color or texture) as a point set or a triangulated mesh. However, these outputs are usually unstructured and might have errors and holes because of sensor noise and occlusion, respectively. Therefore, data can't be directly used and actions such as fitting of scanned data, filling of surface holes and remeshing of current models are needed to be done. Hence, a surface reconstruction technique called Poisson reconstruction method [15] can be used to recreate the surface from point samples due to its benefits over other techniques such as Power Crust [16] and Robust Cocone [17]. This method merges the advantages from both global and local fitting schemes which lead to higher flexibility and tolerance with the existence of imperfect data and it eventually reconstructs the surface with high quality.

This method states surface reconstruction from oriented points as the solution to a Poisson equation. Given the point set, the main steps involved in this method are adapting an Octree which approximates the raw data, computing vector field that is zero almost everywhere except the points near the surface, splatting the samples, computing indicator function and divergence, solving the Poisson equation and finally extracting isosurface.

### 2.4.2 Stripe Generator

An active non-contact scanner based on structured light is called stripe generator which can construct a 3D shape by two main actions. First, single light strips (usually line) are projected onto the object surface. Then, they will be scanned across the surface of the object by the camera. The advantage of using stripe generator over the laser scanners is relatively lower equipment costs. However, many images and lots of time are required to generate high-resolution 3D shapes. In the case of using grid as stripe pattern, complicated algorithms [18] are required to properly identify the object. An interesting work by Proesmans et al. [19] describes how to construct a good dynamic 3D shape using a slide projector in a frame-by-frame manner from a video. A novel 3D shape measurement method proposed by Jia [20] is called triangular-pattern phase-shifting profilometry which is based on structured light. Such technique speeds up the acquisition speed of 3D object surface since it utilizes a simple intensity-ratio computation and smaller number of images or measurement steps which are necessary to reconstruct the 3D object. However, using this technique the accuracy measurement can be affected by the reflectance of the measured object surface.

### 2.4.3 Kinect

Despite the fact that 3D scanning tools like Cyberware laser scanners acquire range data with very high-quality details, the majority of them require special hardware which is usually very expensive. In 2010, Microsoft Corporation released a motion capture device called Kinect (Figure 2.3) which was initially used as a controller for Microsoft Xbox 360 video game console. Since then, various open-source drivers have been developed for Kinect and eventually make it usable for a wide range of applications on personal computers. Remarkable lower price of Kinect

compared with other scanning technologies makes it an ideal choice for capturing individualized 3D avatars which are widely used in gaming, 3D modeling and augmented reality applications. The main structure of Kinect contains a RGB camera, a 3D microphone and the depth sensors consisting of an infrared (IR) projector and an IR camera. The capturing process of Kinect starts with casting thousands of IR rays by IR projector through the entire scene. The reflectance from the surface of objects are captured and recorded by IR camera sensor. Afterwards, the depth values are measured for all received rays. Then, all data is retrieved on chip system and creates a depth grid with the resolution of 640x480 pixels at 30 frames per second. The practical ranging limit of Kinect is 1.2-3.5m. Later, the image captured by the RBG camera is aligned to depth map and eventually a texture map is constructed by combining the image and depth map.

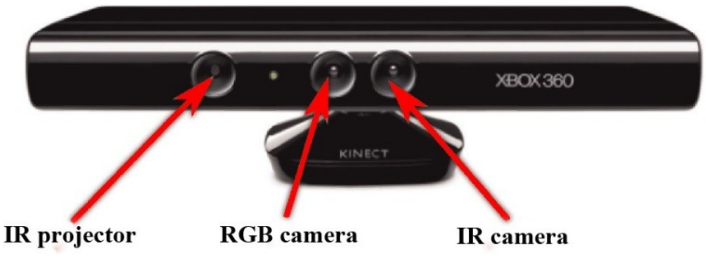


Figure 2.3 : Microsoft's Xbox Kinect.

The main problem using Kinect as data capturing tool is that the accuracy of recorded data can be significantly affected by external factors such as lighting condition and distance to Kinect sensor. Hence, many techniques have been proposed to reduce the embedded noise in the raw Kinect data and improve geometric quality of reconstructed surface with finer details. While reconstruction methods based on Kinect speed up 3D surface creation, the resultant models still hold lower quality and resolution compared with the models acquired by 3D scanners like

Cyberware laser scanner. Fulfillment of the accuracy and quality requirements by reconstruction methods is necessary for obtaining realist head models.

#### 2.4.3.1 Enhancement of Raw Kinect Data

Leveraging Kinect sensor, Microsoft introduced an interactive reconstruction system called KinectFusion [21]. It utilizes a standard Kinect camera which can be moved by the user through any indoor space to quickly capture live depth data. Such method makes real-time 3D reconstruction of a physical object possible. The method is almost executed entirely on the graphics card so that real-time speed can be obtained. Initially, the method collects and processes the data from the Kinect frame by frame. Then, it calculates the position of camera in space and fills a predefined 3D volume with surface data. Lastly a ray casting method is utilized to produce isosurfaces from the volume. The main drawback of KinectFusion is that it requires high-computational power and a powerful GPU to work. While it considerably enhances the quality of captured data by Kinect, the result still lacks the details of object surface required by high-level applications.

Another example of utilizing Kinect is an automatic method proposed by Zollhöfer et al. [22] which computes individualized avatars with only one RGB image and the corresponding depth map by fitting a morphable face model [5] to the captured data. The reconstruction process initiates by capturing an RGB image and the corresponding depth map by a Microsoft's Kinect sensor. The calibration between the IR and the RGB camera is substantially made and depth data are approximated as a metric point cloud where each point is associated with a corresponding RGB value. The raw data acquired by Kinect at each frame is noisy and consists of holes. To remove or lighten such flaws, some pre-processing approaches are utilized which significantly

enhance the quality and smoothness of input data. Afterwards, the main regions containing the face, nose, and eyes are determined on RGB image and mapped to 3D point cloud which has same topology as image. Then the locations of few characteristic feature points are automatically estimated based on their corresponding 3D regions. The system provides user the opportunity to optimize his pose with respect to camera which eventually corrects location of detected facial features in real time. A coarse alignment between the scanned data and average face of the morphable model is obtained by a generalized Procrustes analysis. Then, it is followed by a non-rigid registration of rough aligned result to the average face which minimizes distance between models. In order to animate or analyze the output model more semantic information is required than few detected control points. So, a morphable model is used to build the face which best estimates the deformed average face. Face realism is improved further by creating texture map using captured RGB images. The drawbacks of this method are handling only frontal face of the models instead of the entire heads (including back part, ears and neck) and also providing limited expressions of individuals due to characteristics of the morphable model. Moreover, such method produces the results which suffer from lack of details required for high-level applications including computer-generated films.

## 2.5 Feature Point-based Adaptation with Generic Model

### 2.5.1 Facial Feature Points

Feature points are the prominent points on the face that represent the characteristic face features. Such points are utilized to supply spatial references to main positions on an individual's face (e.g. as inputs to deformation functions). In other words, they provide us meaningful

correspondences to important characteristics such as where the mouth, eyes and lips are. This semantic information would also be essential for animation or other purpose works on the head. Hence, researchers have proposed a variety of models on the number and distribution of feature points according to the field of usage including facial tracking, modeling, animating and retargeting. The number of feature points typically ranges between 50 and 300 in order to achieve a closer approximation of the model surface.

One of the most common models is the one established by MPEG-4 standard. Not only this standard determines the location of feature points, but also it determines the magnitude of the feature point displacement used for animating the face. MPEG-4 defines 84 points to properly represent the facial anatomy and movement mechanics of an individual's face as illustrated in Figure 2.4. Filled dots represent the feature points that are affected by facial animation parameters with the purpose of animating the facial region.

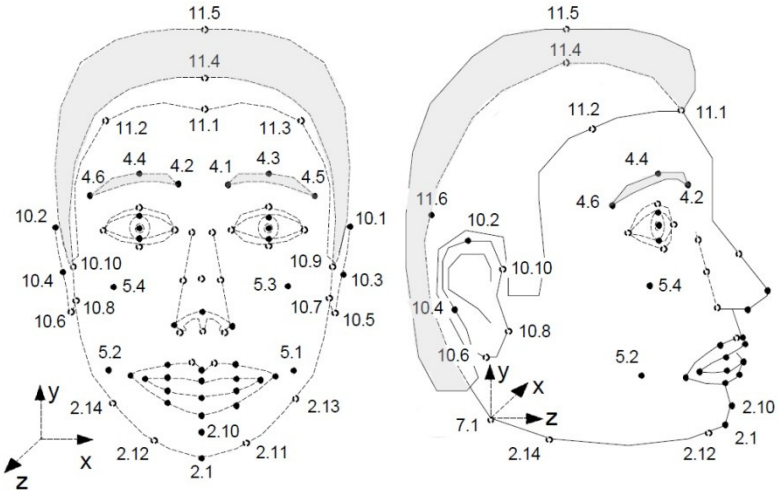


Figure 2.4: Defined feature points distribution by MPEG-4 standard.

Recently, the commercial software called Live Driver [23] was released which makes a real time face tracking, analyzing and animating possible. Based on the captured data by a regular

webcam camera, it automatically detects a sparse set of 2D facial features (64 points) and 100 expressions per frame. One of the advantageous features of such tracker is that it can still efficiently operate under different lighting and environmental conditions. Hence, there is a growing trend towards using Live Driver in different applications. While Live Driver provides a real time and robust tracking of facial features, it fails to capture small-scale and high-quality details (wrinkles and folds) necessary for realist facial construction and animation. Also, tracking accuracy is degraded if the user is not facing the camera.

### 2.5.2 Model Adaptation using Feature Points

To extract patterns of the feature points either from 2D images or 3D models, many works has been introduced varying in their level of autonomy. The Figure 2.5 illustrates the high-level view of the works for model adaptation using feature points and a generic mesh.

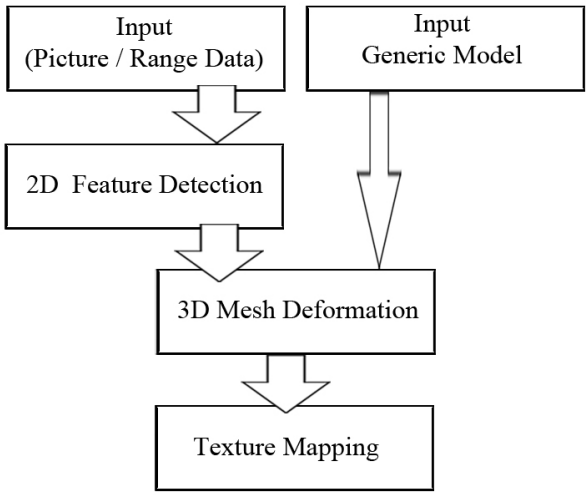


Figure 2.5: The flow diagram of generic model adaptation based on feature points.

In 1991, Kurihara et al. [24] proposed a method to create and animate a face model first by interactively specifying a few control points on pictures of an individual's face. Then, the

displacement of the feature points is used by the transformation to displace the rest of the points by linear interpolation in a 2D parameter space. As a result, a generic model is fitted to the input picture. However, this technique fails to properly construct the shape from a very different generic model due to using few points and interpolation which eventually causes inaccurate texture fitting.

An automatic feature extraction technique is described by Ip et al. [25] to locate feature points in the images by using a local maximum curvature tracking algorithm. Extracted facial features are used to construct 3D head models of specific persons. Also, a texture map is constructed by blending two facial images (front and side views) and mapped onto the model surface. For such purpose, a linear interpolation function is used. In addition to that, a technique outlined by Lee et al. [10] is also based on feature point extraction from either pictures or range data so that a generic mesh model can be fitted to a person-specific head. The first step to approximate an individualized head is to detect feature points which capture rough shape of target face on two orthogonal images or range data and then obtain 3D positions of the feature points to adapt a generic model using a geometrical deformation. We will explain various 3D geometrical deformation techniques in the next section 2.6 more in detail. The feature detection and localization is accomplished in a semi-automatic manner using the structured snake method which makes sure the points on frontal and side views correspond to each other. User interaction is also provided in such system in the case of not proper localization of features. The texture map is made using multi-resolution technique for image mosaic. However, the results achieved by this method still suffer from lack of details and smoothness (due to the deformation technique) required for high-level applications such as computer-generated films.

Moreover, some works have represented the principle of fitting a generic model to 3D input data using feature points. Highly detailed range data can be captured by scanning systems such as Cyberware color scanner which are discussed in details in the previous sections. Lee et al. [11] [12] introduce an adaptation approach which fits a generic model to the scanned data. The constructed individual's face owns a structure consisting of tissue, muscle and bone which makes the creation of the realistic facial expressions feasible due to the existence of physics-based structure.

An adaptation-based method is defined by Zhang et al. [26] to create animated models from 3D scanned faces of specific persons. Based on 6 3D landmarks placed by a user on the corners of eyes and mouth, generic mesh model is globally adapted to the surface of the scanned model. Then, it is followed by a local adaptation which deforms the geometry of the generic model to fit all of its surface points to the points of scan surface. The employed generic model consists of structured layers for physically-based animation which are fully described in section 2.10.2.

Facial features can be localized on 2D pictures or 3D surfaces manually which is a tedious, time consuming and even erroneous task especially when a large number of feature points exists to deal with them. On the other hand, the majority of available approaches for automatically detecting facial features are not robust and manual adjustments are still needed due to the large differences the faces can have in a scene due to factors such as position, expression, illumination, skin color, wrinkles, spots, etc. Therefore, semi-automatic method provides more reliable result compared with automatic or manual methods.

## 2.6 Control Point-based Deformation in 3D

In the previous section, we introduced feature definition on face and generic model adaption methods using feature points. Here, we introduce general deformation techniques where we deform a mesh based on a set of control points. The control points are corresponding to features points in the previous section. This type of deformation can be used for various objects, such as animation of clothes.

### 2.6.1 Free Form Deformation

Free-form deformation (FFD) was initially introduced by Sederberg et al. [27]. FFD is based on the idea of deforming 3D models by warping the surrounding space. Conceptually, the object is encapsulated in an imaginary, flexible and cubic control box called lattice which contains a 3D grid of control points. Such lattice defines a trivariate Bezier volume. In FFD method, the control point set around the object is moved which makes the lattice squashed, bent, or twisted and consequently deforms the shape of the object in a smooth manner. It is important that the objects are flexible so they can be deformed along with the lattice enclosing them. Surrounding the object by a single control box and applying changes to the entire object and some interconnected structures to deform the shape locally provides global and local shape manipulation, respectively.

It is noted that the shape topology might be preserved after FFD deformation; however, this method suffers from high computational cost due to calculation of new point position, aliasing artifacts and imposed constraint on the shape of lattice (rectangular structure). As a result, the expression of any surface point to deform relative to the lattice is limited. For instance, circular bumps on the surface can be hardly created using FFD. All the aforementioned drawbacks make

FFD not a suitable approach for deforming detailed 3D objects especially human face. Hence, a number of approaches are proposed based on FFD to advance the box–shaped lattices of FFD method to achieve greater generality in the shape and topologies of control lattice while the majority of FFD approaches preserve the notation of a volume trapped by the control lattice which encapsulates objects.

### 2.6.2 Extended Free Form Deformation

In 1990, Coquillart [28] introduced extended free form deformation (EFFD) as an extension of FFD which use arbitrary-shaped grids or a combination of grids instead of the parallelepiped lattice of FFD. This method provides the ability to edit the lattice by moving, merging and deleting the control points before associating the lattice with the object of interest. Therefore, it provides additional flexibility and controllability for shape deformation compared with regular cubic lattice used by FFD even though it is still limited by the expressibility of the FFD algorithm. It is also noted that surface continuity is preserved by this method even though lattice continuity might not. The considerable disadvantage of EFFD usage over FFD is higher computational cost.

### 2.6.3 Dirichlet Free Form Deformation

In preceding FFDs, the topology construction of complex lattice can be a relatively tedious and challenging task and it is almost impossible to assess the effect on resulting deformations by a given topology.

A generalized freeform deformation called Dirichlet Free Form Deformation (DFFD) is proposed by Farin [29] which combines FFDs and Dirichlet surfaces. It utilizes a generalized

natural neighbor coordinates known as Sibson coordinates [30] rather than the typical local coordinates (e.g. rectangular control box). It provides the ability to place control points at arbitrary locations including on the object, inside or outside of the object instead of locating on a regular lattice and subsequently provides greater flexibility. Since the control of deformation is done exclusively by the parameterization defined by natural neighbor interpolants and these functions cover singularities thus the results might suffer from unwanted deformation artifacts.

Lee et al. [10] used DFFD method in order to deform a generic mesh model based on feature points obtained from two pictures of a person.

### 2.6.4 Radial Basis Functions

Radial basis function networks [31] have become a well-established tool for all types of scattered data interpolation problems either in two or more dimensions. RBF method is the common choice for constructing implicit surfaces from scattered data and creating a range of facial expressions and animations [32] [33] [34].

Essentially, this deformation method is based on a point set selected from the surface of the object and named as centers. Any displacement of such centers is used for training of RBF networks and computing the weight vector. Afterwards, all the surface points are inputted into RBF network so that their new locations can be calculated. The choice of centers (their numbers and locations) is significantly important to achieve desirable deformed surfaces with high accuracy especially when the data becomes so dense. It is noted that RBFs consider smoothness of the surface as a minimal constraint and as a result, RBFs guarantee smooth geometric deformation particularly in facial deformations. Applicability of RBFs in almost any dimension is one of the prominent advantages of such method because in general, there are trivial

constraints on the way the data are defined and determined. Another advantage of using RBFs is its low memory consumption as only the centers, the radial basis functions and the computed weights are sufficient to represent the surface of the model.

## 2.7 Non-Feature based Shape Manipulation

Shape manipulation is very vital operation in many computer-assisted industrial and artistic design applications. It has been carried out also by deformation methods which don't rely on sets of feature points. Some of such methods used extensively in fields of geometric modeling and computer animation are described in this section.

### 2.7.1 Deformation using Eigen-based Approach

A novel surface deformation framework is introduced by Rong et al. [35] [36] based on the Laplace-Beltrami eigenvectors considered as the model signature which hold sufficient information in order to differentiate between models. Such framework performs mesh adaptation in the spectral domain which reduces the size of embedded linear system to a set of eigenvectors. Typically, a minimum number of 100 eigenvectors are required for representing a shape and increasing the number of eigenvectors makes the deformation result more realistic. Moreover, using a reduced set of parameters increases the computational speed in order to edit large and complex shapes. However, this deformation is mainly aimed to provide shape isometry by preserving Laplace-Beltrami operator which might sacrifice the surface smoothness through deformation. It is noted that the deformation transfer approach used to add the details back might cause artifacts in the final results of deformation. On the other hand, altering the scope of the deformation requires the user to move the locations of the restricted points.

While the aforementioned method is mainly focused on keeping mesh volume intact, a recent deformation method proposed by Dey et al. [37] is primarily focused on preserving mesh smoothness and preserving mesh volume has lower priority through the deformation. This approach utilizes a skeleton which is created implicitly by leveraging only 8 eigenvectors to navigate deformations. The skeleton is adapted by estimating a new set of eigen coefficients and the details are added back to produce the final shape of the mesh. Such approach directly manipulates all kind of meshes without solving any non-linear system. Therefore, it reduces computational time and enhances numerical accuracy. The advantageous aspect of this approach is that model deformation through stretching of local parts doesn't globally affect the model and is restricted to related areas. Hence, more natural result can be obtained. However, the utilized skeleton is a "high-level abstraction" of the model shape and it doesn't include trivial features and details and eventually affects the deformation quality.

## 2.8 Mesh Parameterization

Mesh parameterization is referred to the act of establishing one-to-one and onto mapping between any triangular mesh and a surface with similar topology which is called parameter domain [38]. Mesh parameterization has been playing an important role in a variety of applications in geometry processing and computer graphics including texture mapping, morphing, remeshing, compression, object recognition and detail transfer due to easier processing of parameterized mesh compared with irregular original mesh [39]. As a result, a large body of methods has been developed in order to deal with parameterization of meshes required by the aforementioned applications.

Praun et al. [40] introduced a method which enables the establishment of consistent parameterization for a set of models that are not needed to be geometrically close to each other. The property of parameterization consistency comes from relying on only one parameter domain and also the correspondences between features of different models. Based on the features which can be determined either automatically or manually, parameterization is achieved by partitioning of the mesh into patches and tracking patch boundaries which form a "net". It is also noted that connectivity of the net is topologically equivalent to parameter domain connectivity. Consequently, parameterizations are utilized for obtaining remeshes with equal connectivity. One of the applications of parameterization is various affine combinations of different subjects such as horse, human and cow using same parameter domain.

Depending on a simple common parameter domain, Kraevoy and Sheffer [41] present a method to directly establish a shape-preserving and cross parameterization between two mesh models. In contrast with the method of [40], this approach doesn't considerably rely on the shape of the patches. It is also noted that a sparse set of matching feature points on the two shapes are required to be manually defined by the user. While the feature point sets have manually been identified, automatically determining of a meaningful correspondence and at the same time being robust to both rigid and non-rigid geometric transformations is extremely challenging.

Although, the preceding methods are introduced for consistent parameterization of a mesh set containing topologically dissimilar meshes, due to existing large degree of intra-class variation, inconsistent mesh quality and topology in many real-world object classes, it is tedious or impossible to determine a single consistent parameterization among the class objects, unless the meshes are very alike [42]. Therefore, some methods have been developed to only deal with consistent parameterization of a mesh set in which meshes belong to specific object classes like

the work proposed by Blanz and Vetter [5] that is focusing on the faces as described in details in section 2.3.1.

Another example is the work introduced by Allen et al. [43] which is focusing on the consistent parameterization of human bodies captured by a laser scanner. While all the bodies have the same overall structure in this method, they might contain holes and might be considerably different in shape. By matching a set containing 70 marker constraints, parameterization is accomplished through mapping of bodies to a template human mesh and consequently the scanned meshes are completed. Such solution is constrained to very specific inputs and can cause severe approximation errors if the input models considerably differentiate in their geometry.

## 2.9 Multi-resolution Modeling

A method used in order to encode an object with a wide range of levels of detail is called multi-resolution modeling. Such method can be used to reconstruct the object with any one of those levels on demand according to the existing challenges that should be properly handled such as rendering, storage, transmission, modeling and editing.

Lee et al. [44] present a multi-resolution surface representation called displaced subdivision surface which expresses a detailed surface model as a scalar-valued displacement over the smooth subdivision surface. Similarly, Guskov et al. [45] introduces Normal Meshes in which multi-resolution mesh subdivision is relying on the displacement of vertices by a single scalar value in the normal direction. The popularity of the preceding methods is because of easier modification and/or storing of a single scalar value.

In 2007, Lee and Soon [46] introduced an efficient method to raise the resolution of any scanned face at a low resolution by applying 3D skin from a higher-resolution face in the database. The resolution of obtained result is extremely high so that the face can be used in a variety of applications including computer-generated films. Initially, they utilize a large 3D database called CAESAR as the low-quality scanned database. While such database consists of a large number of human models and truly captures global shape of the human heads, models are almost imperfect due to existence of horizontal sliding lines, missing points and holes specially on the face and ears areas. High-resolution database consist of four human subjects scanned with closed eyes and mouth and they all have different resolution and size. It is noted that only face region is retained in their methodology and the other body parts of models including ears, neck and head back are discarded. Each face model is constituted of very large number of triangles (millions) which make the fine and unique details such as pores clearly apparent. Next, same mesh parameterization is created between all the face models either high or low resolution. For such purpose, a base model is built according to a two-step adaptation approach. Initially, a global surface approximation is attained by adapting the generic model to the high-quality scanned model using RBF networks and facial feature points. These facial features are detected on the model surface in a semi-automatic manner from front and side views. The noise contained on the surface of low quality models is eliminated with transition to base model. Moreover, smoother surface is obtained using Loop's subdivision technique [47] which raises the number of mesh triangles. Then it is followed by a second adaptation which utilizes a cylindrical projection to enhance the surface approximation. After that, a 3D skin bank is created by calculating scalar offset field between each high-quality scanned model and its corresponding base model. These skins can be applied to generated base models of CAESAR database to create models with high-

quality level. Applying different skins per each subject lead to generating a larger dataset. As the result, this method provides the user the ability to control the resolution of current model as desired with time, cost and resources efficiency.

One of the drawbacks of the preceding technique is that only the facial region of each individual is captured and reconstructed, while full human head is typically used in many applications. On the other hand, facial expressions of the resultant model are constrained due to closed eyes and mouth in high-quality faces. Moreover, the low quality models might have open eyes while the high-quality faces used to create the 3D skin bank have closed eyes and consequently affects the augmentation result.

## 2.10 Face Animation

Face animation can typically be achieved either using physically-based animation methods which move the skin surface relying on the muscle anatomy of human face or methods which can ignore the underlying structure of the face and focus on capturing the surfaces from the outside of the faces, mostly using video cameras.

### 2.10.1 Performance Capturing for Modeling and Animation

These methods use a set of cameras to capture facial movement and animate the face either online or offline. Such methods can be classified in two main categories: marker-based methods and markerless methods. Marker-based methods utilize some easy-to-detect markers stuck on face and are focused on capturing the motion of markers to interpret the face moment. On contrary, markerless methods do not use any markers, but they leverage more sophisticated image processing methods to extract the face motion.

#### 2.10.1.1 Marker-based Techniques

While 3D scanning technologies described earlier are commonly used to capture high-resolution static facial geometry, motion capture systems are predominant technologies for dynamic acquisition of facial geometry in term of acquisition speed. Hence, these two acquisition tools are basically complementary to each other.

The idea of tracking facial performances in a video sequence and translating these into animation control parameters of a 3D scanned face started from the work introduced by Williams [48]. Based on that, many works [13] [49] have been proposed in the consecutive years to robustly and accurately track a set of facial markers of a real performer and use them to animate a scanned 3D facial geometry. The main drawback of aforementioned techniques is that they don't have the ability to acquire small details such as nose wrinkles and skin mesostructure manipulation. In 2011, Huang et al. [50] proposed a system which leverages motion capture and 3D scanning technologies to inherit characteristics of both technologies. As the result, high-fidelity facial performances of individuals are acquired. Meanwhile, this method makes it possible to capture realistic dynamic wrinkles and the facial details which are subtle but perceptually important. Initially, dynamic facial movements of an actor are recorded using the motion capture system [51] which is based on 100 retro-reflective markers placed on the face. Meanwhile, a video camera is synchronized with the marker-based motion capture system to acquire the images of corresponding facial expressions. The images serve as references for scanning static facial movements of the actor. Then, the motion captured data are automatically evaluated by an analysis algorithm to define a minimal set of high-quality facial scans needed for perfect facial reconstruction. Thereby, such automatic process enhances 3D facial reconstruction

efficiency in terms of facial accuracy and scanning time and effort. Afterward, corresponding synchronized reference images are employed as reference facial performances for scanning of the high-quality facial models. A laser scanner is utilized to acquire high-quality static facial meshes of an actor. Next, motion capture markers are registered to all the facial scans since such data would be combined to reconstruct high-quality faces. It is followed by a two-step registration process including large scale (overall face shape) registration and fine scale (skin mesostructure) registration to automatically obtain dense, consistent correspondences through all the facial scans. Inaccurate establishment of correspondences would cause undesirable visual artifacts in the captured facial expressions due to existence of high-frequency details in facial faces. Hence, the user interaction is provided within the system to manually adjust the correspondences. After all, motion captured data are accurately combined with the minimal set of facial scans in the blendshape interpolation framework to efficiently reconstruct high-resolution 3D facial performances. It is noted that a higher quality 3D scanning system like XYZ RGB systems [52] can be employed to enhance the resolution of constructed facial performances. The problems with such method are uncertainty in synchronization between eyes and lips movements which is vital for creation of convincing facial animation and also not modeling the entire head of the actor.

While the preceding approaches attain much success, the acquired performance is challenging to be used over again so a new performance is needed every time when a new sequence of speech or animation is made. Leveraging a sparse set of facial markers not only puts a restriction on the movement and location for construction process, but also might call for manual adjustments or physical deformation constraints to recreate the fine scale dynamics.

#### 2.10.1.2 Markerless Techniques

Recently, a passive multi-view stereo system is proposed by Beeler et al. [53] which captures expressive facial performances with high-resolution geometric details based on anchor frames. The process begins with capturing a sequence of user's images utilizing seven cameras while keeping the illumination uniform. Each frame in the sequence is processed independently to approximate the corresponding initial mesh using the 3D reconstruction technique described in [54]. Next, one frame is chosen by hand as the reference frame and frames with higher resemblance in terms of expression and orientation are automatically identified and considered as anchor frames. According to occurrence of anchor frames, the whole sequence is subdivided into clips of frames which eventually eliminates issues related to processing of sequential motion and also provides parallelization of the calculation. Then, an image-space tracking technique is carried out to track pixel correspondences directly from the reference frame to every anchor frame and then to the non-anchor frames of the sequence in an incremental order. According to achieved results of image-space tracking, a reference mesh is propagated to the entire sequence to calculate the transformed 3D positions of the reference mesh vertices for all frames. Lastly, a mesh refinement technique is imposed on the preliminary propagation to establish consistency with image data while employing spatial and temporal priors to obtain coherent mesh surfaces in the whole sequence.

In general, the preceding system passively creates 3D facial expressions with high-resolution details and also represents robustness in the presence of fast motions. However, selection of only a frame as reference might not produce a proper distribution of anchor frames in the entire sequence. Therefore, the profits of reconstruction based on anchors are vanished in some cases.

The other drawback of this technique is that the frames in image space are not identified as anchors unless they have not only similar expression but also similar head orientation with respect to the cameras. Moreover, such technique is focused on modeling the frontal face part of the actor.

Another noteworthy example of markerless approaches is a system proposed by Weise et al. [55] which enables real-time performance-based character animation. Primarily, the system involves an offline process which initiates with a sequence of facial expressions performed by the user. These expressions are captured using the Kinect sensor as the training poses. Multiple scans are combined for each expression over time to improve the surface smoothness and also reduce noise level. Then, user specific expressions are reconstructed using a generic template model adapted with non-rigid registration methods. Registration accuracy can be enhanced by manually marking feature points as additional texture controls in the mouth and eye regions. Afterwards, example-based facial rigging introduced by Li et al. [56] is used to create the full set of user's blendshapes. Finally, the digital avatars are animated online using the blendshape weights calculated by tracking algorithm which matches a user's specific expression model to the captured 2D image and 3D depth map. In addition, a statistical animation prior estimated from available blendshape sequences is leveraged to normalize the tracking. Temporal coherency is obtained by evaluating a window of successive frames.

While this technique provides plausible real-time tracking of user and animating 3D avatars, not only it doesn't capture high-quality geometric (small wrinkles) and motion details of the user but also it fails to correctly acquire the lip movements. As a result, the reconstructed and performed expressions would be different. Moreover, the resolution of captured data is degraded by using the wide-angle lens of the Kinect. The other drawback using this technique is that the

blendshape and priors embedded in the system lead to produce a limited range of expressions for the user.

Excluding training session for facial expression construction, a real-time facial performance capture system has been presented [57] which is specifically suitable for face retargeting application. Such system is built on a run time shape correction approach for combined video and depth data input captured by Kinect sensor. The process initiates with only a single face scan of the user in a neutral expression. Like the technique described in [55] few consecutive face scans are combined to attain a higher quality 3D point cloud of the neutral performance. Then, a set of primary personalized blendshapes are automatically created as rough approximations of the user's generic expressions. From there, the user's facial performance is tracked by solving for the best fit to the input data using the generic blendshapes and leveraging global rigid transformation, 2D facial features constraints (obtained from Live Driver [23]) and Laplacian deformation algorithm. The process continues by mapping onto an adaptive principal component analysis (PCA) space including anchor shapes and extra corrective shapes and guarantees and enhances the fitting accuracy through the process. The corrective shapes are trained by accumulating new expression models that don't belong to the leveraged adaptive PCA space and such models are utilized to refine the corrective shapes using the incremental PCA method. The final result in the PCA space needs to be mapped back to blendshape space for the sake of retargeting using the technique described in [56]. However, due to calibration free characteristic of the system, the initial blendshape coefficients are used for performance retargeting. Hence, small scale and high-resolution details acquired by correctives are not shifted to the desired avatar. While the proposed facial animation system is highly deployable and easy to use, not only it suffers from limited expressiveness of blendshapes, but also the accuracy of embedded

2D feature tracking technique is affected by user's movement and consequently affects the result of its following processes.

### 2.10.2 Physically-based Animation

The preceding techniques described for surface deformation neglect the underlying physics of object deformations. On the other hand, it is essential to have models with accurate and realistic behavior in applications such as the surgical training, facial animation. Hence, physically-based surface deformations were introduced and developed repeatedly in the last decades. In comparison with geometric deformation methods, physically-based methods usually provide more realism in the animation of the objects and characters because of considering inherent physical properties in the surface.

Zhang et al. [26] introduces a method for creating a personalized 3D facial model based on a generic control model [58] developed with the purpose of physically-based facial animation. In general, they model the shape of the face and derived facial animations from the physical behaviors of three major structural components. These components include skin layer, muscle layer and skull layer. The skin layer is represented by a triangulated mesh and attached to the muscles layer to provide animation control. The anatomical base for facial muscle attachment is provided by the corresponding skull (triangulated mesh) which is only used for creating muscles during the initialization of the animation system. The components are modeled for eyes and teeth distinctly to improve the general realism.

Overall, physically-based methods may lead to greater realistic animations in contrast with the purely geometric approaches; however, because of their high computational cost, they might not

be suitable for real-time application and geometric techniques remain to be predominantly used methods in the animation industry because of their generality and control.

## 2.11 Mesh Repair

3D mesh models can be generated from various sources such as 3D range scans and typically contain a number of defects and flaws. Mesh flaws can mainly be classified into three categories based on local connectivity including isolated vertices, dangling edges and singular edges, global topology including topological noise and orientation and geometry including holes, gaps, degenerate elements and self-intersections. These flaws make mesh model fail to meet some specific quality requirements and geometric criteria important for most practical applications. Hence, mesh models cannot be directly used and repairing such flaws in order to obtain required compatibility highly becomes an essential task. On the highest level, we differentiate between repairing methods that leverage a global strategy and approaches that utilize a local method.

### 2.11.1 Global Repairing Methods

Global repairing methods are usually built on remeshing of the entire input that guarantees global correctness and robustness, but they generally cause loss of sharp features. One of the early works on globally cleaning up mesh models is the one proposed by Murali and Funkhouser [59] which constructs manifold surfaces and solid regions. This method forms a BSP tree from the triangle soup (which is a group of unorganized triangles) by dividing the volume into cells and then solidity likelihood is automatically calculated for each cell. However, BSP-trees creation for large or noisy inputs (containing significant hole) is time consuming and error prone.

On the other hand, the output mesh might considerably differentiate from the input in the presence of geometric defects even for small input errors.

Nooruddin and Turk [60] proposes a volumetric approach which is based on filtering (using parity counting) and ray stabbing. This method extracts a consistent, watertight model from volumetric representation obtained from the triangle soup conversion. One of the drawbacks of such method is that it fails to properly scale for large, detailed models and makes interpolation of input mesh complicated due to its voxel-based characteristics. Therefore, it might lead to generation of additional handles in regions situated further from the existing ones. Arbitrary polygonal soups can be repaired by employing a volumetric algorithm [61] which depends on hole patching and parity counting. While, such approach makes recreation of sharp features in the original geometry possible, it tends to fail in the presence of significant gaps. Also, it is not suitable for extrapolating the missing surface at holes for human figure due to its dependency on location and size of the holes. In order to fill holes on the surface of mesh, some outstanding works [62] [63] are developed based on normals plus diffusion and line-of-sight respectively.

### 2.11.2 Local Repairing Methods

Local repairing methods are based on manipulating the mesh only in the neighborhood of each flaw. While these approaches are less invasive in contrast with global methods, they typically guarantee handling only few mesh flaws. In order to close the gap on the mesh surface, the methods introduced by Sheng and Meier [64] as well as Barequet and Kumar [65] progressively perform “zipping” pairs of boundary edge chains. While such methods provide better control over topology, new degeneracies and intersections are potentially introduced.

Bischoff and Kobbelt [66] propose a repairing algorithm on CAD models. This method provides manifold patches by applying input division. The resulting patches do not self-intersect and they are sewed together to create a polygonal mesh. A spatially-varying threshold is essential to be determined for gap filling purpose in such method. Potentially, new degeneracies are introduced using preceding technique. To specifically solve self-intersection defects, Campen and Kobbelt [67] present a method based on an intermediate plane-based BSP representation of input mesh. Such technique poses two requirements on input including being a closed mesh and being free of degeneracies.

A semi-automatic hole-filling approach is proposed by T.-M. Ngo and Lee [68] which segments complex holes into smaller and more planar holes in order to properly handle holes and repair the mesh model. This method not only treats detected holes locally, but also tries to recover fine and sharp features in the original mesh model. It is noted that user intervention is leveraged for achieving results with higher accuracy.

Recently, an automatic algorithm is proposed by Attene [69] which applies several sequential repairing algorithms to locally manipulate the mesh as less as possible. First, it converts the input mesh to an oriented manifold model and then it closes all holes and removes all degenerate faces. Such method properly handles not only self-intersection defects on the mesh surface but also degenerate faces and holes. The preceding method poses no requirement on the input mesh and need no user-specified parameter. In contrast with related available algorithms, it is computationally efficient and resultant meshes are more accurate while using smaller number of triangles.

On the other hand, surface reconstruction techniques [15] [70] [71] can be employed as alternative methods for cleaning up the mesh in order to make it consistent and watertight. However, majority of such algorithms are designed to reconstruct surfaces from point cloud inputs while handling different levels of included noise. The structure of input mesh is not guaranteed to be preserved by such algorithms. In other words, sharp features of input might be lost and partial or complete input remeshing might happen. The aforementioned issues make surface reconstruction methods not quite suitable for attaining a clean, watertight mesh.

# Chapter 3. Methodology

This chapter explains in details the techniques used to not only generate animated heads of each input face, but also convert all input models into one mesh structure. Therefore, a consistent parameterization is established between models while preserving the individual shape and expression using a given generic mesh, which contains animation structure.

Our methodology starts with describing integrated generic mesh and utilized input models in the system and the way they have been captured and collected. The input model pool contains head models which may vary in face shape, level of detail and expression and may also contain defects like holes under the neck, missing parts like ears, back of the head, etc (Figure 3.1).

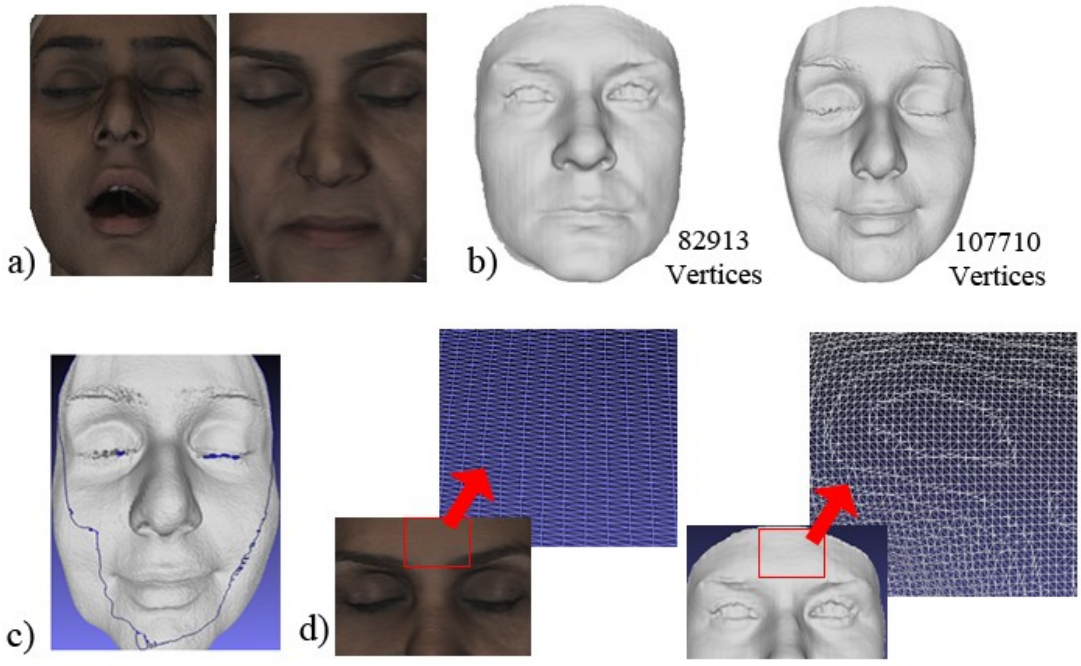


Figure 3.1: Differences between meshes due to a wide variety in a) expression, b) resolution, c) mesh defect, and d) mesh structure.

After collecting data, 3D surfaces are reconstructed and registered to a generic mesh. Then, a two-step adaptation is carried out which initiates with mesh manipulation based on feature points using a modified RBF-based deformation and it is followed by a subdivision approach and then the second mesh adaptation based on surface points. Finally, the quality of the mesh is improved further by some additional post refinement methods and an automatic repairing method.

## 3.1 Generic Model

In the context of face modeling and animation, generic model is referred to a triangular mesh with the features that are properly defined so that high-curvature regions contain fine triangles and low-curvature areas contain larger triangles. Such model forms the basis for precise feature-based adaptation to the input model. The generic model integrated in our system is created using a commercial software called “Autodesk 3DS MAX”.

Figure 3.2: Generic model (12600 triangles).

This model contains a fair number of vertices and triangles (6420 and 12600 respectively). The generic mesh is modeled in a way that the eyes are closed and the upper lip and lower lip are separated from each other as illustrated in Figure 3.2. Therefore, highly accurate match can be

obtained between those regions in the generic model and the corresponding ones in input model due to various facial expressions, shapes and details carried out in our input model pool.

## 3.2 Input Model

Input model pool consists of the individuals' faces which either we captured them by using the Cyberware laser scanner [72] or collected from a publically available database (Bosphorus database [73]) and we want to test these models as the system input for our experiments. In contrast with generic model, input models have higher resolution, while they might have defects, noise errors and visual artifacts.

### 3.2.1 Laser-scanned Model

Each subject of interest can be acquired as a dense 3D mesh model using Cyberware head and face color scanner [72] which contains a 3030/RGB digitizing head and the PS motion system. It includes a rugged and self-contained optical range-finding system adapted to changing lighting conditions and surface characteristics due to its dynamic range. The scanner can be typically used without any adjustment by the user. The data captured by the scanner is represented as a triangular mesh and contains fine details such as wrinkles, pores, etc. In this work, we have created a collection of scanned heads consisting of three subjects aged between 23 and 37. A complete 360 degree model of the head for each subject is captured while the subject is performing neutral expression and different expressions such as happiness, surprise, etc. Figure 3.3 shows a sample captured head containing 362918 vertices and 723614 faces. Since all of the subjects had long hairs which caused abnormal shapes for the back of the heads under the hat, the regions except the frontal face parts were removed.

Figure 3.3: Scanned face of an individual (723614 triangles).

As it is mentioned in the literature, one of the drawbacks of using scanning tools is that scanned data typically includes holes especially around chin, nose and eyes regions due to occlusion or being a black area. Therefore, we need to utilize a method which properly reconstructs the entire surface of a given polygonal model and eventually fills the holes.

### 3.2.2 Publically Available 3D Face Model

In recent years, the number of range scanners has increased rapidly. However, many researchers do not have access to scanning equipment and services or dense polygonal models. Hence, public repositories such as Bosphorus [73] are built to make either large or small range database available to the public.

The Bosphorus database is aimed for research on human face processing tasks (either in 3D or 2D) including facial action unit detection, expression recognition, facial action unit intensity estimation, face recognition under adverse conditions, deformable face modeling and 3D face reconstruction. Bosphorus is a rich and unique database because it consists of 105 subjects (60 men and 45 women) and about one third of subjects are professional actors/actresses. The majority of the subjects are Caucasian and aged from 25 to 35. Various poses (including Systematic head poses), expressions (Figure 3.4) and occlusion conditions such as beard &

moustache, hair, hand and eyeglasses are available per subject. In overall, there are 4666 faces in the database.

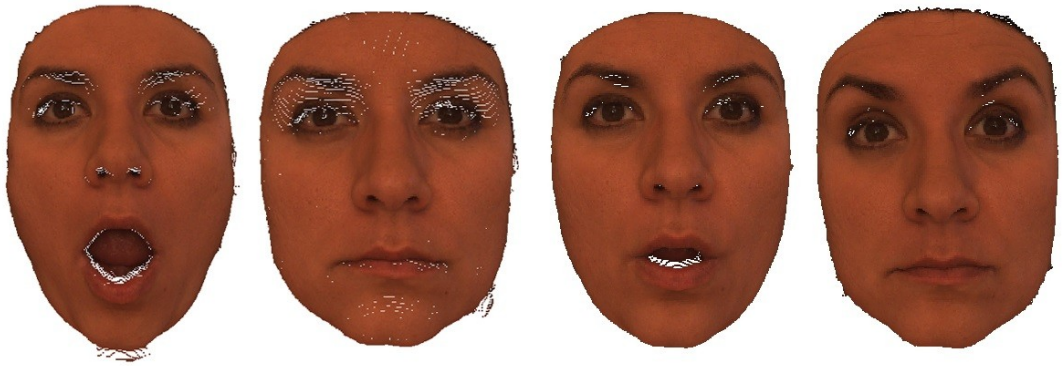

Figure 3.4: Various facial expressions are performed by a female subject.

To acquire facial data, a structured-light based 3D system has been used. Acquisitions are single view and there should be a distance of about 1.5 meters between the location that the subject sits and the 3D digitizer. The resolution of the sensor in x, y & z (depth) dimensions are 0.3mm, 0.3mm and 0.4mm, respectively and color texture images have high-resolution (1600x1200 pixels). Homogeneous lighting is required in order to have good quality texture images and it can be obtained by a 1000W halogen lamp used in a dark room.

Bosphorus offers the users 3D coordinates with corresponding 2D image coordinates which can also be used in texture mapping stage. High-resolution color images just contain face area. However, mesh connectivity are not provided and only point clouds of individuals are available. Hence we need to utilize an algorithm which properly reconstructs the surfaces of given point clouds and transfer it to a triangular mesh.

## 3.3 Surface Construction from Points and Filling Holes

The data in our input model pool is represented either as point set, which doesn't contain mesh connectivity, or triangulated mesh, which typically contains errors and holes due to sensor noise and occlusion, respectively. After evaluation of some works proposed on reconstruction of the surface from point samples, we decided to use Poisson reconstruction method [15]. This method merges the advantages from both global and local fitting schemes which results in higher flexibility and tolerance in the existence of imperfect data and eventually reconstructing high-quality surfaces.

As displayed in Figure 3.5, the surface of input 3D point cloud is completely built using Poisson reconstruction method and it is converted to a triangulated mesh. After running some tests on different Octree levels and observing quality of the results, we decided to set 10 as the maximum and most suitable value of Octree depth so that the surfaces can be reconstructed with high level of details.

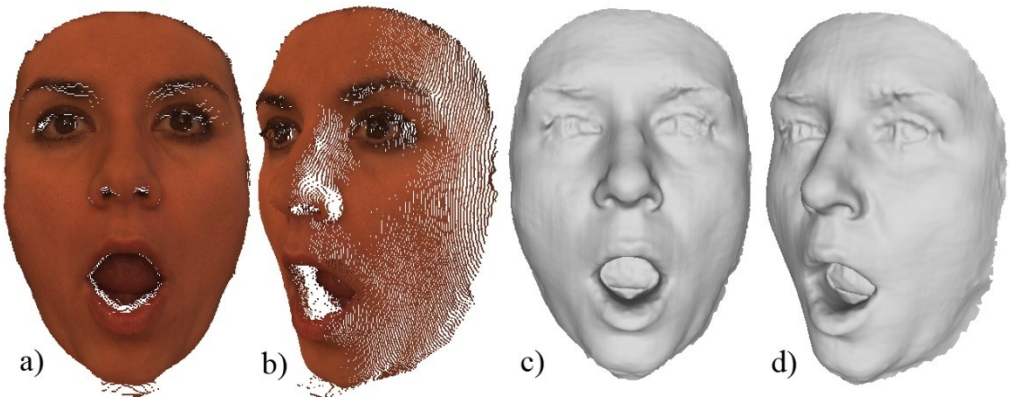


Figure 3.5: Surface reconstruction using Poisson reconstruction method. a-b) Different views of original point cloud, c-d) different views of reconstructed surface.

## 3.4 Registration of Input Face in Generic Face Space

Since 3D surfaces reconstructed from our input models and the generic model are not in the same location and orientation in the space if the deformation approach is applied on a generic mesh based on the initial positions of feature points, the results will not be satisfactory as we expect. Hence, Procrustes Analysis method [74] [75] [76] is adopted to superimpose input mesh models to our consistent generic mesh and consequently minimizes the distance and reaches the optimal alignments among them. This approach is based on matching corresponding facial features from each of the two mesh models (input and generic model). Nine facial features are manually selected on the snapshots taken from frontal view of both models as illustrated in Figure 3.6. These features are mainly distributed on the eyes corner, lips and nose. The dimensions of points are increased from 2D to 3D by using back-projecting method. Then, the least-squared error rigid transformation between feature sets of the generic mesh and the input model is calculated. Such transformation is usually composed of uniform scale, rotation and translation and applied to the input model and as a result, a more precise fit is achieved.

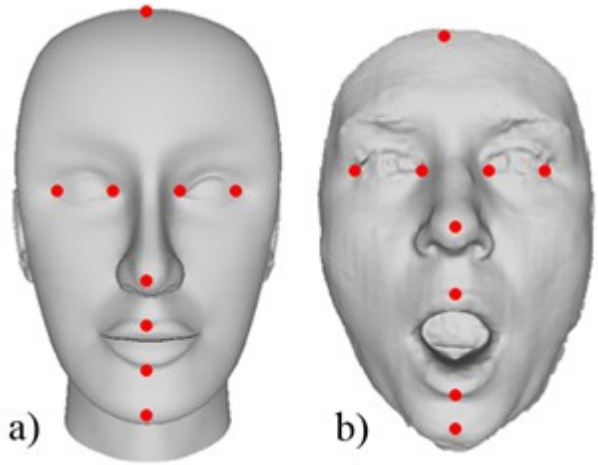


Figure 3.6: Facial feature points on a) the generic mesh, b) the original input face.

Figure 3.7 displays generic mesh and input face before and after registration process. In overall, a fast registration between a large set of input models and the generic head model is obtained by Procrustes Analysis method.

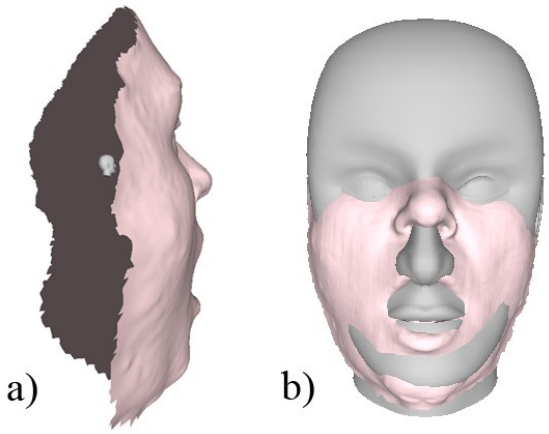


Figure 3.7: Registration process using Procrustes Analysis. a) The generic mesh (shown in white) and the original input face (shown in pink) before registration, b) the input face and the generic mesh after registration.

## 3.5 Feature Points Extraction

The deformation functions utilized in this work (which are described in section 3.6.4) are based on two point sets attained from the surfaces of the generic and the input models. The definition of feature points is very critical in order to obtain desirable results after model deformation. As more points are utilized, deformation control and fidelity are amplified.

Hence, a set of 128 characteristic points which are semantically well-defined are extracted from snapshots taken from frontal view of both the generic and the input models. Feature points extraction is accomplished by leveraging a semi-automatic method due to its advantages over manual and automatic ways. Initially, a 2D mask of feature points is created and predefined on

the snapshots of both models with more points defined around important facial components such as nose, eyes and lips so that such components can properly be captured. In 2D mask creation, we are inspired by feature points designs (for the numbers and the locations of the feature points) defined by the works done by Lee et al. [77] and Lee and Magnenat-Thalmann [10]. As illustrated in Figure 3.8, 20 of 2D mask points are key feature points shown in yellow and the rest are non-key feature points shown in red. The 2D mask is fitted to the snapshot first by only adjusting key points to the correct positions by the user. Then, non-key points are automatically transformed to new positions depending on location of their corresponding key points. This automatic process which reduces the amount of time and effort required to select the feature points is accomplished using an affine map.

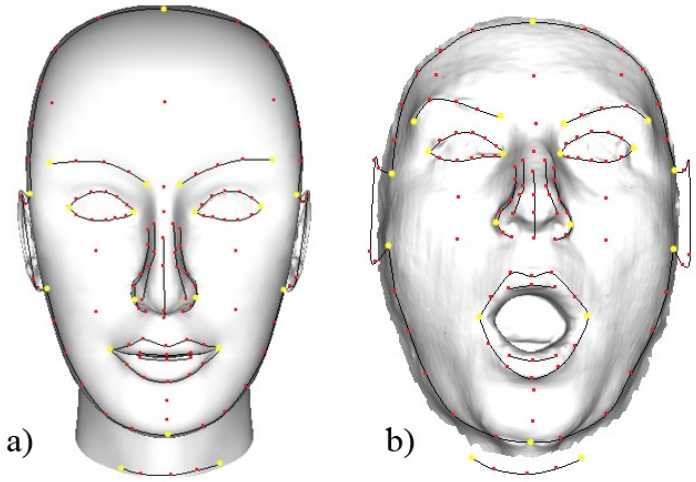


Figure 3.8: 2D distributions of 128 feature points on a) generic model and b) input model.

An affine map is a transformation which preserves points, ratios of distances between points lying on a straight line and straight lines and sets of parallel lines will continue to be parallel with each other after the transformation. However, it does not necessarily preserves angles between lines, lengths or distances between points. An affine transformation is a composition of rotation,

translation, scale and shears. A 2D affine transformation for a given point ($x_1$, $y_1$) has the following general form.

$$T = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} ax_1 + by_1 + c \\ dx_1 + ey_1 + f \\ 1 \end{bmatrix} \tag{3.1}$$

$$x'_1 = (ax_1 + by_1 + c) \quad y'_1 = (dx_1 + ey_1 + f)$$

Where T is the affine transformation matrix and a, b, c, d, e and f are transformation coefficients. $x'_1$ and $y'_1$ are new calculated values for the given point. Transformed values of two control points (($x_1$, $y_1$) and ($x_2$, $y_2$)) can be calculated using only four coefficients as below.

$$x'_1 = (ax_1 + by_1 + c) \;\; y'_1 = (-bx_1 + ay_1 + f) \tag{3.2}$$

$$x'_2 = (ax_2 + by_2 + c) \;\; y'_2 = (-bx_2 + ay_2 + f)$$

Based on known transformed values of two control points, the coefficients of corresponding transformation matrix are computed by solving above four equations for four unknown variables (a, b, c and f). After that transformed values for other points affect by those two control points can be calculated. After adjusting the 2D mask, 3D positions of feature points are recovered automatically by back-projecting them onto the surface of the object.

On the other hand, some parts of the head in the input model are missing or removed such as the back part of the head and the ears due to the existence of errors and artifacts. Hence, a virtual back of the input head is created by modifying extra 10 feature points estimated on the back of a generic model as shown in Figure 3.9. These points on the generic model are calculated depending on a feature point located in the middle of the forehead in a way that they all have same values along X-axis but varying values along Y-axis. Then, corresponding depth for each point is computed as the furthest. Since the projected 2D points have slightly different values

than the original points of the 3D model, the closest correspondence of the desired point on the forehead is found among 3D model points as a point with the minimum distance. Having two 3D points (($x_1$, $y_1$, $z_1$) and ($x_2$, $y_2$, $z_2$)), d is calculated as distance in-between those points using the following formula.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{3.3}$$

Feature points located at the back of the input model are calculated by multiplying 3D correspondences in the generic model by the 2D distance ratios between 10 points on the snapshots of generic and input models from the side view. By default, the mesh inherits the back of the head, neck and ears from generic model after feature-based adaptation step.

A symmetric edition for the eyes is also provided within the system. Once the user applies changes to the location of feature points on one eye, the corresponding feature points on the other eye are changed accordingly which eventually reduces the amount of time we spend on feature point selection.

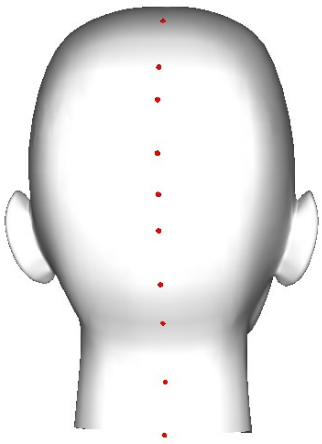

Figure 3.9: 3D feature points at the back of the generic model.

## 3.6 Feature-based Generic Head Adaptation to Input Model

Calculated 3D positions of feature points on the input and generic models in previous section are employed to deform the generic mesh using RBFs method to match the global proportions of the individual's face. It is important that the 2D frame of the feature points is adjusted properly. Otherwise, the result of deformation won't be satisfactory in some cases and feature points selection should be performed again.

Since the input model might be representing an expression like surprise and fear which typically involves moving and separating of upper lip and lower lip, special care has to be taken around mouth region while performing deformation method. Fitting of the generic mesh to an animated input model by only using RBFs deformation method without considering any regional information creates a model with unacceptable shape around the mouth region.

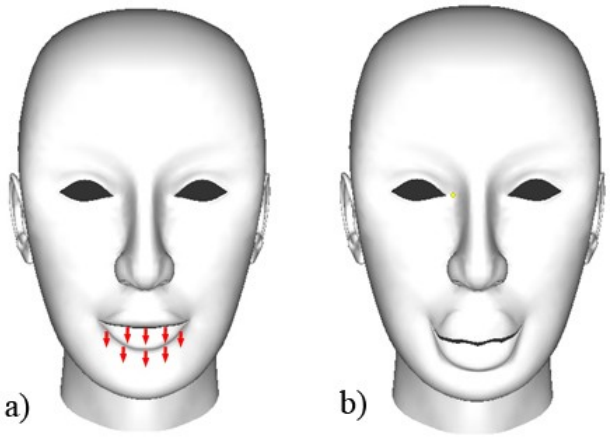


Figure 3.10: A sample of using RBF functions without applying any regional information. a) Generic model before deformation plus illustrated direction of feature point displacements, b) deformed generic model representing unpleasant deformation result around the mouth.

The red arrows in Figure 3.10 show the direction in which the feature points on the lower lip are displaced toward that direction and also the approximated surface by using only RBFs deformation method.

Hence, further actions are taken in order to solve such problem without engaging any mesh modification step in addition to feature-based manipulation. The main idea is to make correct division between regions related to upper lip and lower lip and associate this regional information with deformation functions. As a result, a wide range of facial expressions can be recreated from our input model pool. The steps involved in mouth movement simulation are listed below:

- The points of generic mesh located at the contours of the upper lip and lower lip are determined.
- Two separate predefined frames of points are provided and can be automatically adjusted according to their corresponding feature points.
- The points obtained in the preceding steps are separately combined for upper lip and lower lip to form two masks enclosing the regions of influence.
- All the points of generic mesh and the feature points embedded in the influence regions enclosed by the masks are determined and indexes are set to them depending on their associated regions.
- RBF networks are trained and corresponding weight vectors are calculated first for all features points except those in upper lip region and second for all features points except those in lower lip region.

- Finally, the surface points are deformed depending on their region indexes and corresponding weight vectors.

The following subsections delineate the details of the aforementioned steps.

### 3.6.1 Lips Contour Detection

Initially, the search space on the generic mesh is reduced by using 4 of the feature points illustrated in Figure 3.8 and located on cheek and jaw so that those feature points enclose the mouth region. Thus, only the surface points located in-between such feature points are evaluated and the rest of points would be ignored. Afterward, the next step is to determine the contour points of upper lip and lower lip called lower-opened and upper-opened, respectively.

A vertex is considered to be upper-opened (a contour point of lower lip) [78] if all polygons having the vertex intersect with the line that is parallel with Y-axis and pass through the vertex under the vertex. Also, a vertex is considered to be lower-opened (a contour point of upper lip) [78] if all polygons having the vertex intersect with the line that is parallel with Y-axis and pass through the vertex above the vertex. (Figure 3.11)

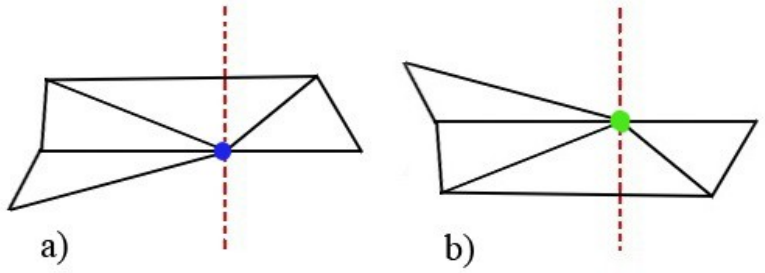


Figure 3.11: Sample representation of a) lower-opened vertex (shown in blue), b) upper-opened vertex (shown in green).

Based on the preceding conditions to determine whether a vertex is upper-opened or lower-opened, an equation of the line passing through the vertex and also parallel with Y-axis is formulated and the intersection vertices between the line and all the triangles containing the vertex are calculated in 3D space. However, this solution is not always reliable and accurate since the line might not intersect at any other vertex except the candidate vertex shared between all the triangles due to 3D characteristic of the surface. Therefore, previous conditions should be evaluated in 2D space by utilizing cylindrical projection method which is described in details in section 3.7.2. At first, each vertex of the generic model is transformed from 3D Cartesian coordinates (x, y, z) to cylindrical coordinates (u, v) using the following formulas.

$$u = \arctan\left(\frac{x}{z}\right),\ v = y \tag{3.4}$$

After transformation of all the vertices to 2D space, for each 2D vertex X, the location of two new vertices ($X_1$, $X_2$) having the same u-value as X but different v-value, respectively lower and higher values with very small difference, is evaluated by calculating barycentric coordinates (α, β). Having an intersection point $X_1$ and a triangle (A, B, C) on the generic model, barycentric coordinates with respect to $X_1$ and vertices of the triangle are calculated as

$$\text{invDenom} = \frac{1}{(v_{AC}.v_{AC})*(v_{AB}.v_{AB})-(v_{AC}.v_{AB})*(v_{AC}.v_{AB})} \tag{3.5}$$

$$\alpha = ((v_{AB}.v_{AB}) * (v_{AC}.v_{AX_1}) - (v_{AC}.v_{AB}) * (v_{AB}.v_{AX_1})) * \text{invDenom}$$

$$\beta = ((v_{AC}.v_{AC}) * (v_{AB}.v_{AX_1}) - (v_{AC}.v_{AB}) * (v_{AC}.v_{AX_1})) * \text{invDenom}$$

The vertex $X_1$ is considered to be located inside the triangle if (α, β >= 0) and (α + β <= 1). After calculation of barycentric coordinates and determining whether $X_1$ and $X_2$ are inside the

triangles containing the vertex X or not, there are three possible scenarios to decide whether a vertex is upper-opened, lower-opened or not, as follows:

- The first scenario is that both vertices are inside the triangles containing the vertex X which means that the vertex X is not located along the contours.
- The second scenario is that only the vertex with higher v-value $X_2$ is inside at least one of the triangles containing the vertex X which means that the vertex X is located along the contour of the upper lip.
- The third scenario is that only the vertex with lower v-value $X_1$ is inside at least one of the triangles containing the vertex X which means that the vertex X is located along the contour of the lower lip.

Using the aforementioned algorithm, 21 vertices along the upper lip contour and 25 vertices along the lower lip contour are detected, converted to 3D space using back-projecting approach and then sorted based on the values along X-axis.

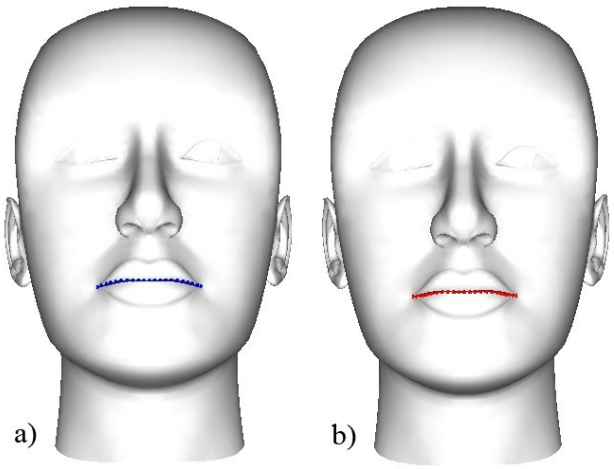


Figure 3.12: Contour vertices on a) upper lip and b) lower lip.

In order to verify positions and numbers of points, commercial software such as “Meshlab” and “Autodesk 3DS MAX” are used. Figure 3.12 shows the contour vertices on the upper lip (shown with blue dots) and lower lip (shown with red dots).

### 3.6.2 Regional Mask Creation

Manual segmentation of the mouth area into distinct regions is a difficult and tedious task. Therefore, two 2D frames of points are implicitly predefined on the frontal snapshot of generic model as illustrated in Figure 3.13. These frames are dynamically designed and can automatically be adjusted according to their corresponding feature points using affine transformation described earlier. So, this automatic process removes the need of user interaction for segmenting the faces into regions and eventually saves a great deal of time and effort. After the frame adjustment, points of lip contours and 2D frames are separately combined for upper lip and lower lip in a clockwise order to form two masks enclosing the influence regions by the mouth. Constructed masks form irregular polygons with large number of sides. It is important that the points of the masks to be ordered in either clockwise or counter-clockwise manner; otherwise it might lead to obtaining wrong solutions. Afterwards, points of the masks are back projected onto the model surface to attain their 3D values.

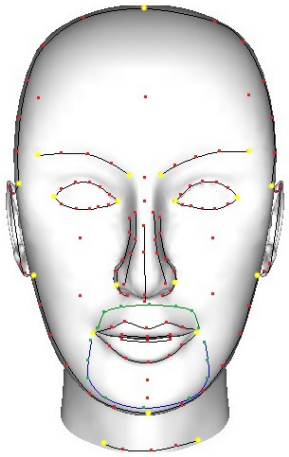

Figure 3.13: Representation of 2D frames of points. Green (upper frame) and blue (lower frame) frames are associated with upper lip and lower lip, respectively.

### 3.6.3 Point Partition based on Region

The next step is to define all the points of generic mesh and the feature points embedded in the masks and set their indexes depending on their corresponding regions (upper lip, lower lip or elsewhere). For such purpose, Cartesian coordinates (x, y, z) of all points are converted to cylindrical coordinates (u, v) using Equation (3.4). Each vertex (mesh points and feature points) is determined to be whether inside one of the aforementioned masks or not relying on Jordan Curve Theorem [79]. This theorem is built on creating horizontal ray (increasing x-value, fixed y-value) out from the candidate vertex and counting the number of edges it crosses. Whenever the ray crosses an edge, it switches between inside and outside. At the end, index values for all points located in the lower lip region, the upper lip region and outside lip regions are set as 0, 1 and 2, respectively. Such indexes are used as conditions to RBF functions.

### 3.6.4 RBF-based Mesh Modification

The lower lip and upper lip should not be affected by each other through deformation of the generic model. Therefore, two different sets (vectors) of feature points are provided for training of RBF networks and also calculating corresponding weight vectors. One set contains all the feature points except the feature points located in the upper lip region with index value set as 0 and the other set contains all the feature points excluding feature points in the upper lip region with index value set as 1. Lastly, the surface points are deformed depending on their region index and corresponding weight vectors.

As a result, the points on the upper lip region will not be affected by the feature points on lower lip region and vice versa. In some cases, the resultant surface might lack smoothness on the boundary between regions after RBFs deformation. This problem would be solved after mesh subdivision and refinement steps. The following subsections describe the technical details involved in RBFs deformation.

#### 3.6.4.1 Interpolation

Initially, the input of deformation RBF networks can be formed as a vector of real numbers $X \in R^3$ and respectively the output of the network would be a scalar function of the input vector, $F: R^3 \rightarrow R$. The number of vertices in 3D space at which the function to be approximated is known is represented by M. These points are the RBF centers (feature points) and named as $X_1$, $X_2,...,X_M$. In our case, M is equal to 128. Moreover, singularity problem arises when the centers having same values are used in interpolation. Smooth interpolation function of F can be approximated in the whole domain of $R^3$. This interpolation function, F, describing the

displacement in the whole domain can be calculated by a sum of basis function $g(r_i)$: $R \rightarrow R$. $r_i$ represents the distance between known vertex position $X_i$ and vertex to be evaluated X and the general expression for each RBF network F(X) is

$$F(X) = \sum_{i=1}^{M} a_i g(|X - X_i|) + c_0 + c_1 x + c_2 y + c_3 z, X = (x, y, z) \quad (3.6)$$

Where $a_i$ is the scalar coefficient. Low order polynomial (first degree polynomial in above formulation) is added as $c_0 + c_1 x + c_2 y + c_3 z$ where $c_0,...,c_3$ are the corresponding coefficients and affine transformation described by these terms cannot be achieved by radial basis functions only. Based on the M values of function $F(x_i, y_i, z_i) = F_i$ , a system consisting of M+4 linear equations can be formulated as GA = F where $F= (F_1, F_2, \ldots, F_M, 0, 0, 0, 0)$ , $A= (a_1, a_2, \ldots, a_M, c_0, c_1, c_2, c_3)$ and G is a (M+4) ×(M+4) matrix:

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & \bullet & \bullet & \bullet & g_{1M} & 1 & x_1 & y_1 & z_1 \\ <br/> g_{21} & g_{22} & \bullet & \bullet & \bullet & g_{2M} & 1 & x_2 & y_2 & z_2 \\ <br/> \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ <br/> \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ <br/> \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ <br/> g_{M1} & g_{M2} & \bullet & \bullet & \bullet & g_{MM} & 1 & x_M & y_M & z_M \\ <br/> 1 & 1 & \bullet & \bullet & \bullet & 1 & 0 & 0 & 0 & 0 \\ <br/> x_1 & x_2 & \bullet & \bullet & \bullet & x_M & 0 & 0 & 0 & 0 \\ <br/> y_1 & y_2 & \bullet & \bullet & \bullet & y_M & 0 & 0 & 0 & 0 \\ <br/> z_1 & z_2 & \bullet & \bullet & \bullet & z_M & 0 & 0 & 0 & 0 \end{bmatrix}$$

Here $g_{ij} = g(\| x_i - x_j \|)$. A good choice of g among many possible options is important for calculation quality and for the interpolants presence. Therefore, Shifted Log Function is used which is formulated as follows:

$$g(t) = \sqrt{\log(t^2 + k^2)}, \; k^2 \geq 1 \quad (3.7)$$

Where K is the variable which controls displacement magnitude and t is a parameter used to modify the displacement with respect to time. The coefficients required in Equation (3.6) can be determined by solving the equation for A values.

### 3.6.4.2 3D Model Deformation

The deformation for M 3D points (Centers) $X_i$ are known and formed as a vector representing 3D displacement ($U_i$) of the centers located on the un-deformed surface. Therefore, the positions of centers are moved to the new positions $X_i + U_i$. Interpolating of these displacements to other surface points is approximated by RBF interpolation technique. Having $X_i$ and $U_i$, three linear equations can be formulated as below with the displacements u as a coefficient described earlier.

$$GA_x = ( u_1^x, u_2^x, \dots, u_M^x, 0,0,0,0)^T \quad (3.8)$$

$$GA_y = ( u_1^y, u_2^y, \dots, u_M^y, 0,0,0,0)^T$$

$$GA_z = ( u_1^z, u_2^z, \dots, u_M^z, 0,0,0,0)^T$$

Calculating the values for $A_x$, $A_y$ and $A_z$ requires a single matrix inversion and three matrix-vector multiplications which results in the coefficients for interpolation displacements required for Equation (3.6). Figure 3.14 shows adapted generic model to an input face using RBFs.

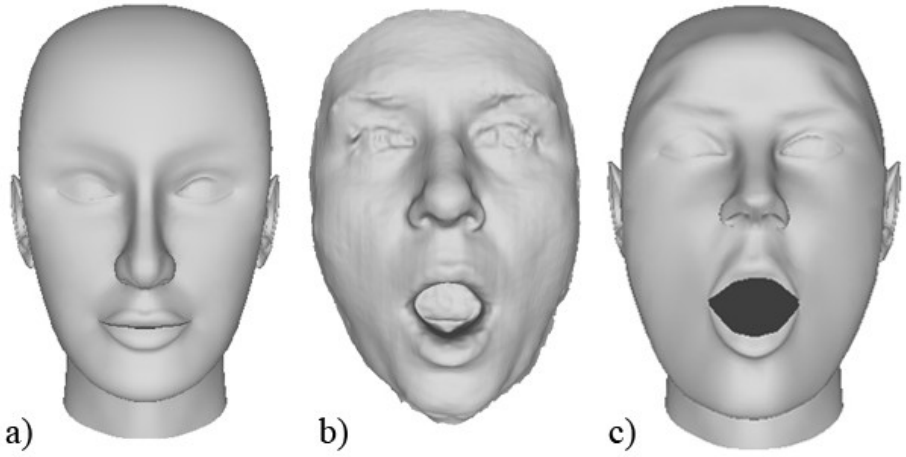


Figure 3.14: Feature-based adaptation using RBFs deformation. a) Generic model, b) input model, c) deformed generic model.

## 3.7 Mesh Refinement

Feature-based approach uses only a limited set of characteristic information from input face. Therefore, mesh model roughly approximates the input face after this approach. The following subsections explain the process how to refine a mesh after feature-based adaptation by adding existing surface details of the corresponding input model.

### 3.7.1 Resolution Increase using Subdivision

In order to increase the resolution of mesh model after RBFs deformation, Loop subdivision approach [47] is utilized which divides each triangle of the mesh into four sub-triangles by splitting each edge by half and linking new generated vertices. This approach can considerably raise the number of mesh triangles and consequently obtain smoother surface of the mesh model. In our case, the total number of triangles almost increases five times more than the number of original mesh triangles, up to 32425 (Figure 3.15).

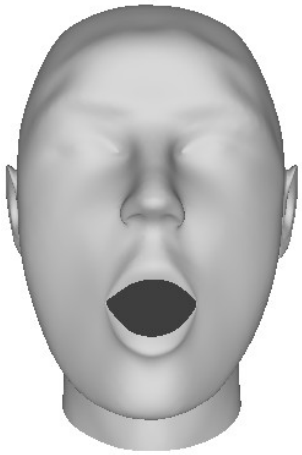

Figure 3.15: Subdivided model after feature-based adaptation.

Despite resolution enhancement, the mesh still doesn't closely resemble the input face. Therefore, mesh refinement approach relying on surface points is carried out to deliver highly accurate surface adaption between subdivided model after feature-based adaptation and input face.

### 3.7.2 Surface Point-based Adaptation using Cylindrical Projection

The surface point-based adaptation method ensures that a perfect match is attained between subdivided model and input face. The main idea is to project each vertex of the subdivided model after feature-based adaptation along the ray passing through the vertex onto the surface of input face. The intersection point on the surface of input model is considered as corresponding 3D position of the vertex on the input model. To calculate the intersection point, first a cylindrical projection technique is employed which encloses the head models with a virtual cylinder as shown in Figure 3.16. The cylinder centerline is located at centroid of the head models (centered at world origin) in the same direction as Y-axis. Intersections on the surface of the input model occur between the rays casted from the cylinder centerline when the rays are parallel to each other which rarely happens. As a result, this projection produces best outcome in the presence of surface complexity and irregularity. Every 3D point of the subdivided model ($x_i$, $y_i$, $z_i$) and input model ($x_i'$, $y_i'$, $z_i'$) is transformed into 2D image plane ($u_i$, $v_i$) and ($u_i'$, $v'_i$) respectively using Equation (3.4). Then, every point of subdivided model ($u_i$, $v_i$) is assumed to be located inside a triangle (($u'_1$, $v'_1$), ($u'_2$, $v'_2$), ($u'_3$, $v'_3$)) on the input model. Hence, barycentric coordinates of the intersection point ($u_i$, $v_i$) are calculated with respect to the vertices of the triangles on the input model using Equation (3.5). As mentioned before, the intersection point lies inside a triangle if it satisfies the conditions defined by the barycentric coordinate systems

((α, β >= 0) and (α + β <= 1)). Based on the 3D values of triangle vertices and computed triangle barycentric coordinates, we can calculate 3D point $X'_i$ on the surface of input model corresponding to $(u_i, v_i)$. Finally, the distance between the calculated $X'_i$ and corresponding 3D point on the surface of generic model $X_i$ is computed. If the calculated distance is below the threshold value defined in the system, the point is shifted toward surface of input model otherwise the point remains unchanged. Since the input models are missing parts such as ears and back of the head, the points of such parts on the surface of subdivided model won't be changed. Figure 3.16 shows the step involved in the aforementioned technique. A sample result of cylindrical projection-based method is displayed in Figure 3.17.

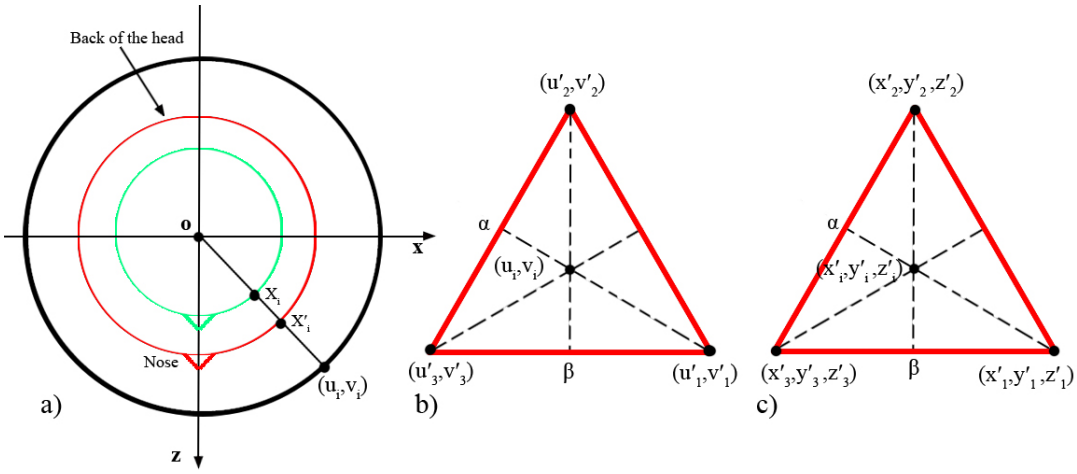


Figure 3.16 : Surface point-based adaptation of subdivided model to the input face using cylindrical projection-based method. a) Transformed point to cylindrical coordinates, b) detected intersecting triangle using barycentric coordinates, c) computed 3D point on the input face.

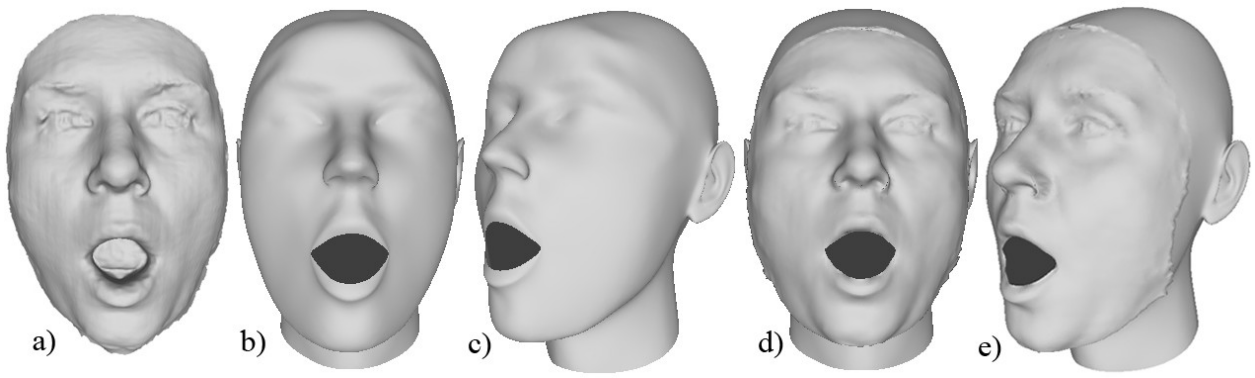


Figure 3.17: Surface point-based adaptation process. a) Input face, b-c) subdivided model from two views, d-e) refined model after cylindrical projection from two views.

As it is shown in Figure 3.17, the facial region of the head model (refined region) after surface point-based adaptation is properly adapted to the input face. However, the rest of the head remains unchanged (un-refined region) since some parts are missing in the input model. As a result, disconnectivity (smoothness related) issue is raised between refined and un-refined regions. On the other hand, un-refined region might have slightly different volume from refined region.

## 3.8 Post Mesh Refinement for Smoothing

This section devotes to solve the problems as shown in Figure 3.17 (e). We introduce two post refinement approaches based on RBFs deformation and averaging neighboring points utilized to generate a natural looking and smooth surface through entire the mesh. These techniques refine the mesh without affecting the sharp features in facial region. As a result, the head model would be visually acceptable enough to be used by practical applications.

### 3.8.1 Smoothing using RBFs

The primary method leveraged for refinement of the surface smoothness and naturalness is the RBF-based deformation method described in details in section 3.6.4. The main idea is to displace the un-refined points based on the displacement of refined points from their correspondences in the subdivided mesh. Initially, all refined points which typically represent facial region are determined. Then, neighboring points of each refined point are evaluated to determine whether those belong to un-refined region or not. The refined points with un-refined points in the neighborhood are considered as border points. Afterward, the correspondences of border points in subdivided mesh are determined. Border points and their correspondences are used as two input point sets for training of RBF networks and calculating weight vectors. All the surface points except the refined points are passed to RBF functions and then their new values are calculated. As a result, smoother surface is approximated. This approach is immediately followed by another approach utilized to improve surface smoothness around the border points.

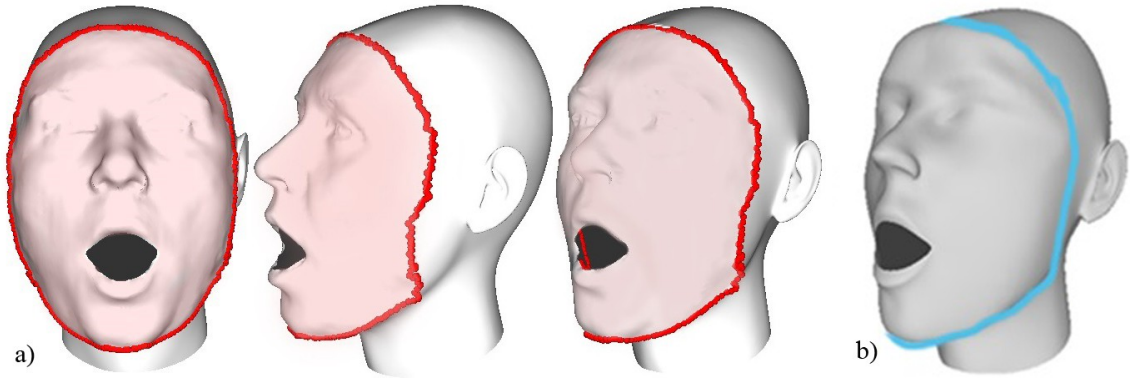


Figure 3.18: Demonstration of a) the refined region (shown in pink) and the border points (shown in red) from three different views and b) the correspondences of the border points (shown in blue).

### 3.8.2 Smoothing using Average Points Calculation

In order to make the mesh surface smoother in the neighborhood of intersection between two regions (refined and un-refined), an iterative smoothing method is used. The process starts with setting the border points as initial input vector and determining all the neighboring points (Figure 3.19) for each border point. Neighboring points which are also border points are excluded. Then neighboring points are set as next iteration input vector. The average value of the neighbor points for each border point is computed and set as the new value of that border point. This process is repeated for 2 iterations, so a relatively smooth distribution of points around the border is achieved. The output after refinement methods is a natural looking head mesh of specific individual as illustrated in Figure 3.20 from three different views.

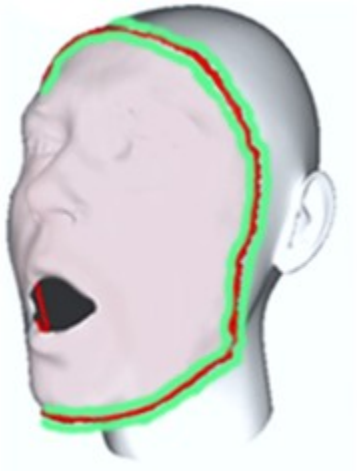

Figure 3.19: An illustration of the border points (shown in red) and their corresponding neighboring points (shown in green).

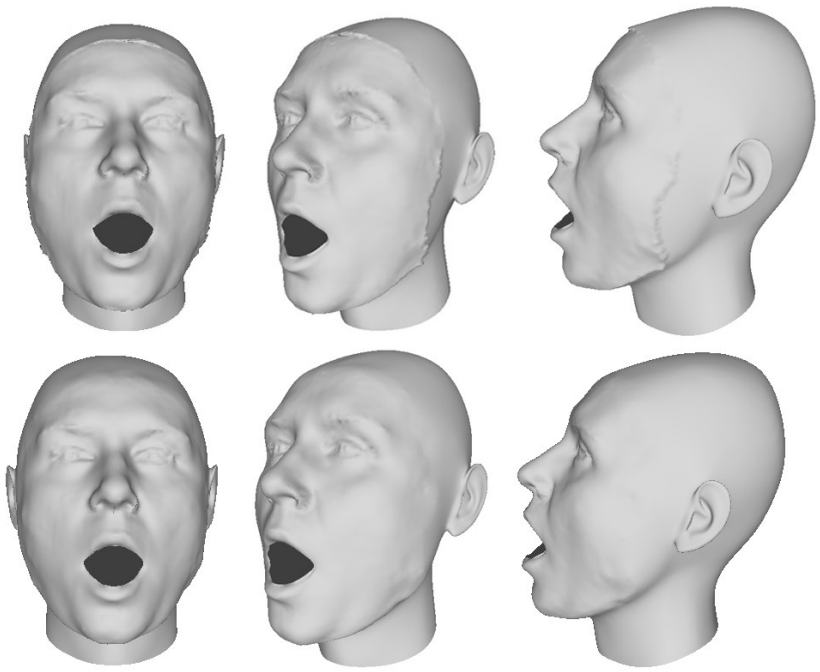

Figure 3.20: Top row demonstrates resultant mesh after surface point-based adaptation and second row shows natural looking and smoothed mesh after two post refinement approaches based on RBFs deformation and averaging neighboring points from three different views.

## 3.9 Mesh Defect Repairing

After surface point-based adaptation is accomplished, the refined model typically suffers from defects such as self-intersections and degeneracies which are caused by wrong projection of the points which create unpleasant visual artifacts on the model. These problems often occur around the nose alae as displayed in Figure 3.21 and chin-neck area if it is included in input model. Since such defects lower the quality of mesh, they should properly be handled in a way that fine details distributed away from defects on the mesh surface can be preserved.

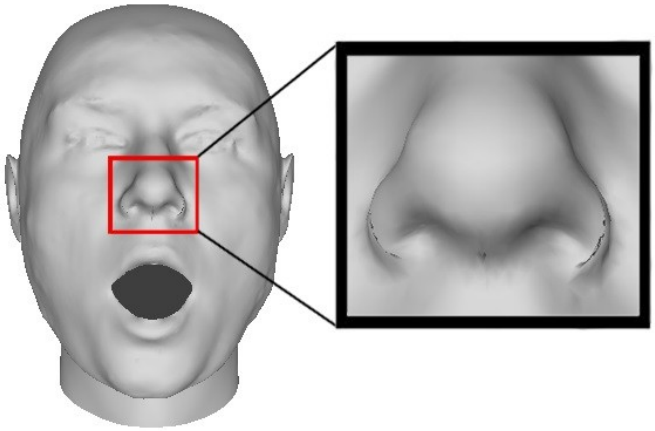

Figure 3.21: Created defects such as self-intersections around nose alae.

Therefore, it is desired to locally repair the mesh in the neighborhood of flaws. According to the related works described in the literature which are suitable for locally handling specific types of defects, we decided to implement the work introduced by Attene [69] to repair the mesh and consequently improve the mesh quality.

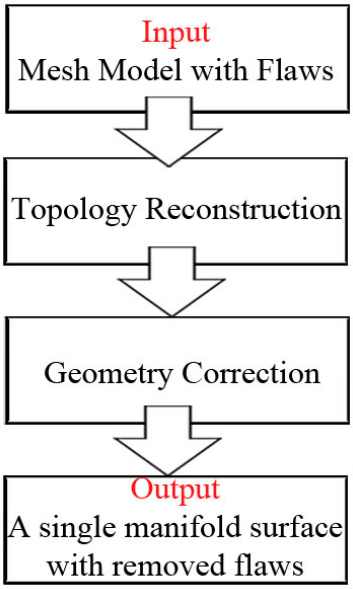


Figure 3.22: The flow diagram of mesh model repairing using the technique proposed by Attene [69].

Figure 3.22 shows the high-level view of the works for mesh model repairing involved by the preceding technique. Since such technique makes the mesh as a single manifold model, it is required to perform some post-processing operations so that undesired parts can be removed around the mouth if created. As a result, the desired shape of the mesh can be recovered. According to the methods explained in section 3.6.1, upper-opened and lower-opened vertices can be extracted around the lips of the mesh obtained after refinement approaches. After finding vertices along the contour of lips, all points (mesh and contour points) are converted to 2D image coordinates. Then, all the mesh points enclosed by the contour points are determined using same approach as the one used in section 3.6.3. These points are considered as undesired vertices which have been added through the repairing method. Finally, such points with their associated faces are removed from the mesh and the system output is obtained as a smooth, clean head mesh.

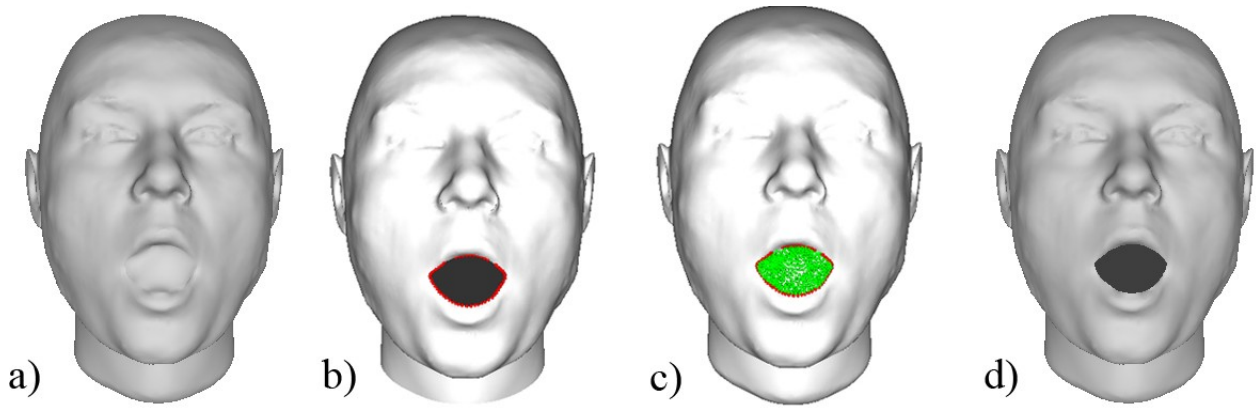


Figure 3.23: Head model after a) repairing method, b) post refinement method with contour points shown in red, c) repairing method with extra points (shown in green) surrounded by contour points (shown in red), d) undesired points removal.

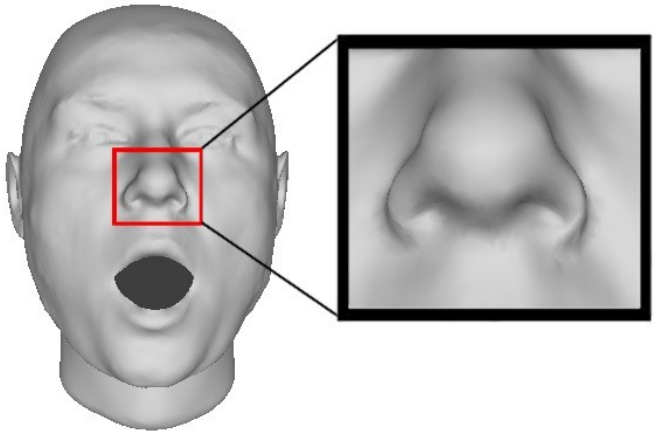

Figure 3.24: Fixed flaws such as self-intersections around nose alae.

# Chapter 4. Result and Validation

In this section, we represent experimental results, analysis and validation of the meshes carried out in order to demonstrate the efficiency and robustness of the proposed framework. Finally, evaluation of the leveraged methods including their corresponding benefits compared with other techniques, general issues and limitations is also covered in this chapter.

## 4.1 Experimental Results

In this section, experimental results of two persons (A and B) are demonstrated. The data of person A with three different expressions (I, II and III as neutral, shouting and surprise, respectively) is collected from the publically available database, Bosphorus database [73], in the form of point clouds. On contrary, the data of person B is in the form of a triangulated mesh where we have performed scanning task with the Cyberware laser scanner [72]. The data captured from person B is reconstructed with only neutral facial expression (I) while the eyes are closed. By the system, we have tested all the input models that don't have back of the head and ears and have eyes and mouth which are either close or open.

The following figures illustrate the sequential steps involved in our methodology as follows:

- Reconstructing the surface of person A (point cloud) with different performed expressions I, II and III and person B (mesh model) with performed expression I.
- Bringing input point face into space of generic mesh.

- Adapting the generic model to the input face using feature point sets and a modified RBF-based deformation which leverages regional information.
- Subdividing mesh model after feature point-based step in order to increase the mesh resolution and transferring surface details from the input face so that a fine match is obtained between the mesh and the input face.
- Enhancing head model quality further by utilizing post refinement methods for entire head smoothing followed by an automatic repairing method for removing mesh defects such as self-intersections.

### 4.1.1 Person A with Expression I

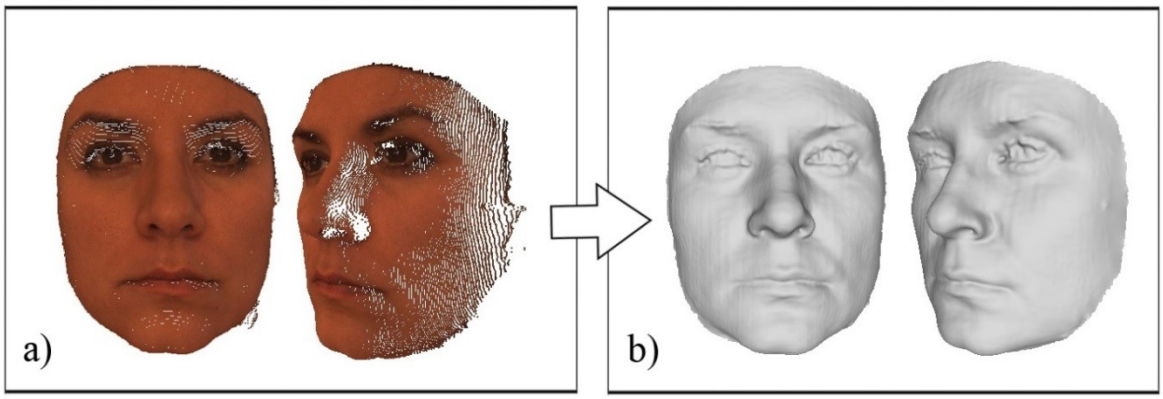


Figure 4.1: Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views.

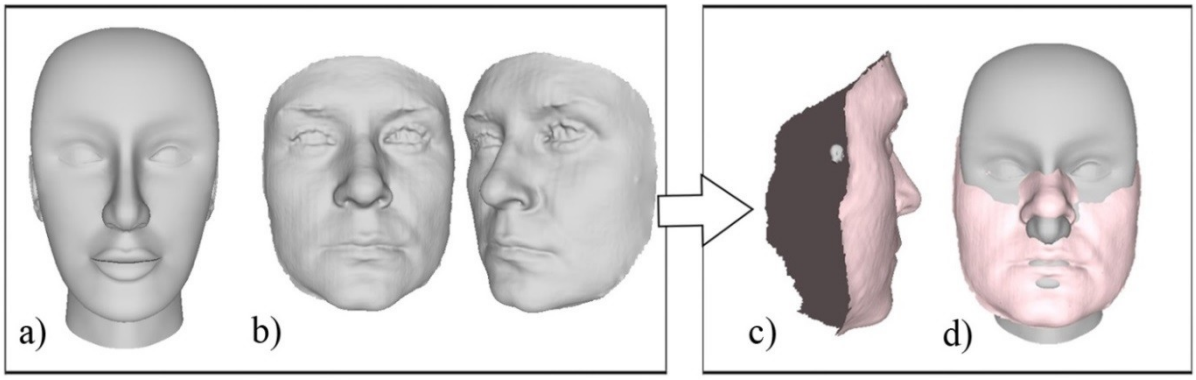


Figure 4.2: Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model.

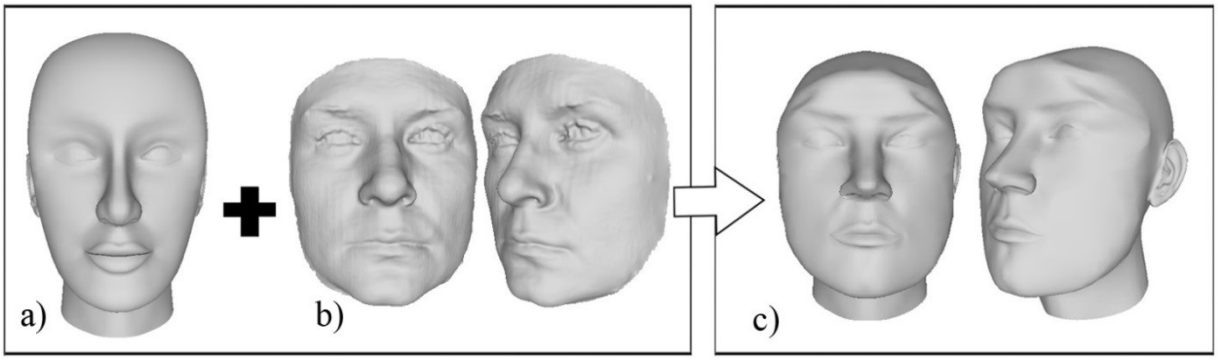


Figure 4.3: Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views.

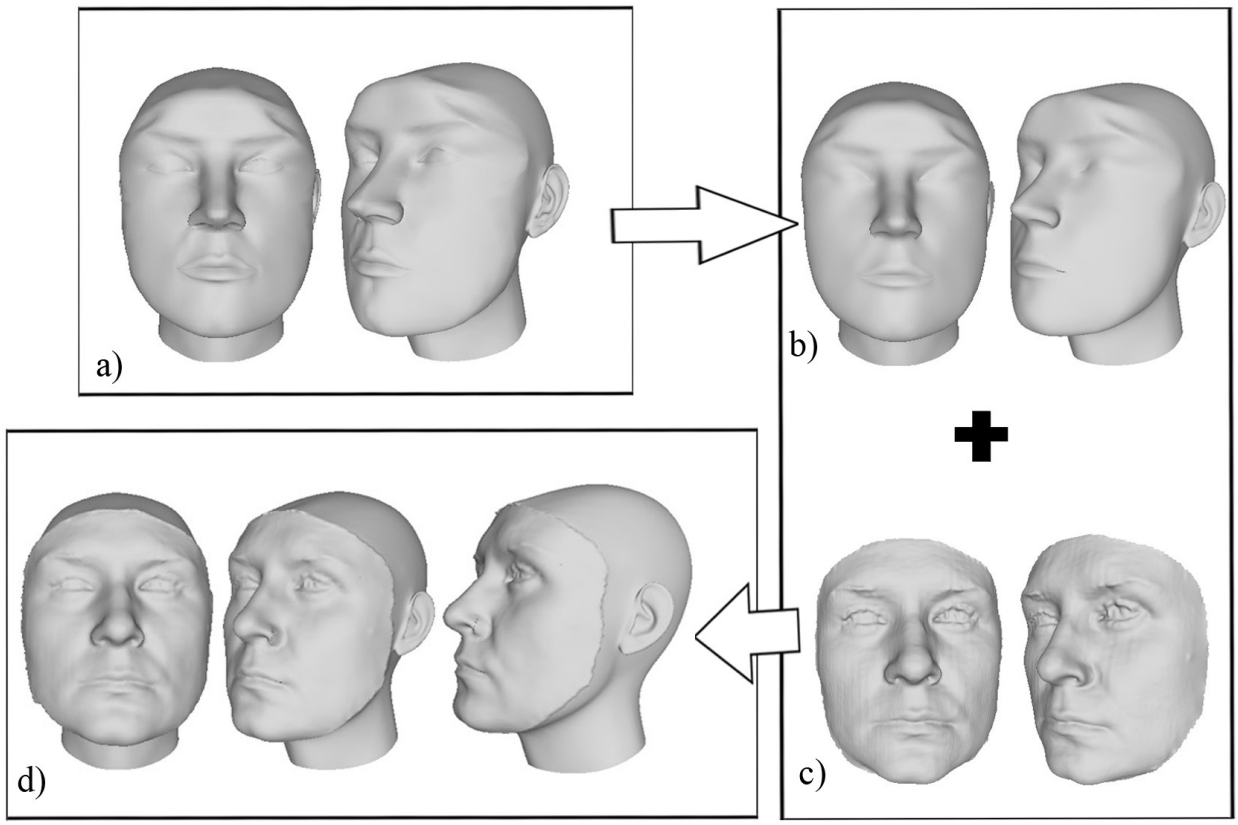


Figure 4.4: Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from three views.

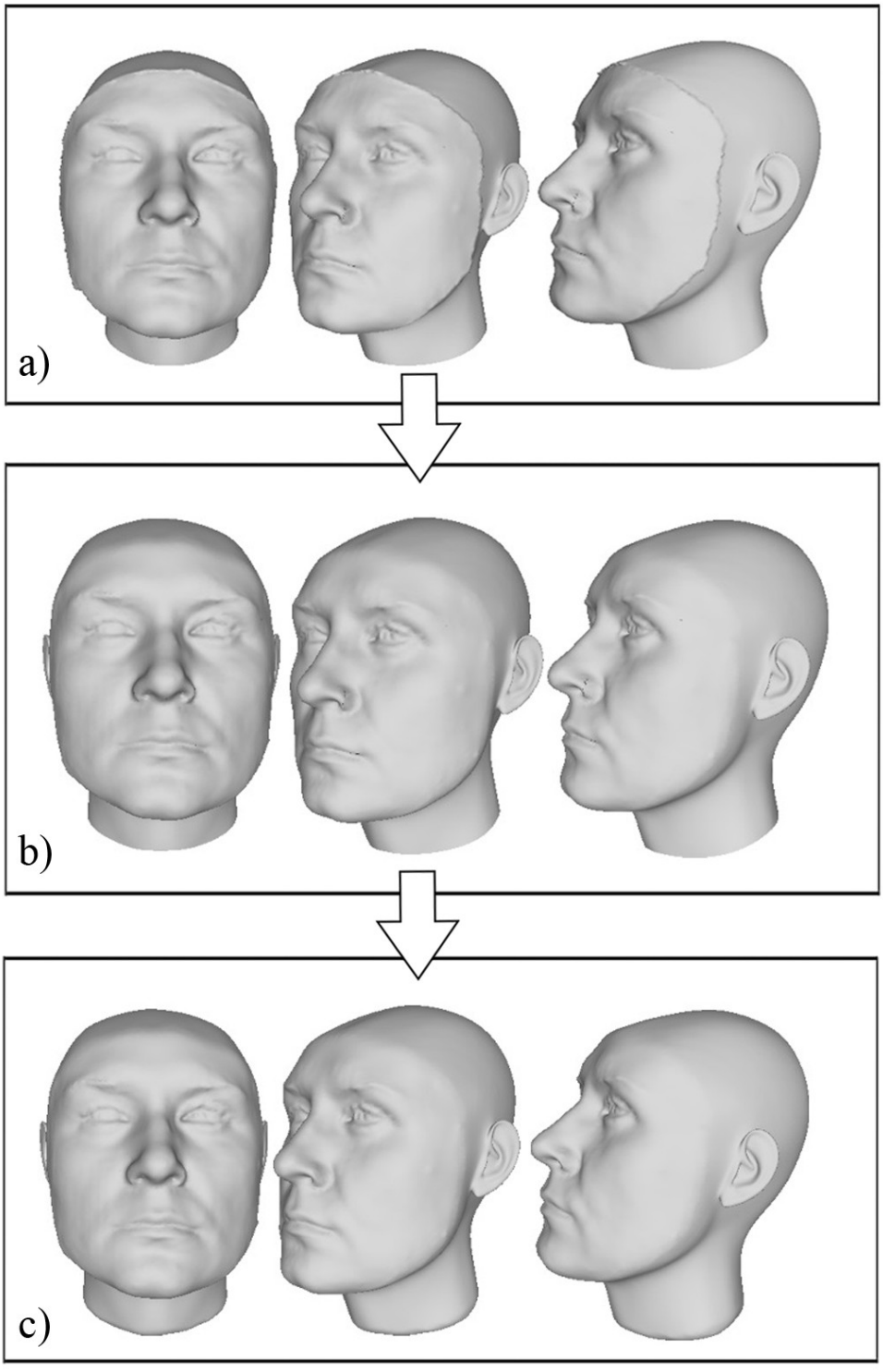


Figure 4.5: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method.

### 4.1.2 Person A with Expression II

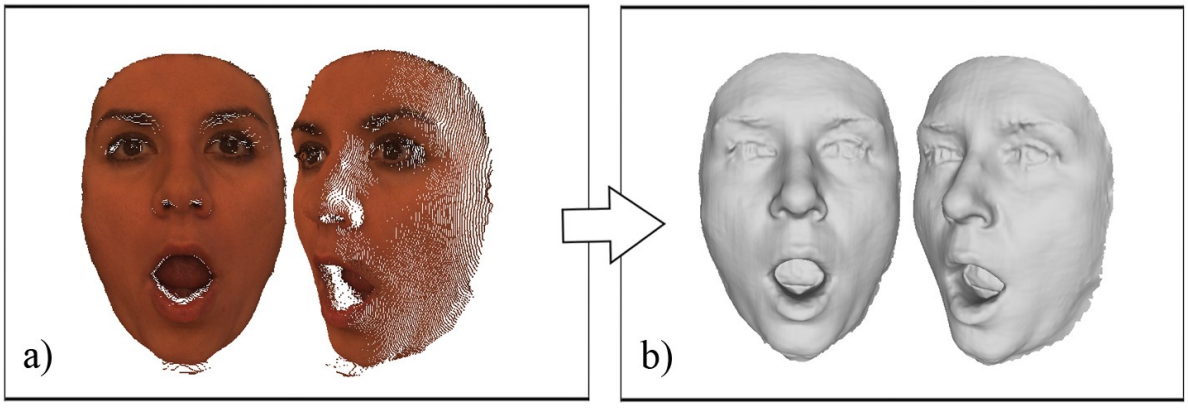


Figure 4.6 : Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views.

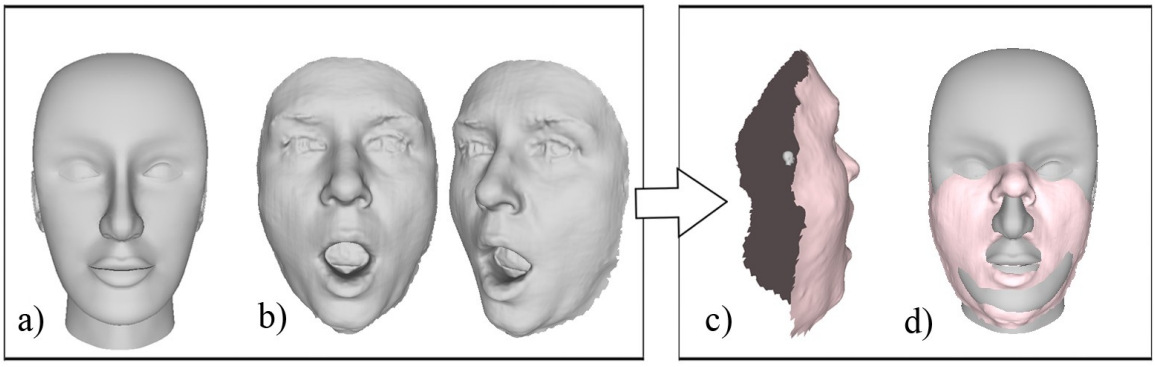


Figure 4.7 : Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model.

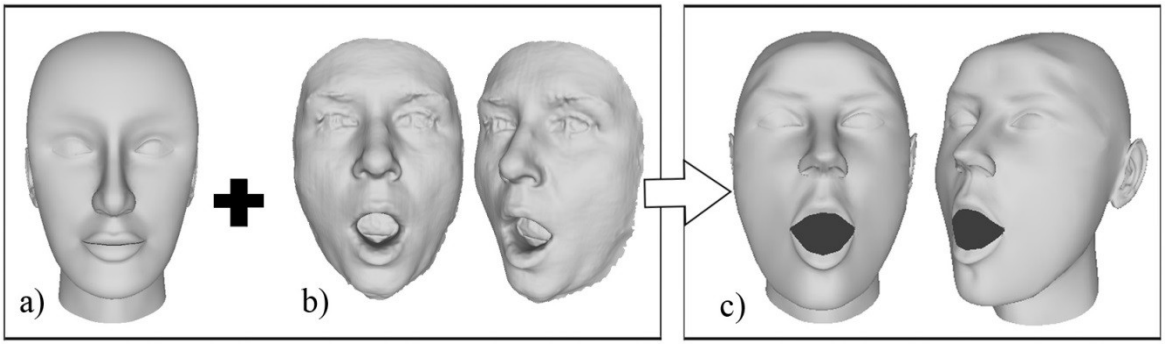


Figure 4.8 : Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views.

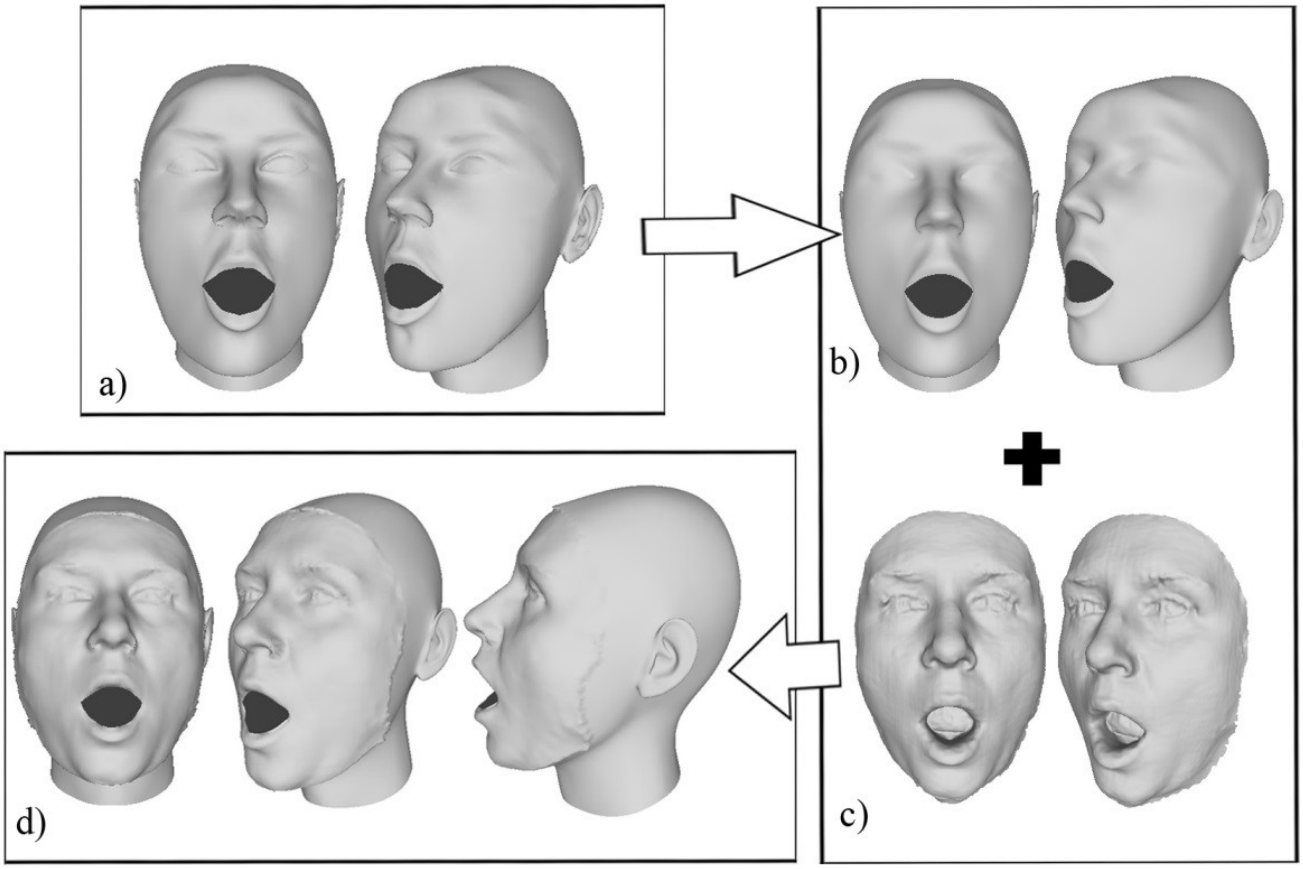


Figure 4.9 : Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from three views.

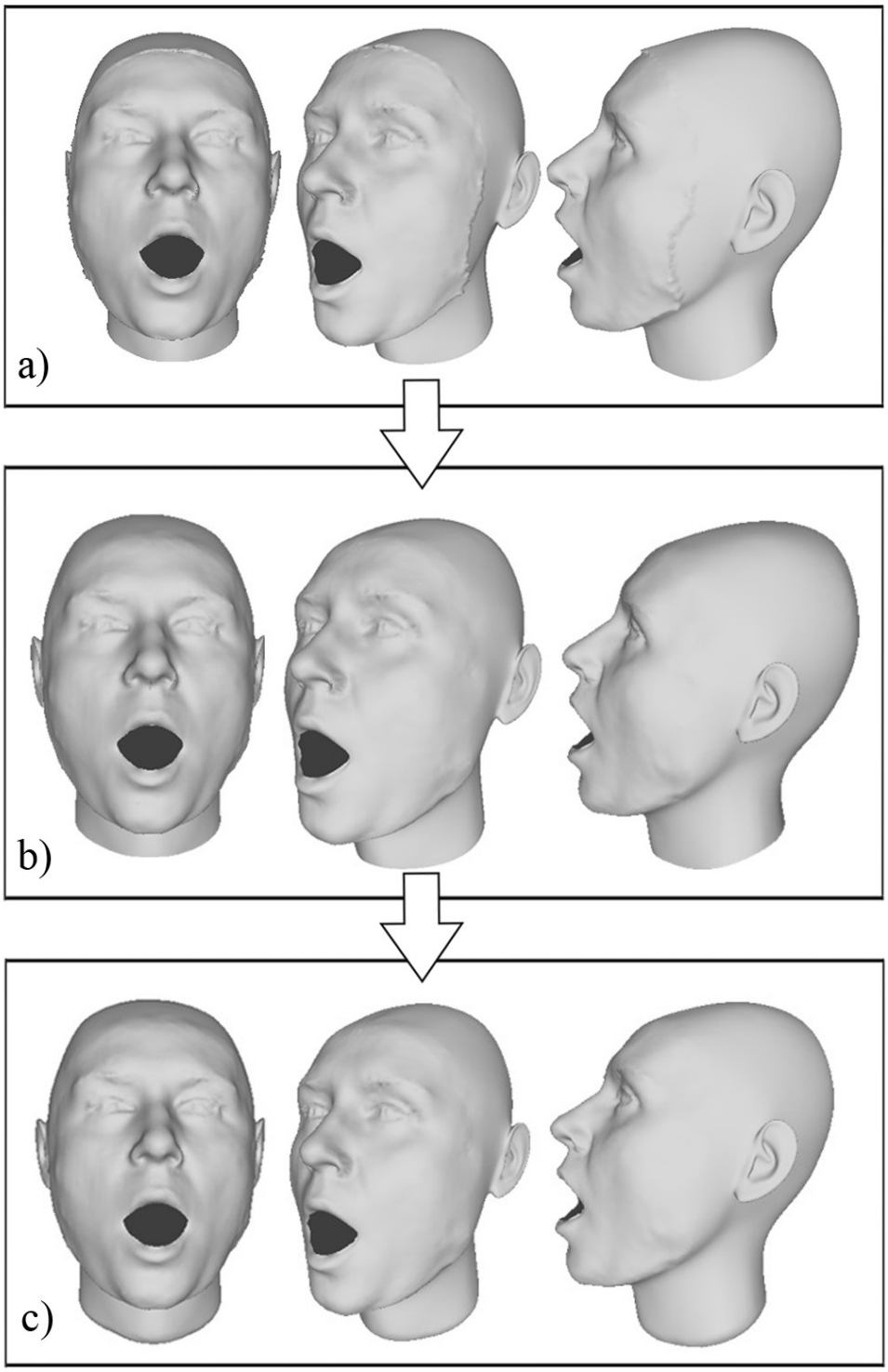


Figure 4.10: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method.

### 4.1.3 Person A with Expression III

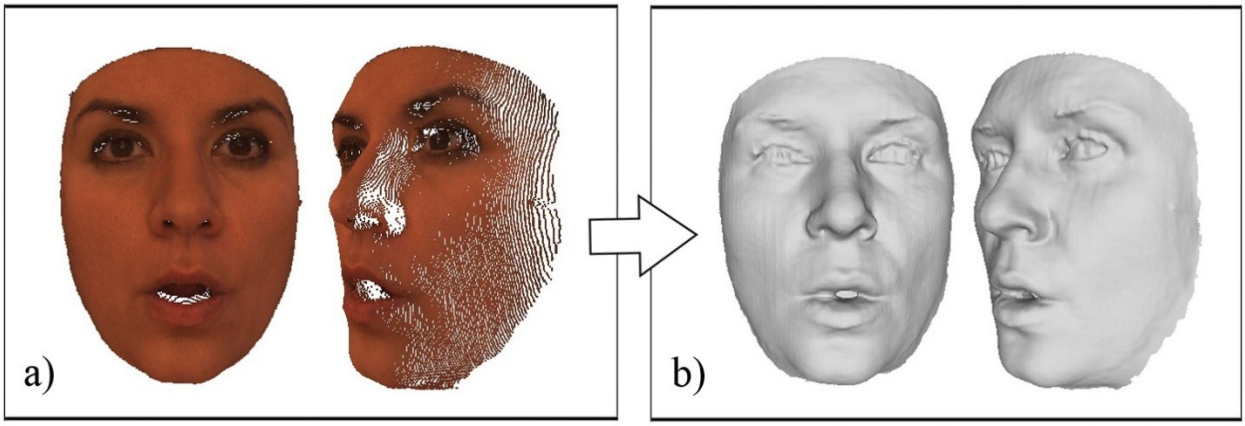


Figure 4.11 : Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views.

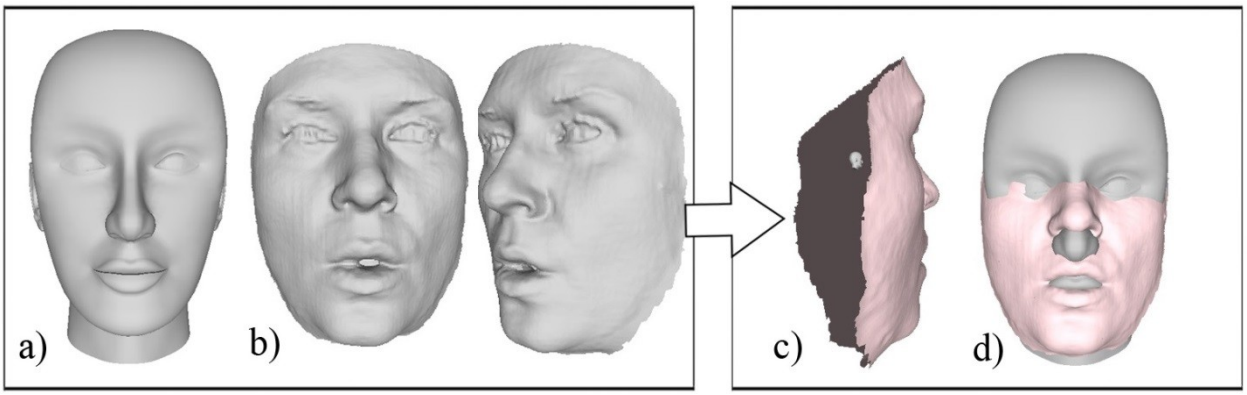


Figure 4.12: Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model.

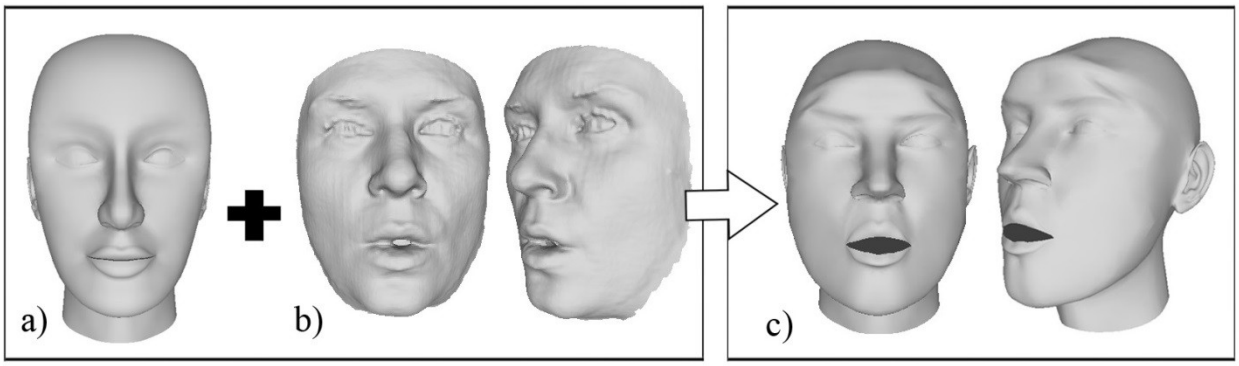


Figure 4.13: Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views.

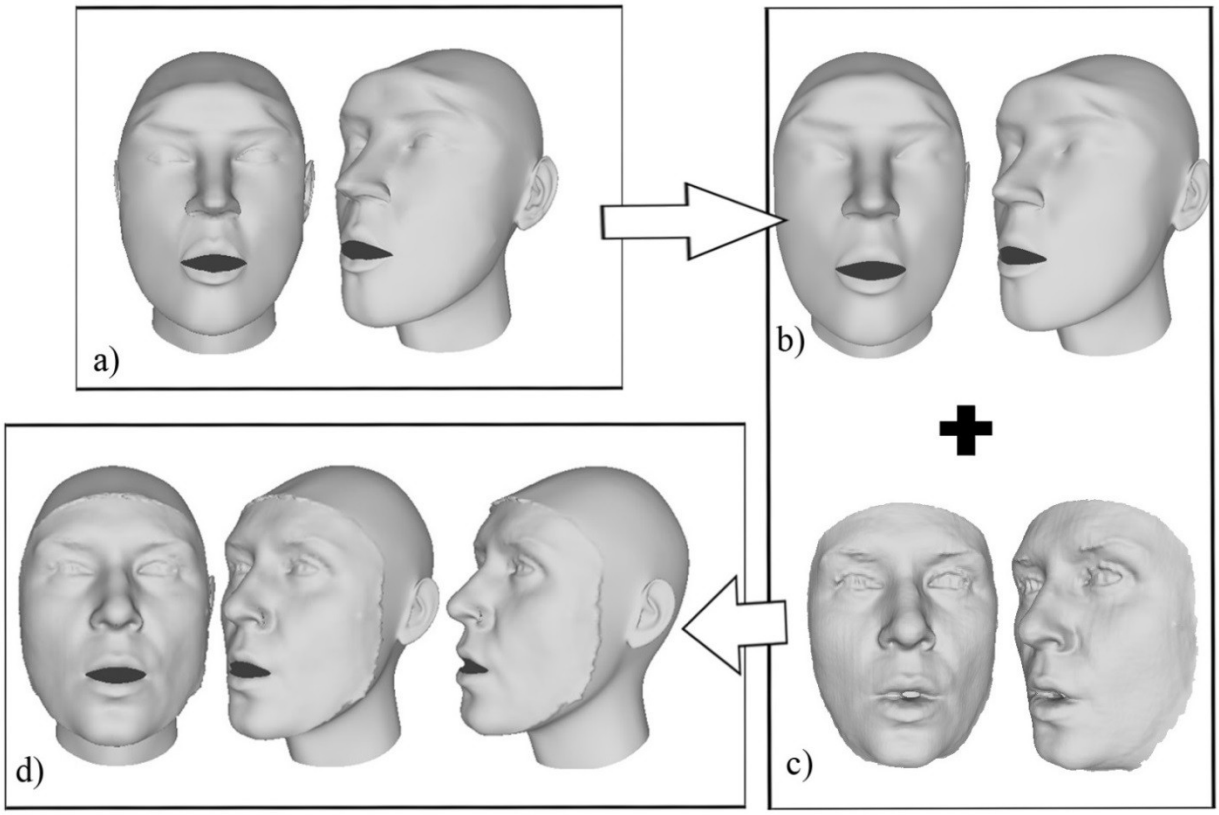


Figure 4.14: Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from three views.

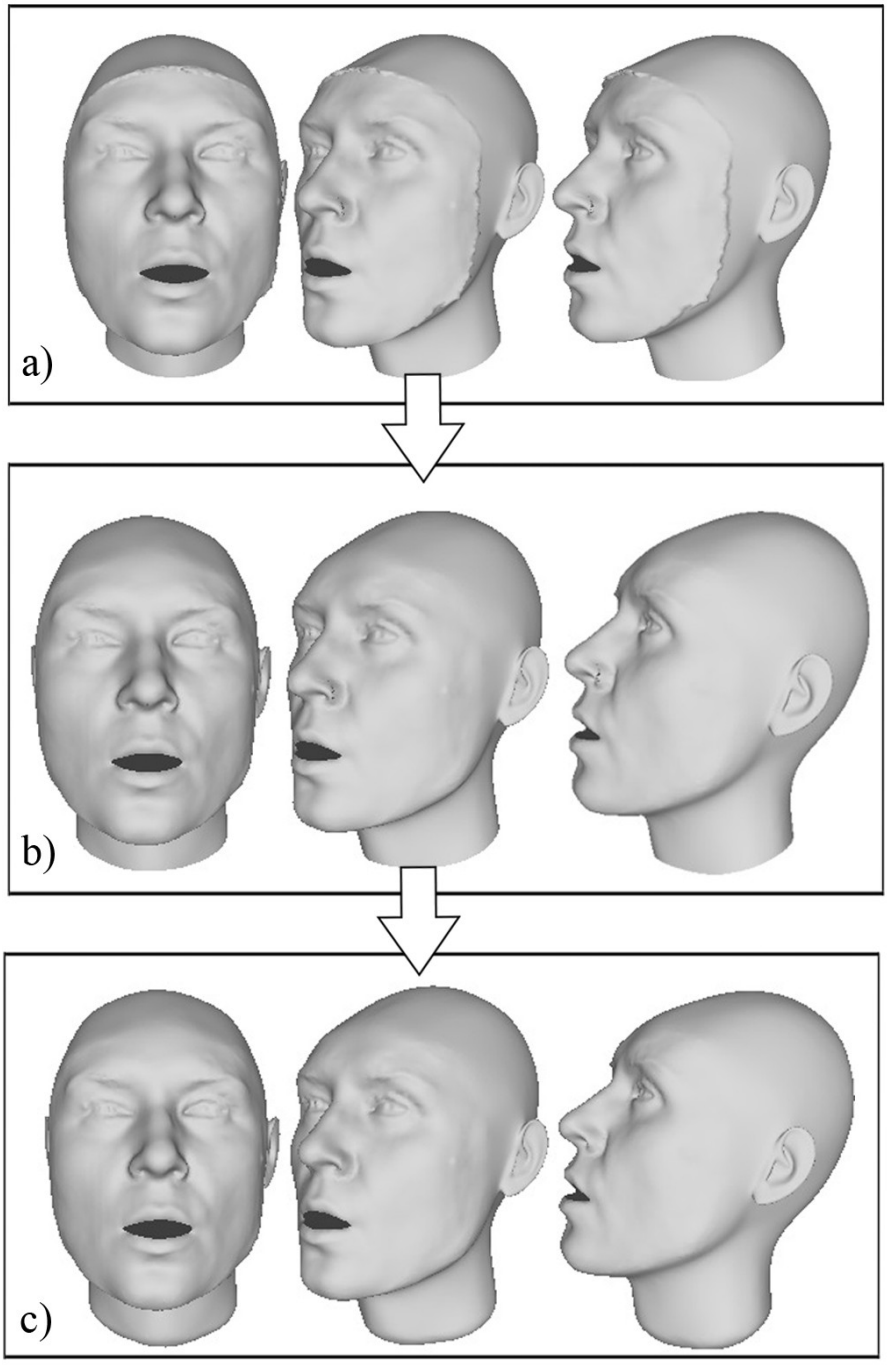


Figure 4.15: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method.

### 4.1.4 Person B with Expression I and Closed Eyes

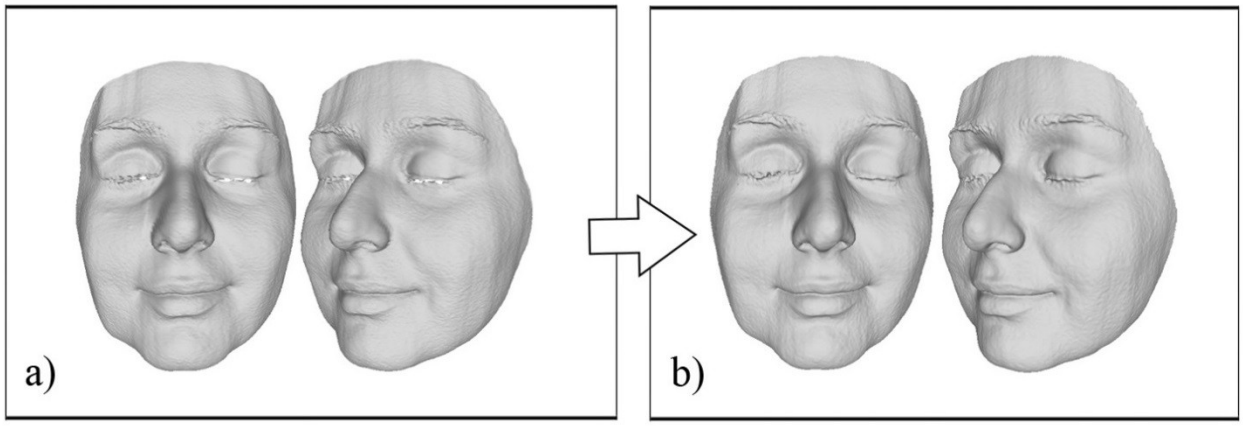


Figure 4.16: Surface reconstruction process. a) Input point cloud from two views, b) reconstructed surface from two views.

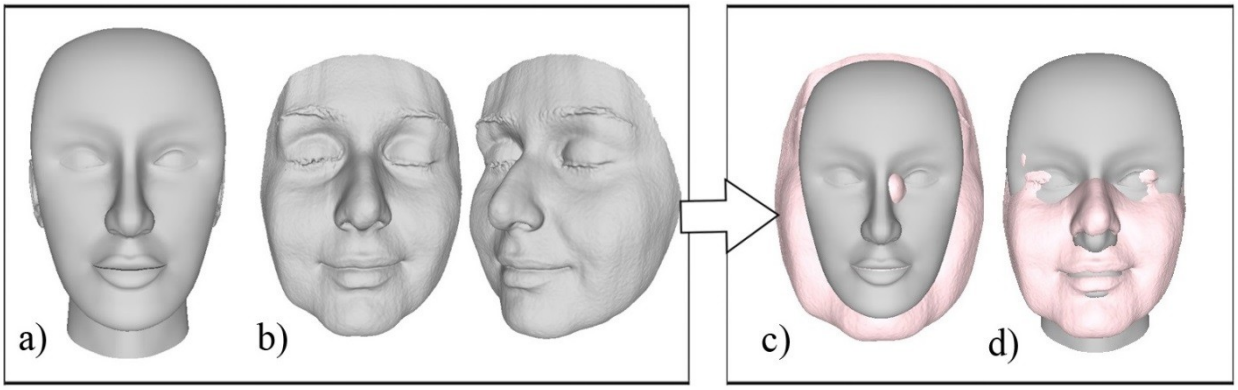


Figure 4.17: Mesh registration process. a) Generic model, b) input face from two views, c) models before registration, d) registered input face (shown in pink) to generic model.

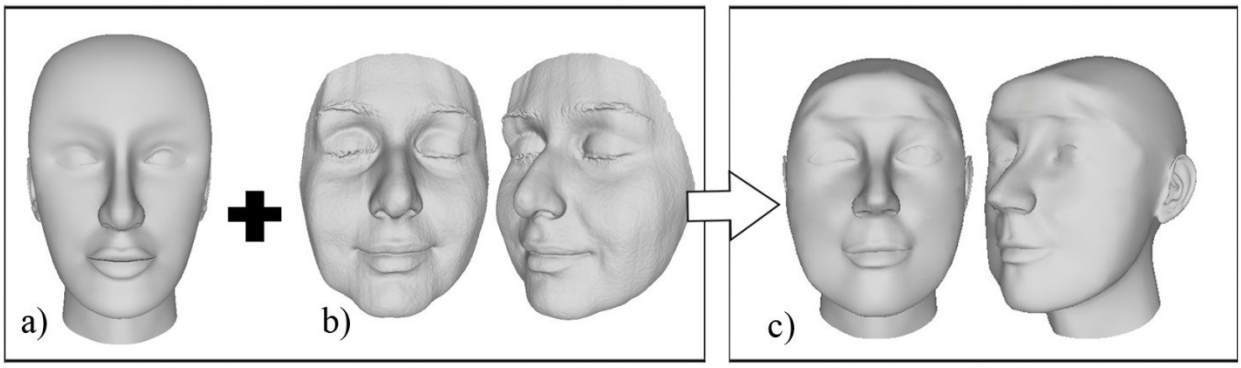


Figure 4.18 : Feature point-based adaption process. a) Generic model, b) input face from two views, c) deformed generic model from two views.

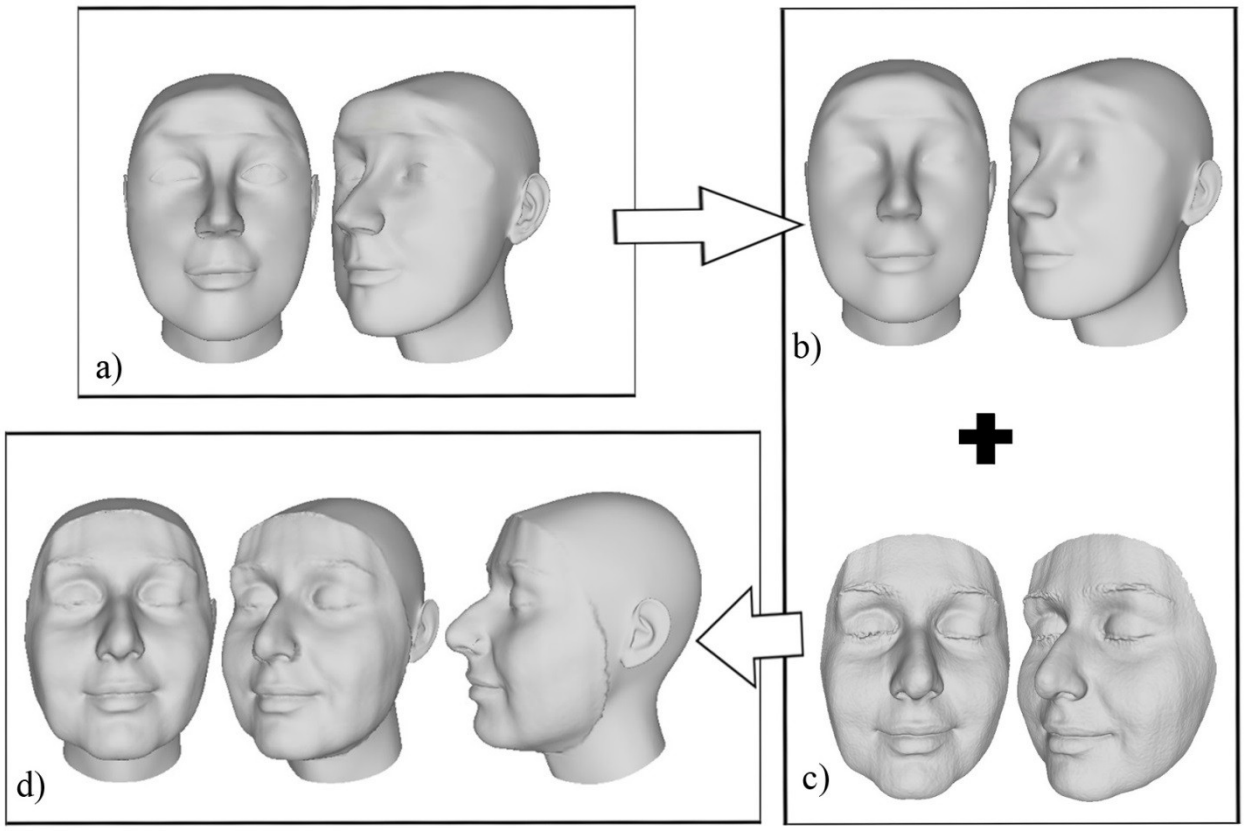


Figure 4.19: Mesh refinement process. a) Feature point-based model from two views, b) subdivided model from two views, c) input face from two views, d) surface point-based adapted model from two views.

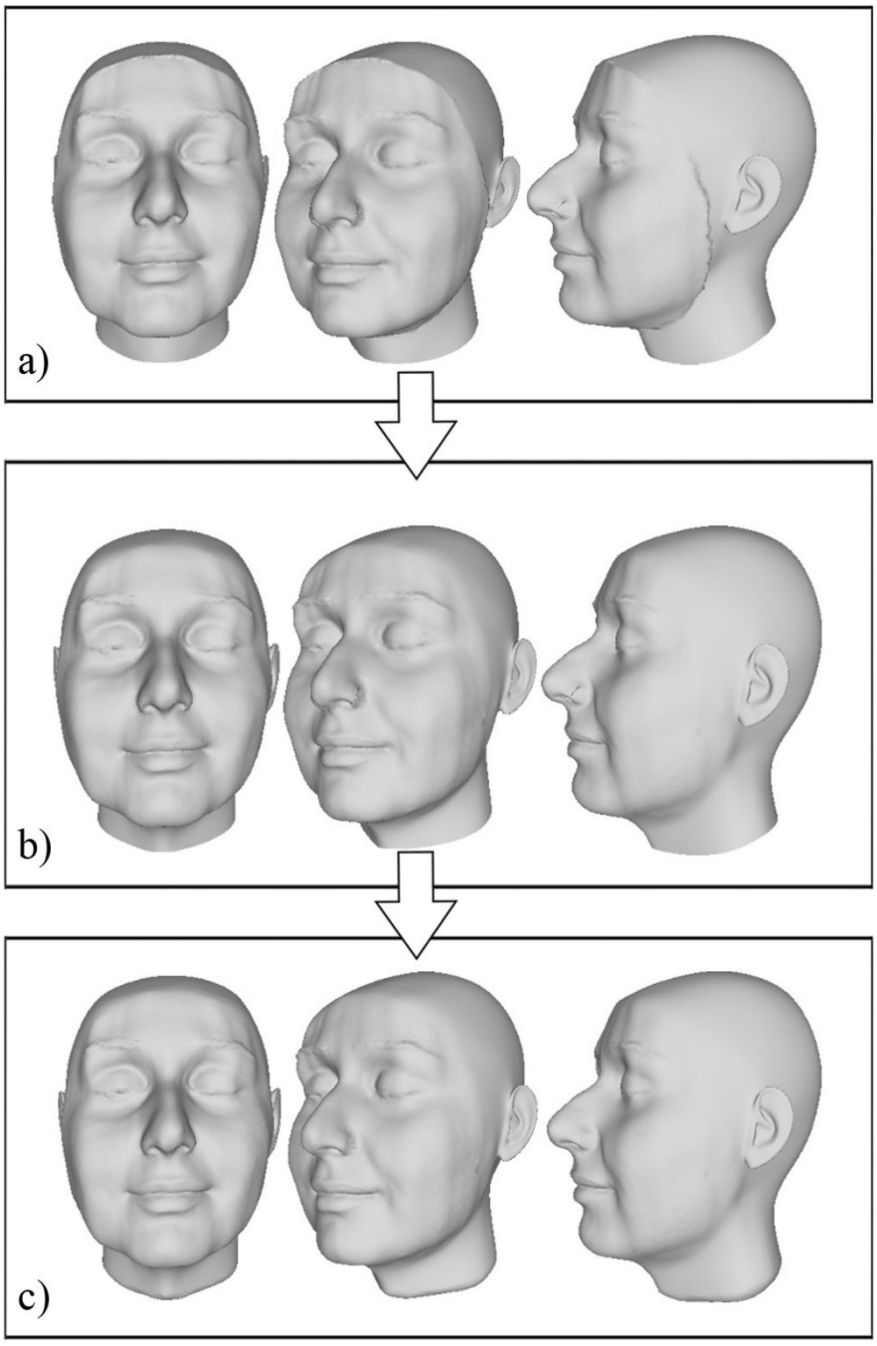


Figure 4.20: Representation of models from three different views after a) surface point-based adaptation method, b) post refinement method, c) mesh repairing method.

## 4.2 Analysis of Meshes, System Performance and Validation

All the system experiments are performed on one modern desktop computer (Intel i7 processor with 4 cores running 2.67 GHz, 6GB of DRR3 RAM running at Windows 7 64bit Operating System). The system is implemented and developed by using OpenGL® (Open Graphics Library), Microsoft Visual C++ 2010 and Matlab.

Despite proving robustness of the system by obtaining good visual results from several test data, we provide additional information for analysis of mesh qualities and system performance and consequently validate both system efficiency and robustness. Such information which is included in the following tables and figure represents the average computation time required for each process, the effect of each process on vertices/faces number and the calculated distance between input and output surfaces (as reconstruction error). In order to compute distance between two surfaces, we utilize Hausdorff distance [80] that samples one of the two surfaces and then finds for each sample the closest point over the other surface (Figure 4.21).

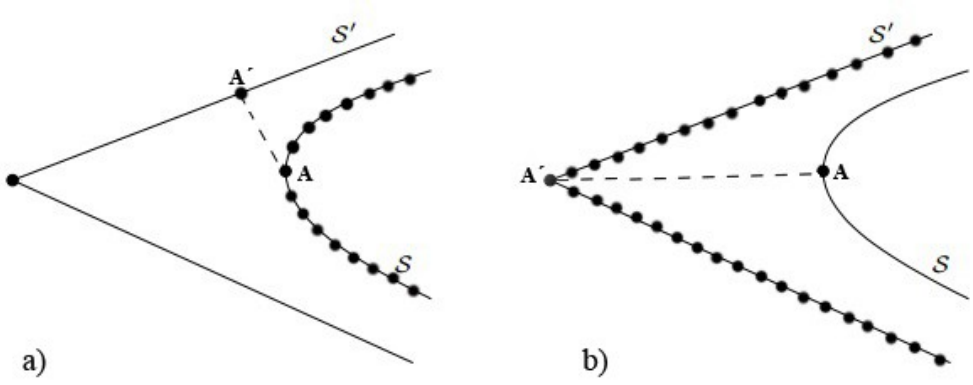


Figure 4.21: a) Having the sampled surface S, the closest point to the point A is the point A' on the surface S', b) having the sampled surface S', the closest point to the point A' is the point A on the surface S.

Table 4.1: Performance analysis for person A with expression I.

| Person A with Expression I | | |
|---|---|---|
| **Process** | **Number of Vertices / Faces** | |
| | **Generic Model** | **Input Face** |
| Surface Reconstruction | 6420 /12600 | 82913 / 165147 |
| Registration | - | - |
| Modified RBF-based Deformation | - | - |
| Subdivision | 32425 / 64670 | - |
| Cylindrical Projection-based Adaptation | - | - |
| Post Mesh Refinement | - | - |
| Mesh Repairing | 34820 / 69636 | - |

Table 4.2: Performance analysis for person A with expression II.

| Person A with Expression II | | |
|---|---|---|
| **Process** | **Number of Vertices / Faces** | |
| | **Generic Model** | **Input Face** |
| Surface Reconstruction | 6420/12600 | 95713 /190800 |
| Registration | - | - |
| Modified RBF-based Deformation | - | - |
| Subdivision | 32425/ 64670 | - |
| Cylindrical Projection-based Adaptation | - | - |
| Post Mesh Refinement | - | - |
| Mesh Repairing | 33880/67698 | - |

Table 4.3: Performance analysis for person A with expression III.

| Person A with Expression III | | |
|---|---|---|
| **Process** | **Number of Vertices / Faces** | |
| | **Generic Model** | **Input Face** |
| Surface Reconstruction | 6420 /12600 | 81677 /162863 |
| Registration | - | - |
| Modified RBF-based Deformation | - | - |
| Subdivision | 32425 / 64670 | - |
| Cylindrical Projection-based Adaptation | - | - |
| Post Mesh Refinement | - | - |
| Mesh Repairing | 34220 / 68387 | - |

Table 4.4: Performance analysis for person B with expression I and closed eyes

| Person B with Expression I and Closed Eyes | | |
|---|---|---|
| **Process** | **Number of Vertices / Faces** | |
| | **Generic Model** | **Input Face** |
| Surface Reconstruction | 6420 /12600 | 107710 / 213790 |
| Registration | - | - |
| Modified RBF-based Deformation | - | - |
| Subdivision | 32425 / 64670 | - |
| Cylindrical Projection-based Adaptation | - | - |
| Post Mesh Refinement | - | - |
| Mesh Repairing | 34219 / 68434 | - |

In the following tables, the mean values are the average errors calculated over the entire sampled surfaces.

Table 4.5: Reconstruction error calculation between input mesh and output mesh for person A with expression I.

| Person A with Expression I | | |
|---|---|---|
| **Distance** | **from Output to Input (mm)** | **from Input to Output (mm)** |
| Min | 0.000000 | 0.000000 |
| Max | 0.508250 | 0.038868 |
| Mean | 0.006017 | 0.000460 |

Table 4.6: Reconstruction error calculation between input mesh and output mesh for person A with expression II.

| Person A with Expression II | | |
|---|---|---|
| **Distance** | **from Output to Input (mm)** | **from Input to Output (mm)** |
| Min | 0.000000 | 0.000000 |
| Max | 0.935488 | 0.049997 |
| Mean | 0.124524 | 0.006655 |

Table 4.7: Reconstruction error calculation between input mesh and output mesh for person A with expression III.

| Person A with Expression III | | |
|---|---|---|
| **Distance** | **from Output to Input (mm)** | **from Input to Output (mm)** |
| Min | 0.000000 | 0.000000 |
| Max | 0.934314 | 0.049999 |
| Mean | 0.131328 | 0.007028 |

Table 4.8: Reconstruction error calculation between input mesh and output mesh for person B with expression I.

| **Person B with Expression I and Closed Eyes** | | |
|---|---|---|
| **Distance** | **from Output to Input (mm)** | **from Input to Output (mm)** |
| Min | 0.000000 | 0.000000 |
| Max | 0.302599 | 0.022376 |
| Mean | 0.006649 | 0.000492 |

Table 4.9: Average mesh processing times of all processes in our system for all four models.

| Process | Average Processing Time (s) |
|---|---|
| Surface Reconstruction | 5 |
| Registration | 11 |
| Modified RBF-based Deformation | 4 |
| Subdivision | 3 |
| Cylindrical Projection-based Adaptation | 7.5 |
| Post Mesh Refinement | 6 |
| Mesh Repairing | 4 |

As it is shown in the above tables, the number of vertices is only increased after subdivision method (almost five times) and repairing method. Among the processes involved in the system, feature point selection takes longer time (less than a minute) to be performed due to its semi-automatic nature and dependency on user input. Moreover, a color scale is used to visualize the difference from the input surface to the output surface as displayed in Figure 4.22. This figure shows that the reconstruction error is very minor, mostly near zero and also the preceding tables

show very small quantitative error. Hence, the preceding tables and the following figure prove that the facial region is being well adapted to the input face in all four cases.

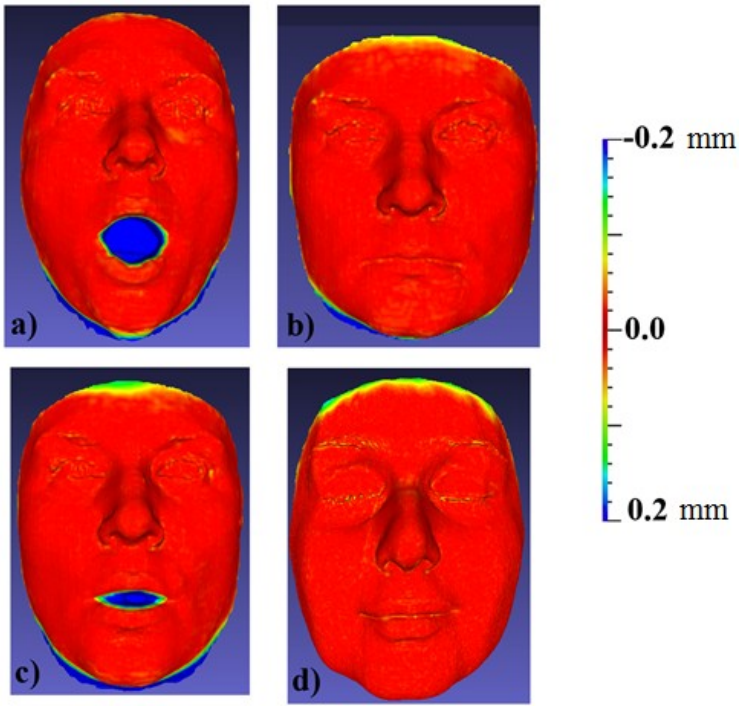


Figure 4.22 : Color scale of the difference from the input surface to the output surface in the range [-0.2 0.2] for person A with expression a) I, b) II, c) III, and d) person B with expression I. In this color scale, pure red represents zero distance between two surfaces and colors from red to blue represent increase in distance.

As it is demonstrated in Figure 4.22, the error mainly exists around the mouth area for the input faces which have open mouth (see Figure 4.22 (a) and (c)). Such error is mostly because of the additional parts (undesired) which are added to the input face after filling gaps inside the mouth through surface reconstruction stage. So, it means that our output mesh is quite similar to the original expression in the input face. After mouth region, nearby head border represents higher error due to modification applied to the surface by post refinement method to obtain

smooth surface at border line. Also, eyes and eyebrows areas have the structures that are very different between the generic and the input models which cause minor reconstruction errors.

## 4.3 Discussion

By performing tests on various input models of the system, we found out that the system offers the main properties of an ideal system including efficiency due to processing huge models within reasonable time and space and robustness due to constructing parameterized animated head for any input model.

Our system provides the convergence of a generic mesh to match the input data to obtain highly accurate surface through registration process followed by two-step adaptation including feature-based and surface point-based. The sequential post refinement procedure which treats surface disconnectivity is designed in a way that it doesn't affect or smooth the features and skin of facial region which must remain intact. To prove that our method produces good visual results, we implemented HC Laplacian smooth method [81] which attempts to globally smooth the surface and compared the result of HC Laplacian smooth method with our result as illustrated in Figure 4.23. While HC Laplacian smooth method provides a smooth surface after a couple of iterations, it distorts the fine facial features and skin characteristics of the model.

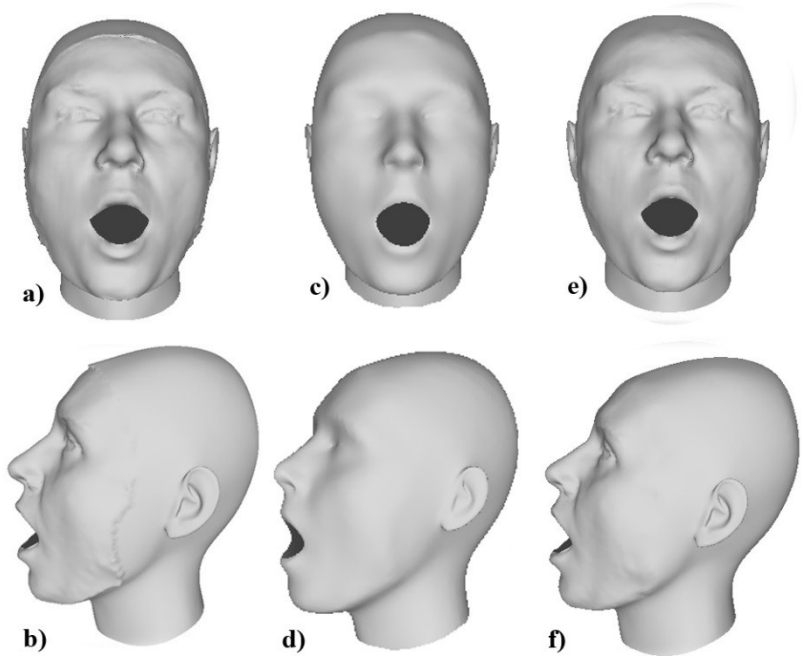


Figure 4.23: Smoothing results on a-b) surface point-based adapted model after c-d) HC Laplacian smooth method [81], e-f) our proposed smoothing method.

After evaluation of output meshes (obtained by our smoothing method) from different input faces, we also found out that despite obtaining a good smooth result near the border lines, the considered region for iterative averaging neighboring points is still quite narrow. Moreover, facial and the remaining areas have different skin characteristics (Figure 4.24) which cause an apparent stripe between them in some cases which might be required to be resolved for some applications.

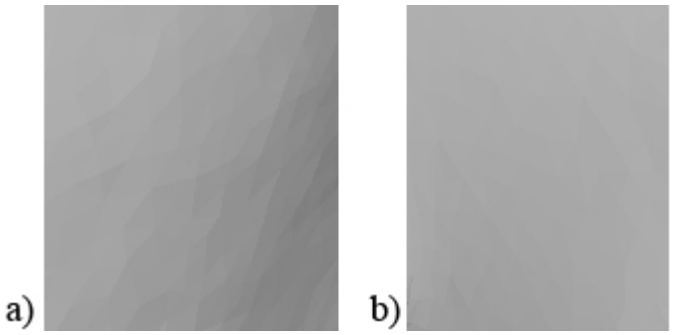


Figure 4.24: Skin representations from a) cheek area and b) back of the head area.

Also, the mesh repairing method which is primarily employed for handling self-intersection defect caused by wrong point projection especially around the nose in our case, has advantages over other techniques such as the work proposed by Soon and Lee [46] and surface reconstruction algorithms in reconstructing a clean mesh without flaws. Soon and Lee [46] propose to use a second cylinder line placed closer to nose based on positions of a few feature points so that points can properly be projected. Therefore, all the output models would share a common pattern (e.g. number of vertices) after mesh repairing. However, this method is not always faithful to produce a mesh without self-folding triangles around the nose and it is limited to the nose region and the rest of head mesh which might contain defects is ignored.

On the other hand, surface reconstruction techniques such as Poisson reconstruction technique are not quite suitable for attaining a clean, watertight mesh either. The majority of such algorithms are focused on surface reconstruction from points while handling included noise. These algorithms don't guarantee to preserve the structure of the input mesh. In other words, fine features of the input might be lost and partial or complete input remeshing might happen. Another drawback of such a reconstruction method is that it might considerably increase the number of mesh vertices as shown in Figure 4.25, the number of sample mesh vertices is

originally 32425, but it increases up to 107998, almost twice, after Poisson reconstruction method. Also, surface distortion occurs around the neck region. Hence, the aforementioned issues make the Poisson reconstruction method not a proper option for clean recreation of a mesh.

In contrast with the preceeding approaches, our employed method not only causes a slight increase in the number of vertices up to 33880 ,but also it doesn't distort the shape of the neck. Such an approach also reconstructs the surface smoother specifically around the nose as illustrated in Figure 4.25 compared with Poisson reconstruction method.

Another issue that we found out after evaluation of the system outputs comes from not properly or fully acquiring 3D range data by capturing resources. So after surface reconstruction, symmetry requirement might not be preserved between the available parts and their corresponding reconstructed parts especially in the facial region around the nose, which would degrade the realism of head models obtained by our system.

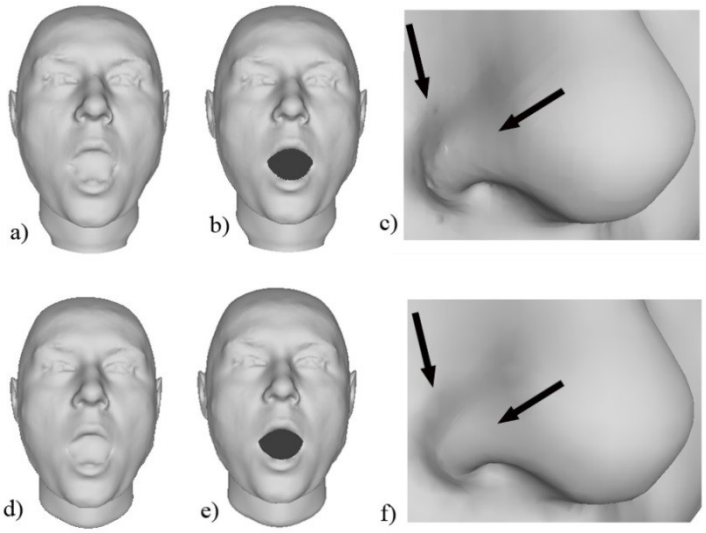


Figure 4.25: Mesh representation after surface repairing methods. a-c) obtained surface after Poisson reconstruction method, d-f) reconstructed surface after employing our method.

# Chapter 5. Conclusion

In this chapter we summarize and conclude the work developed and some gained insights within this dissertation. We also outline our key contributions and point out the potential and future work direction.

## 5.1 Summary

This thesis proposes a robust system for parameterized generation of realistic animation head model which is easily adapted to individual's face with relatively fast performance. Our system accepts two different kinds of input including point clouds and mesh structure plus one generic mesh containing animation structure. Mesh manipulation is primarily based on feature points extracted by a semi-automatic method which leads to achieving reliable results. Such manipulation named feature-based adaptation provides a rough matching between models using a modified RBF-based deformation. Fine matching between models is provided by surface point-based adaptation which employs a cylindrical projection-based method. This two-step adaptation plus mesh subdivision guarantees dense correspondences between a generic model and input data. To entirely make mesh surface smooth and natural looking, some post refinement procedures based on RBFs deformation and averaging neighboring points are leveraged. Lastly, the defects on the mesh surface generated through the preceding processes are resolved by performing an automatic repairing method. We also attempt to minimize the user interaction so that our system is automated as much as possible. Efficiency and robustness of the system is analyzed and proved through experiments performed on different individuals' faces varying in ethnic groups,

age range and expression. In overall, two individuals with three different expressions are reconstructed from our two input types. Reconstruction of publically available face data eliminates the cost associated with commercial equipment, while reconstruction of laser-scanned data is limited due to costly or sophisticated devices.

## 5.2 Contribution

According to the comparative study on a number of related approaches discussed in literature, our contributions can be itemized as follows:

- Establishing of consistent parameterization on expressed scanned model rather than doing consistent parameterization on neutral expression and then animating it artificially either using design tool or motion capture data.
- Introducing a practical, robust and generalized solution for reconstruction of whole head model by leveraging individual's front face data from several possible inputs in form of point clouds or triangular mesh while properly handling various surface defects, errors, missing parts (back of head) and/or triangles.
- Proposing a novel mouth movement simulation by integrating regional information to radial basis functions to robustly reproduce not only neutral expression of input faces, but also various facial expressions.
- Introducing a novel approach in order to resolve disconnectivity problem on the mesh surface caused by surface point-based adaption approach and consequently gaining smooth and natural looking surface which is suitable enough to be used by different practical applications.

## 5.3 Future Research

Even though this thesis provides a robust framework from different aspects, there are still some open issues and several research directions that can be pursued to enhance the obtained results and the performance of the proposed system. The following points can be taken into consideration when we decide to further develop the work accomplished in this thesis.

- Mesh surface can be further smoothen and uniform around the border between available facial area in the input model and missing regions (usually back of head). We have obtained a good result smoothing near the borderlines, but the considered region is still quite narrow. This can be improved by calculating new value of un-refined points based on a larger number of their corresponding refined points using a proper choice of interpolation technique (e.g. Kriging interpolation technique [82]).

- In order to resolve different skin characteristics between facial region and the rest of the head model, one solution is to transfer the skin from a small region (e.g. of the model cheek) and repeat it on the head excluding the facial region. The skin can be considered as a displacement vector between the refined model surface after fine matching technique and the surface of the model after subdivision method.

- To improve realism of head models generated by our system, symmetric surface reconstruction especially on the facial region can be integrated into the surface reconstruction stage. Since capturing devices might fail to properly or fully acquire 3D range data, symmetry requirement might not be preserved between the available parts and their corresponding reconstructed parts. In our technique the remaining parts are inherent from the generic mesh.

# References


[1] N. Magnenat-Thalmann and D. Thalmann, "The Direction of Synthetic Actors in the film Rendez-vous à Montréal," *Computer Graphics and Applications, IEEE,* vol. 7, pp. 9-19, 1987.

[2] T. DeRose, M. Kass, and T. Truong, "Subdivision surfaces in character animation," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 85-94.

[3] A. LeBlanc, P. Kalra, N. Magnenat-Thalmann, and D. Thalmann, "Sculpting with the'ball and mouse'metaphor," in *Proc. Graphics Interface*, 1991, pp. 152-159.

[4] G. Sannier and N. M. Thalmann, "A user-friendly texture-fitting methodology for virtual humans," in *Computer Graphics International, 1997. Proceedings*, 1997, pp. 167-176.

[5] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 187-194.

[6] J. R. Bergen and R. Hingorani, "Hierarchical motion-based frame rate conversion," Technical report, David Sarno Research Center1990.

[7] W.-S. Lee and N. M. Thalmann, "Head modeling from pictures and morphing in 3D with image metamorphosis based on triangulation," in *Modelling and Motion Capture Techniques for Virtual Environments*, ed: Springer, 1998, pp. 254-267.

[8] W. Lee, P. Beylot, D. Sankoff, and N. Magnenat-Thalmann, "Generating 3D virtual populations from pictures of a few individuals," in *Algorithms and Data Structures*, ed: Springer, 1999, pp. 134-144.

[9] W.-S. Lee, P. Kalra, and N. Magnenat-Thalmann, "Model based face reconstruction for animation," *Proc. Multimedia Modeling (MMM),* vol. 97, pp. 323-338, 1997.

[10] W.-S. Lee and N. Magnenat-Thalmann, "Fast head modeling for animation," *Image and Vision Computing,* vol. 18, pp. 355-364, 2000.

[11] Y. Lee, D. Terzopoulos, and K. Waters, "Constructing physics-based facial models of individuals," in *Graphics Interface*, 1993, pp. 1-1.

[12] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 55-62.

[13] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, "Making faces," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 55-66.

[14] J. Taylor, J.-A. Beraldin, G. Godin, L. Cournoyer, M. Rioux, and J. Domey, "NRC 3D imaging technology for museums & heritage," *Organized by MiraLab, University of Geneva and 3ème Cycle Romand d'Informatique (cuso),* 2002.

[15] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[16] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proceedings of the sixth ACM symposium on Solid modeling and applications*, 2001, pp. 249-266.

[17] T. K. Dey and S. Goswami, "Provable surface reconstruction from noisy samples," in *Proceedings of the twentieth annual symposium on Computational geometry*, 2004, pp. 330-339.

[18] L. Zhang, B. Curless, and S. M. Seitz, "Rapid shape acquisition using color structured light and multi-pass dynamic programming," in *3D Data Processing Visualization and Transmission. Proceedings. First International Symposium on*, 2002, pp. 24-36.

[19] M. Proesmans and L. Van Gool, "Reading between the lines—a method for extracting dynamic 3D with texture," in *Proceedings of the ACM symposium on Virtual reality software and technology*, 1997, pp. 95-102.

[20] P. Jia, *Real-time full-field range-sensing for three-dimensional surface-shape measurement* vol. 68, 2007.

[21] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559-568.

[22] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth, "Automatic reconstruction of personalized avatars from 3D face scans," *Computer Animation and Virtual Worlds,* vol. 22, pp. 195-202, 2011.

[23] (2012). *IMAGEMETRICS*. Available: http://www.image-metrics.com/livedriver/

[24] T. Kurihara and K. Arai, "A transformation method for modeling and animation of the human face from photographs," in *Computer Animation'91*, 1991, pp. 45-58.

[25] H. H. Ip and L. Yin, "Constructing a 3D individualized head model from two orthogonal views," *The visual computer,* vol. 12, pp. 254-266, 1996.

[26] Y. Zhang, T. Sim, and C. L. Tan, "Rapid modeling of 3D faces for animation using an efficient adaptation algorithm," in *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, 2004, pp. 173-181.

[27] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *ACM Siggraph Computer Graphics*, 1986, pp. 151-160.

[28] S. Coquillart, *Extended free-form deformation: a sculpturing tool for 3D geometric modeling* vol. 24: ACM, 1990.

[29] G. Farin, "Surfaces over Dirichlet tessellations," *Computer Aided Geometric Design,* vol. 7, pp. 281-292, 1990.

[30] N. Sukumar, "Sibson and non-Sibsonian interpolants for elliptic partial differential equations," in *Proceedings of the first MIT Conference on Fluid and Solid Mechanics*, 2001, pp. 1665-1667.

[31] M. J. Powell, "Radial basis functions for multivariable interpolation: a review," in *Algorithms for approximation*, 1987, pp. 143-167.

[32] D. Fidaleo, J.-y. Noh, T. Kim, R. Enciso, and U. Neumann, "Classification and volume morphing for performance-driven facial animation," in *International Workshop on Digital and Computational Video*, 2000.

[33] J.-y. Noh, D. Fidaleo, and U. Neumann, "Animated deformations with radial basis functions," in *Proceedings of the ACM symposium on Virtual reality software and technology*, 2000, pp. 166-174.

[34] J.-y. Noh and U. Neumann, "Expression cloning," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 277-288.

[35] G. Rong, Y. Cao, and X. Guo, "Spectral surface deformation with dual mesh," in *Proc. Internat. Conf. on Comput. Animation and Social Agents*, 2008, pp. 17-24.

[36] G. Rong, Y. Cao, and X. Guo, "Spectral mesh deformation," *The Visual Computer,* vol. 24, pp. 787-796, 2008.

[37] T. K. Dey, P. Ranjan, and Y. Wang, "Eigen deformation of 3D models," *The Visual Computer,* vol. 28, pp. 585-595, 2012.

[38] K. Hormann, B. Lévy, and A. Sheffer, "Mesh parameterization: Theory and practice," 2007.

[39] A. Asirvatham, E. Praun, and H. Hoppe, "Consistent spherical parameterization," in *Computational Science–ICCS 2005*, ed: Springer, 2005, pp. 265-272.

[40] E. Praun, W. Sweldens, and P. Schröder, "Consistent mesh parameterizations," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 179-184.

[41] V. Kraevoy and A. Sheffer, "Cross-parameterization and compatible remeshing of 3D models," in *ACM Transactions on Graphics (TOG)*, 2004, pp. 861-869.

[42] A. Golovinskiy and T. Funkhouser, "Consistent segmentation of 3D models," *Computers & Graphics,* vol. 33, pp. 262-269, 2009.

[43] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans," in *ACM Transactions on Graphics (TOG)*, 2003, pp. 587-594.

[44] A. Lee, H. Moreton, and H. Hoppe, "Displaced subdivision surfaces," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 85-94.

[45] I. Guskov, K. Vidimče, W. Sweldens, and P. Schröder, "Normal meshes," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 95-102.

[46] W.-S. Lee and A. Soon, "Mesh resolution augmentation using 3D skin bank," *Computer-Aided Design,* vol. 39, pp. 610-621, 2007.

[47] C. Loop, "Smooth subdivision surfaces based on triangles," 1987.

[48] L. Williams, "Performance-driven facial animation," in *ACM SIGGRAPH Computer Graphics*, 1990, pp. 235-242.

[49] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister*, et al.*, "Multi-scale capture of facial geometry and motion," in *ACM Transactions on Graphics (TOG)*, 2007, p. 33.

[50] H. Huang, J. Chai, X. Tong, and H.-T. Wu, "Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition," in *ACM Transactions on Graphics (TOG)*, 2011, p. 74.

[51] (2011). *VICON SYSTEMS*. Available: http://www.vicon.com

[52] (2011). *XYZ RGB SYSTEMS*. Available: http://www.xyzrgb.com/.

[53] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman*, et al.*, "High-quality passive facial performance capture using anchor frames," in *ACM Transactions on Graphics (TOG)*, 2011, p. 75.

[54] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," *ACM Transactions on Graphics (TOG),* vol. 29, p. 40, 2010.

[55] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," *ACM Trans. Graph.,* vol. 30, p. 77, 2011.

[56] H. Li, T. Weise, and M. Pauly, "Example-based facial rigging," *ACM Transactions on Graphics (TOG),* vol. 29, p. 32, 2010.

[57] H. Li, J. Yu, Y. Ye, and C. Bregler, "Realtime facial animation with on-the-fly correctives," *ACM Transactions on Graphics (TOG),* vol. 32, p. 42, 2013.

[58] Y. Zhang, E. C. Prakash, and E. Sung, "Efficient Modeling of An Anatomy-Based Face and Fast 3D Facial Expression Synthesis," in *Computer Graphics Forum*, 2003, pp. 159-169.

[59] T. Murali and T. A. Funkhouser, "Consistent solid and boundary representations from arbitrary polygonal data," in *Proceedings of the 1997 symposium on Interactive 3D graphics*, 1997, pp. 155-ff.

[60] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 9, pp. 191-205, 2003.

[61] T. Ju, "Robust repair of polygonal models," *ACM Transactions on Graphics (TOG),* vol. 23, pp. 888-895, 2004.

[62] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, 2002, pp. 428-441.

[63] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303-312.

[64] X. Sheng and I. R. Meier, "Generating topological structures for surface models," *Computer Graphics and Applications, IEEE,* vol. 15, pp. 35-41, 1995.

[65] G. Barequet and S. Kumar, "Repairing CAD models," in *Visualization'97., Proceedings*, 1997, pp. 363-370.

[66] S. Bischoff and L. Kobbelt, "Structure preserving cad model repair," in *Computer Graphics Forum*, 2005, pp. 527-536.

[67] M. Campen and L. Kobbelt, "Exact and Robust (Self-) Intersections for Polygonal Meshes," in *Computer Graphics Forum*, 2010, pp. 397-406.

[68] H. T.-M. Ngo and W.-S. Lee, "Feature-First Hole Filling Strategy for 3D Meshes," in *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, ed: Springer, 2013, pp. 53-68.

[69] M. Attene, "A lightweight approach to repairing digitized polygon meshes," *The Visual Computer,* vol. 26, pp. 1393-1406, 2010.

[70] A. Hornung and L. Kobbelt, "Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006, pp. 41-50.

[71] C. Shen, J. F. O'Brien, and J. R. Shewchuk, "Interpolating and approximating implicit surfaces from polygon soup," in *ACM Transactions on Graphics (TOG)*, 2004, pp. 896-904.

[72] Available: http://cyberware.com/

[73] T. Smith, R. S. Rana, P. Missiaen, K. D. Rose, A. Sahni, H. Singh*, et al.*, "High bat (*Chiroptera*) diversity in the Early Eocene of India," *Naturwissenschaften,* vol. 94, pp. 1003-1009, 2007.

[74] A.-N. Ansari and M. Abdel-Mottaleb, "3D face modeling using two views and a generic face model with application to 3D face recognition," in *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance*, 2003, pp. 37-44.

[75] S. Berretti, A. Del Bimbo, and P. Pala, "3D Face Reconstruction from Two Orthogonal Images for Face Recognition Applications," *International Journal of Digital Library Systems (IJDLS),* vol. 1, pp. 42-58, 2010.

[76] R. Everson, "Orthogonal, but not orthonormal, procrustes problems," *Advances in computational Mathematics,* 1998.

[77] W.-S. Lee, M. Escher, G. Sannier, and N. Magnenat-Thalmann, "MPEG-4 compatible faces from orthogonal photos," in *Computer Animation, 1999. Proceedings*, 1999, pp. 186-194.

[78] D. Heylen, M. Poel, and A. Nijholt, "Exporting vector muscles for facial animation," in *Smart Graphics*, 2003, pp. 251-260.

[79] A. Korniłowicz, "Jordan curve theorem," *Formalized mathematics,* vol. 13, pp. 481-491, 2005.

[80] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: Measuring errors between surfaces using the hausdorff distance," in *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, 2002, pp. 705-708.

[81] J. Vollmer, R. Mencl, and H. Mueller, "Improved Laplacian smoothing of noisy surface meshes," in *Computer Graphics Forum*, 1999, pp. 131-138.

[82] W. C. Van Beers and J. P. Kleijnen, "Kriging interpolation in simulation: a survey," in *Simulation Conference, 2004. Proceedings of the 2004 Winter*, 2004.