



University of Bradford eThesis

This thesis is hosted in [Bradford Scholars](#) – The University of Bradford Open Access repository. Visit the repository for full metadata or to contact the repository team



© University of Bradford. This work is licenced for reuse under a [Creative Commons Licence](#).

MODELLING FACIAL ACTION UNITS
USING PARTIAL DIFFERENTIAL EQUATIONS

Nur Baini Binti ISMAIL

Submitted for the degree of
Doctor of Philosophy

Centre of Visual Computing
Faculty of Engineering and Informatics
University of Bradford

2015

Abstract

Nur Baini Binti Ismail.

Modelling Facial Action Units using Partial Differential Equations.

Keywords: geometric modelling, 3D face modelling, PDE method, action units, facial expressions.

This thesis discusses a novel method for modelling facial action units. It presents facial action units model based on boundary value problems for accurate representation of human facial expression in three-dimensions. In particular, a solution to a fourth order elliptic Partial Differential Equation (PDE) subject to suitable boundary conditions is utilized, where the chosen boundary curves are based on muscles movement defined by Facial Action Coding System (FACS). This study involved three stages: modelling faces, manipulating faces and application to simple facial animation. In the first stage, PDE method is used in modelling and generating a smooth 3D face. The PDE formulation using small sets of parameters contributes to the efficiency of human face representation. In the manipulation stage, a generic PDE face of neutral expression is manipulated to a face with expression using PDE descriptors that uniquely represents an action unit. A combination of the PDE descriptor results in a generic PDE face having an expression, which successfully modelled four basic expressions: happy, sad, fear and disgust. An example of application is given using simple animation technique called blendshapes. This technique uses generic PDE face in animating basic expressions.

To
my lovely and supportive husband, Azahan Ali,
my patient and understanding children, Irfan, Aisyah and Danish.

Acknowledgements

I would like to express my deep sense of gratitude to my supervisor, Prof. Hassan Ugail for his invaluable guidance, support and encouragement throughout this research. Special thanks go to Dr Gabriela Gonzalez for her patient guidance at the earlier stage of this research.

I would like to thank all the people in the Centre for Computer Visual Computing at University of Bradford, for their continuous support and valuable discussion. A special note of thanks goes to Zahra Sayed, Norsheen Hussain, and Ali Bukar.

Lastly, but most importantly, I would like to thank Ministry of Higher Education, Malaysia and Universiti Malaysia Terengganu for funding me to undertake this study.

Table of contents

Abstract	i
Dedications.....	ii
Acknowledgements	iii
Table of contents.....	iv
List of Figures.....	viii
List of Tables	xii
List of Abbreviations	xiii
List of Mathematical Symbols.....	xiv
1. Introduction.....	1
1.1 Background	1
1.2 Motivation.....	3
1.3 Objectives.....	6
1.4 Research contributions.....	7
1.5 Publications	8
1.6 Structure of the thesis.....	8
2. Facial animation.....	11
2.1 Introduction.....	11
2.2 The Facial Action Coding System (FACS).....	11
2.3 Geometry based facial animation	15
2.3.1 Key framing and blendshape interpolation.....	16

2.3.2 Direct parameterisation	17
2.3.3 Muscle based animation	18
2.4 FACS-based animation	20
2.5 Summary	22
3. Conventional methods of surface modelling and PDE-based face modelling.....	24
3.1 Introduction.....	24
3.2 Conventional methods of surface representation	25
3.3.1 Parametric surface.....	26
3.2.2 Bezier surface	27
3.2.3 B-Spline surface	29
3.2.4 NURBS surface	30
3.3 PDE surfaces	31
3.3.1 PDE surface patches	32
3.3.2 Solution to the PDE method.....	34
3.3.3 Boundary conditions modification	37
3.3.4 Advantages of the PDE method.....	40
3.4 PDE-based facial model	42
3.5 Animation of facial expressions based on FACS using conventional surface generation methods.....	44
3.6 Summary	46
4. PDE-based Facial Modelling for Animation.....	48
4.1 Introduction.....	48

4.2 Database of 3D faces	48
4.3 Boundary curves extraction	50
4.3.1 Pre-processing.....	51
4.3.2 Curve extraction.....	53
4.4 PDE-based face generation	56
4.4.1 Fourier coefficients of facial curves.....	57
4.4.2 Textured face models	59
4.5 Results and discussions	60
4.6 Summary	64
5. PDE-based Facial Expressions	66
5.1 Introduction.....	66
5.2 Facial expressions based on FACS	67
5.3 Facial action units extraction	70
5.3.1 Identification of facial muscles	70
5.3.2 Curve extraction.....	74
5.4 PDE descriptor for an action unit.....	78
5.5 Facial expression using PDE descriptor	80
5.6 Results and discussions	81
5.7 Summary	87
6. Animation of PDE-based Faces.....	88
6.1 Introduction.....	88
6.2 Facial animation using blendshapes	89

6.3 PDE-based face animation using blendshapes	91
6.4 Results and discussions	96
6.5 Summary	98
7. Conclusions	99
7.1 Summary	99
7.2 Contributions	100
7.3 Research limitations	102
7.4 Future works.....	103
Bibliography.....	105
Appendix A	116
Appendix B	117
Appendix C.....	118
Appendix D.....	119

List of Figures

Figure 2-1 The major muscles of the face extracted from [31].....	12
Figure 2-2 Relation between the scale of evidence and intensity scores taken from [50].....	15
Figure 2-3 Parke's face model of the same person in different shading, left was rendered using polygonal shading and right was rendered using Gouraud's smooth shading. The images were taken from [10].	16
Figure 2-4 PDE face with six facial expression namely (from left to right) joy, sadness, anger, fear, disgust and surprise. The face of expressions were taken from [47].....	18
Figure 2-5 Water's facial muscles distribution taken from [39].....	19
Figure 2-6 Facial expressions generated by Water's namely neutral, happy, anger, fear, surprise and disgust (from left to right). All facial expressions are extracted from [31].	20
Figure 3-1 Parametric surface patch taken from [71].	27
Figure 3-2 Illustration of position vectors and derivative vectors	35
Figure 3-3 A cylinder generated by four boundary curves. (a) boundary curves (b) PDE generated cylinder (c) PDE cylinder with texture.....	38
Figure 3-4 Shape manipulation by changing the positional curves: (a) and (b) are boundary curves and (c) and (d) are corresponding PDE surface patch respectively.	39
Figure 3-5 An example of complex object of PDE surface with two different views.	40
Figure 3-6 Boundary curves for surface generation of six patches object.....	40

Figure 3-7 PDE face model taken from [46]. (a) represent 28 boundary curves (b) the resulting PDE face	42
Figure 4-1 Example of 3D image taken from database	49
Figure 4-2 Example of face model taken from database. (a) two pod half-face image (b) face model with texture	50
Figure 4-3 Images of Eva in 3D FACS database	51
Figure 4-4 Standardized and normalized face mesh.....	52
Figure 4-5 Feature points extracted from [96].....	54
Figure 4-6 PDE face generated using 19 boundary curves, (a) randomly chosen points, and (b) uniformly chosen points	55
Figure 4-7 A neutral face and position of extracted curves,(a) generating boundary curves, (b) a face from the FACS database	56
Figure 4-8 PDE generated neutral face, (a) ten patches of continuous PDE surface (b) wireframe of full PDE face	58
Figure 4-9 Procedure for texturing PDE face, (a) two mesh overlapped together, where green mesh is PDE face (b) the closest mark with yellow vertices, and (c) PDE face with texture	60
Figure 4-10 Two meshes, PDE face (green mesh) and original face mesh overlapped to each other with three different cut-planes	62
Figure 4-11 Comparison of the accuracy between original mesh and PDE mesh for different cut-plane.	63
Figure 4-12 Flowchart shows methodology used for modelling PDE face	65
Figure 5-1 Some action units that relate to six basic expressions. This figure is reproduced from [99].....	67
Figure 5-2 Facial muscle anatomy taken from [51]	72

Figure 5-3 Facial musculature (extracted from [16]) given in (a) is compared to boundary curves in (b).	73
Figure 5-4 Facial feature points on Eva's face	75
Figure 5-5 Two meshes overlapped; neutral mesh and AU1 mesh	76
Figure 5-6 Boundary curves extraction process for AU1 (a) boundary curves from neutral face, (b) different mesh on forehead area of AU1 and (c) closest points indicated by yellow points.....	77
Figure 5-7 PDE generated facial action units (a) AU4 (b) AU12 (c) AU14 and (d) AU17	79
Figure 5-8 PDE generated facial expressions (a) happy (b) sad (c) fear (d) disgust	80
Figure 5-9 Comparison of the accuracy between original mesh of AU4 and PDE mesh of AU4 for three different cut-planes.	82
Figure 5-10 Comparison of the accuracy between original mesh of disgust and PDE mesh of disgust expression for three different cut-planes.....	83
Figure 5-11 Comparison of five different AUs using original mesh and generic PDE.	84
Figure 5-12 Flowchart shows methodology used for modelling and manipulating a PDE face.....	86
Figure 6-1 Overview of PDE face animation process.....	88
Figure 6-2 A portion of blendshape slider interface taken from [106].....	90
Figure 6-3 Target face and base face	93
Figure 6-4 Varying weighting value for happy expression from (a) 0 (b) 0.229 (c) 0.534 (d) 1	95
Figure 6-5 Blendshape sliders for fear, disgust and sad face expressions	96
Figure 6-6 Animation sequences for happy expression	97

Figure 6-7 Animation sequences for disgust expression..... 97

List of Tables

Table 2-1 Upper face action units adapted from [57]	13
Table 2-2 Lower face action units taken from [57]	14
Table 2-3 Miscellaneous actions [57].....	14
Table 5-1 AU combinations based on primary and auxiliary AUs taken from [99]	68
Table 5-2 Expression description adapted from [101].....	68
Table 5-3 Proposed AU combinations for facial expression.....	69
Table 5-4 Action unit related with facial expressions and their corresponding facial muscles adapted from [51] and [57].	71
Table 5-5 Action units and the corresponding boundary curves affected.....	74
Table 5-6 Action units and corresponding facial feature points affected	76

List of Abbreviations

AU	Action Unit
CAD	Computer Aided Design
CAGD	Computer Aided Geometric Design
CGI	Computer Generated Imagery
FACS	Facial Action Coding System
FAP	Facial Animation Parameter
NURBS	Non-uniform Rational B-Splines
PDE	Partial Differential Equation

List of Mathematical Symbols

$B_{i,m}(u)$	Bernstein polynomial
$N_{i,m}(u)$	B-Spline basis functions
$\underline{p}_{i,j}$	Control points
$\underline{X}(u,v)$	Parametric surface patch
$D_{u,v}^m$	Partial differential operator of order m in independent variables u and v
$\underline{R}(u,v)$	Remainder function
$\underline{F}(u,v)$	Vector-valued function of u and v

1. Introduction

1.1 Background

Geometric models can be either in two-dimensional (2D) or three-dimensional (3D) form. Geometric modelling includes both the definition of the geometry and the development of the design method [1]. Computer Aided Design (CAD) is the use of computer software and design for modelling the 3D virtual objects [2]. Conventionally, surface design in CAD is based on polynomial interpolation such as Bezier surfaces [3], B-Splines surface [4], and non-uniform rational B-Splines (NURBS) surface [5]. Usually, these conventional surface generation techniques use a polynomial based method and the surface is generated using a set of control points. In order to change the shape of the objects, the designer needs to manipulate the control points to get the desired shape. This conventional method has the disadvantage of being time consuming: complex objects are usually represented by a large number of control points and consequently, it is difficult to manipulate underlying geometry in a timely manner [6].

In order to overcome this limitation, Bloor and Wilson [7] introduced the Partial Differential Equations (PDE) method. Their pioneering work was firstly used for blending surfaces and the generation of free-form surfaces [8]. The PDE method regards the generated surfaces as the solution of boundary-value problems. More precisely, the PDE method considers a surface as the graphical representation of the solution to elliptical partial differential equations. Generating surfaces as the solution to boundary-value problem

results in smooth surfaces along the boundaries and at the same time, satisfies the continuity conditions of blend shapes. The PDE method gives flexibility to manipulate the object shape as it is based on boundary conditions imposed on each particular case.

The human face is a complex 3D geometric form [9] and is very flexible [10]. These characteristics make human facial modelling very challenging in computer graphics. Facial modelling is a technique for representing the face on a computer [11] and more attention is given to modelling 3D human faces as a 3D face model has the robust features of a human face [12]. It is hard to model a geometry with no rigid structure [10].

Facial expressions are important for communication between people in order to convey their emotional state [13]. Humans are special as they can understand other people's facial expressions by understanding a message communicated verbally or by nonverbal communication [14]. As humans become heavily dependent on computers, it will be useful if computers can synthesis human facial expressions in order to improve human-computer interaction [15].

In order to form a desired facial expression using a computer, there are many methods that can be used for describing the muscle movement on a human face. Two popular methods are MPEG-4 facial animation standard [16] and facial action coding system (FACS) [17]. MPEG-4 standard used the Facial Animation Parameters (FAP) coding system to defines facial motions by creating displacement on a predefined location on a face with respect to neutral face [18], while FACS described the facial expression in terms of action

units [17]. An action unit (AU) represents the simplest visible facial movement, which cannot be decomposed into more basic ones [17].

Facial animation is a branch in computer-generated imagery (CGI) and studies a way to design virtual faces for creating human-like facial expressions [19]. Animating a human face is a complex task that requires modelling and animating the motion of facial movement [20]. Facial expression animation is one of the challenging tasks in computer graphics as facial expressions are very complex and they are paramount in nonverbal communication channels [21].

3D face models are useful in extensive applications such as automatic facial recognition [22], facial animation [23], human-computer interaction ([24], [25]), avatar [26] and etc. In most application scenarios, human face modelling and animation get extra attention in computer graphics due to the diversity and complexity of facial expressions [27]. Advanced development of computer facial animation technique was driven by significant requirements in the entertainment industry such as films and video games, where face and facial expressions are important to describe the emotions of an animated character [28]. In communication applications, the interactive talking faces make human-computer interaction superior by providing a friendly interface [29].

1.2 Motivation

Generating realistic facial animation often involves complex modelling techniques based the structure of the facial model for determining its animation potential [30]. For example, to animate a desired facial expression, one can

choose either to build multiple face layers [31] or to carry out deformation on the face models [32]. Hence, the quality of facial animation is determined by both the employed methods of facial modelling and facial animation [33]. Since the pioneering work by Parke [10], significant research effort has been attempted for modelling the geometry of human faces including blendshapes, direct parameterisations and muscle based modelling [16].

Blendshape models provides a compressed representation of the facial expressions by specifying the blendshape weight [34]. Facial motion can be represented by varying the weights of linear combinations [35]. Blendshape technique is one of the popular facial animation due to its efficiency and intuitive controllability [36]. However, modelling a complex facial expression using a blendshape model is very time consuming and needs a lot of storage for large libraries of blendshapes [37].

Parametric models offer an alternative approach to a blendshape model as a parameterised face can be manipulated directly by changing a few corresponding parameters [38]. A parameterisation technique can divide a face into different small parts in order to give animators more control over facial configurations [39]. Most parametric models employ MPEG-4 Facial Animation Parameters [40] or FACS [41] for describing facial expressions. In contrast, it is hard to determine suitable parameters for controlling facial expressions of a parametric model [16] and manual tuning is required when adapting a new facial model to the existing parameterised model [42].

A muscle based model is a polygonal surface mesh that is constructed based on human facial anatomy [31]. The muscle based modelling imitates

the physical properties of the facial muscles and articulations [43]. Muscle based method produces a realistic 3D animation by precisely simulating the facial muscular tissue contraction for representing facial expressions [16]. Furthermore, facial expressions in a muscle based model are controlled by a set of parameters that are related to action units for describing muscle movement [17]. The muscle model is broadly used due to its compact representation and is not dependent on facial mesh structure [39]. Although, a muscle based model can generate a realistic face model, the modelling process of an anatomical facial structure is an extremely tedious task, which needs massive computation of the underlying structure and requires artistic skill to place the muscle points correctly to the face model [44].

Based on the three facial models mentioned above, it is concluded that the modelling of a human face needs further improvement in terms of representation of facial data, storage capacity for storing facial information and a well-defined set of control parameters for easy manipulation in order to save time. The researchers still need to find a more sophisticated technique for representing a 3D human face in an accurate and efficient way [45] and at the same time the face geometry can be manipulated easily [21]. The face model should also be compatible with animation systems to avoid a new face needing to be modelled for each time frame [32].

Thus, the research goal of this work addresses the above limitations in a previous 3D face model using a boundary value approach, to be exact, the PDE method. With the rapid development of computer graphics, the PDE based geometry modelling technique has recently been considered as a powerful tool for 3D face modelling compared to a polynomial based technique

such as Bezier, B-Spline and NURBS [46]. The PDE-based face is considered a powerful tool to describe face geometry because a very small parameter is needed to specify and generate a specific face with a specific expression that is produced with low cost [47]. Furthermore, integrating FACS on boundary curves in the PDE method would be advantageous as these boundaries play a significant role in facial modelling.

1.3 Objectives

The aim of this research is to develop a mathematical formulation based on a boundary-value problem for representing an action unit for a human face. In particular, muscle movement of action units is formulated as the solution to elliptical partial differential equations. This research will explore a technique using PDE based formulation to represent and model action units. A PDE-based face has some benefits such as efficient storage of facial data, smooth PDE surface with associated FACS based muscle movement, efficient manipulation of facial geometry and time saving when animating facial expressions.

The objectives of this research are summarized as follows:

- i. To utilize PDE based formulation for generating a smooth human face in 3D space and apply a simple texturing method on PDE face for a realistic looking face.
- ii. To exploit the formulation of the PDE method by representing facial muscle movement on human faces using a small number of control parameters, known as a PDE descriptor.

- iii. To generate various action units of a human face based on the use of PDE method by using a different PDE descriptor.
- iv. To explore different combinations of PDE descriptors for generating basic facial expressions of a human face.
- v. To apply the formulation of PDE descriptors in one of the simple animation technique by animating the facial expressions of PDE-based face.

1.4 Research contributions

This thesis offers the following contributions:

- i. The PDE-based face adopted in the work is achieved by mapping the boundary curves with key features of the face to generate a smooth face surface (Chapter 4).
- ii. Simple texturing method for realistic looking models (Chapter 4).
- iii. PDE descriptors that are used for representing different action units will reduce the storage capacity for storing the facial data (Chapter 5).
- iv. Generated PDE-based face of different facial expression by combining different action units is an effective way for manipulating a PDE face from a neutral configuration to an expressive face (Chapter 5).
- v. Animation of four different basic expressions using a generic PDE face will save a lot of time as a generic model can be used without manual adjustment to get desired facial expressions (Chapter 6).

1.5 Publications

The list of publication of the works in this thesis are listed as follows:

1. H. Ugail and N.B. Ismail, "Method of Modelling Facial Action Units using Partial Differential Equations", in Advances in Face detection and Facial Image Analysis (Springer, 2015), To Appear. (2015).
2. N. B. Ismail and H. Ugail, "Modelling of Facial Expressions using Partial Differential Equations", Under preparation, to be submitted to The Visual Computer (2015).
3. N. B. Ismail and H. Ugail, "Blendshape Based Facial Animation using Partial Differential Equations", Under preparation, to be submitted to The Visual Computer (2015).

1.6 Structure of the thesis

The rest of the thesis is structured as follows:

- Chapter 2, "**Face model and animation**", gives an introduction to the FACS and facial animation. The first section provides some details on action units including both parts of FACS classification; upper face action units and lower face action units. The second section examines various animation techniques based on geometric facial animation, which includes key framing and blendshapes interpolation, direct parameterization and muscle based animation. Some literature of FACS-based animation is also discussed in this chapter.

- Chapter 3, “**Geometric modelling of surfaces and human faces**”, presents the literature of surface generation including the conventional method in CAD and the proposed method, PDE method. More discussion focuses on the PDE method as it is the main approach of this work. The last part of this chapter gives some literature of work that has been done in modelling human faces using PDE surface generation.
- Chapter 4, “**PDE-based facial modelling**”, offers detail on the methodology of PDE face parameterisation. There are five procedures involved for parameterisation of PDE face: selecting 3D mesh from database, pre-processing, extracting boundary curves, solving elliptic PDE equation and texturing.
- Chapter 5, “**PDE-based facial expression**”, presents a methodology for manipulating a generic PDE face of neutral expression to a face with an expression. This chapter will demonstrate a technique to formulate a method to store the action units known as PDE descriptors. Manipulation of a generic PDE face from a neutral PDE-based face to a given action unit also will be explained in this chapter. Furthermore, this chapter also will be described a method for generating an expressive PDE-based face models by combining related PDE descriptors that results of generation of four different basic expressions namely happy, sad, fear and disgust.
- Chapter 6, “**Animation of PDE-based face**”, presents a technique using blendshapes and key-framing for animating basic facial

expressions by using the generic PDE face. The result is given graphically as animation sequences.

- Chapter 7, “**Conclusions**”, concludes the overall research that has been done. This chapter will summarise the research that has been done, state the contributions, give the research limitations and suggest some future works.

2. Facial animation

2.1 Introduction

Facial modelling can be grouped into two different approaches namely image-based facial modelling and geometric-based facial modelling [48]. In image-based facial modelling, the creation of a facial model is based on a collection of facial images captured from a real person's face [49]. In contrast, geometric-based facial modelling refers to generation and manipulation of geometry representation of 3D face models [11]. In geometric modelling, the detailed data for representing geometry of facial data can be obtained by photographic techniques, 3D digitizers, or 3D laser-based scan [17]. The approach used in this work is based on geometric modelling.

In this chapter, the related research papers is discussed. The first section provides some details on the Facial Action Coding System (FACS), as FACS is used to describe muscle facial movement. Then it is followed by a summary on the state of the art 3D facial animation that focuses on a geometric-based technique. Later, some literature of animation techniques based on FACS is given.

2.2 The Facial Action Coding System (FACS)

FACS was originally developed by a group of physiologists, Paul Ekman, Wallace Friesen and Joseph Hager, in the 1970s [50]. Since then, FACS has been widely used. It is a versatile method for measuring and describing facial

behaviour ([51], [52]). Facial behaviour is determined by measuring the stretch of facial muscles that affects the changes on the appearance of the face ([53], [54]). The measurement unit of FACS is called an Action Unit (AU). For example, AU1 is inner brow raiser, AU4 is brow lower, etc. There are 44 different action units defined by FACS, each of which corresponds to a given activity in a distinct muscle or group and produces characteristic facial distortions [55].

FACS is classified action units into two main groups: upper face action units and lower face action units. These action units are based on muscle movement [24] and the type of muscle [56]. In fact, the muscle can be categorised according to the fasciculi (the individual fibers of the muscles) that may be parallel, linear, oblique, or spiralled relative to the direction of pull where it is attached [57]. The main muscles on a human face, according to Waters [31], are shown in Figure 2-1.

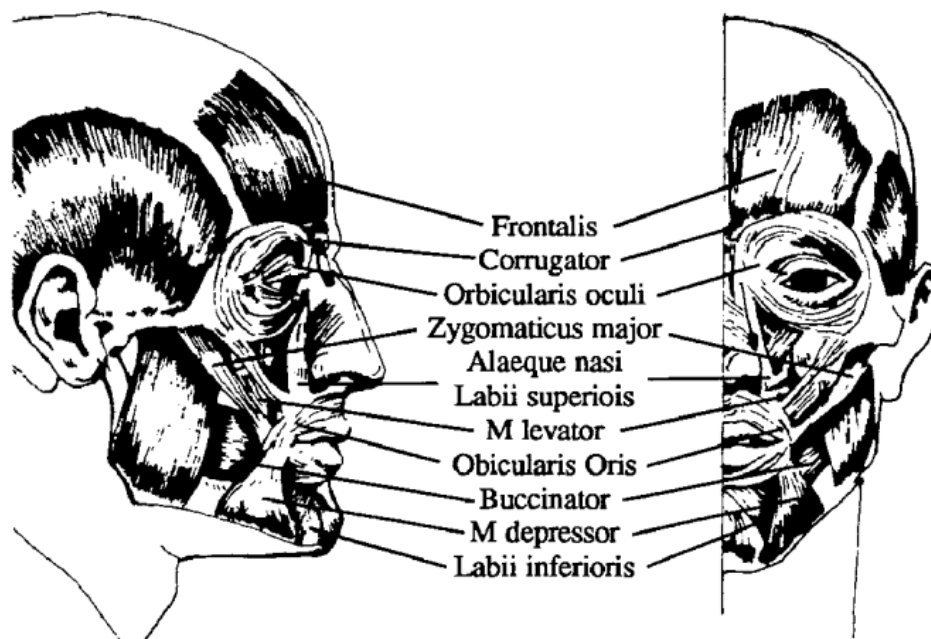


Figure 2-1 The major muscles of the face extracted from [31]

Based on Figure 2-1, the upper facial muscles are responsible for changing the appearance of the eyebrows, forehead, and upper and lower lids of the eyes [31]. The related contraction of a specific of upper face facial muscles is adapted from [57] is shown in Table 2-1.

Table 2-1 Upper face action units adapted from [57]

AU Number	FACS name	Facial muscle
1	Inner brow raiser	Frontalis, Pars Medialis
2	Outer brow raiser	Frontalis, Pars Medialis
4	Brow lowerer	Depressor Glabella, Depressor Supercilli, Corrugator
5	Upper lid raiser	Levator Palpebrae Superioris
6	Cheek raiser	Orbicularis Oculi, Pars Orbitalis
7	Lid tightener	Orbicularis Oculi, Pars Palebralis

On the other hand, the lower face action units, are classified in five major groupings as follows [31]:

- *Up/down actions*; they move the face upwards towards the brow and conversely towards the chin, i.e. AU9 for nose wrinkler.
- *Horizontal actions*; the muscles that contract horizontally towards the ears and conversely towards the centre line of the face, i.e. AU14 or known as dimpler.
- *Oblique actions*; they contract in an angular direction from the lips, upwards, and outwards to the cheekbones, i.e AU12 for lip corner puller.
- *Orbital actions*; they are circular or elliptical in nature, and run round the eyes and mouth, i.e. AU18 for lip pucker.
- *Sheet muscles*; carries out miscellaneous actions, particularly over the temporal zones and platysma muscle, i.e. AU19 (tongue show) and AU21 (neck tighten).

The full list of lower facial action units taken based on [57] are described in the Table 2-2.

Table 2-2 Lower face action units taken from [57]

AU Number	FACS name	Facial muscle
9	Nose wrinkle	Levator Labii Superioris, Alaeque Nasi
10	Upper lip raiser	Levator Labii Superioris, Caput Infraorbitalis
11	Nasalobial furrow deepener	Zygomatic Minor
12	Lip corner puller	Zygomatic Minor
13	Sharp lip corner	Caninus
14	Dimpler	Buccinator
15	Lip corner depressor	Triangularis
16	Lower lip depressor	Depressor labii
17	Chin raiser	Mentalis
18	Lip puckerer	Incisivii Labii Superioris, Incisivii Labii Inferioris
20	Lip stretcher	Risorius
22	Lip funneler	Orbicularis Oris
23	Lip tightener	Orbicularis Oris
24	Lip pressor	Orbicularis Oris
25	Lips part	Depressor Labii, or Relaxation of Mentalis or Orbicularis Oris
26	Jaw drop	Masseter, Temporal and Internal Pterygoid Relaxed
27	Mouth Stretch	Pterygoid, Digastric
28	Lip suck	Orbicularis Oris
41	Lip Droop	Relaxation of Levator Palpebrae Superioris
42	Slit	Orbicularis Oris
43	Eyes closed	Relaxation of Levator Palpebrae Superioris
44	Squint	Orbicularis Oris
45	Blink	Relaxation of Levator Palpebrae and Contraction of Orbicularis Oris, Pars Palpebralis
46	Wink	Orbicularis Oris

In total, there are 30 AUs that anatomically relate to a contraction of a specific set of facial muscles (as listed in Table 2-1 and Table 2-2). Another 14 AUs, are known as miscellaneous AUs that is shown in Table 2-3. They are not based on their muscular basis and their specific behaviour is not entirely distinguished [31].

Table 2-3 Miscellaneous actions [57]

AU Number	Description
8	Lips towards
19	Tongue show
21	Neck tighten
29	Jaw trust

30	Jaw sideways
31	Jaw clench
32	Bite lip
33	Blow
34	Puff
35	Cheek suck
36	Tongue bludge
37	Lip wipe
38	Nostril dilate
39	Nostril compress

An additional feature of FACS is that of scoring measurements for the intensity with which action units are performed. These scores are scaled from A to E. Figure 2-2 shows the relationship of intensity scores and scale of evidence [50].

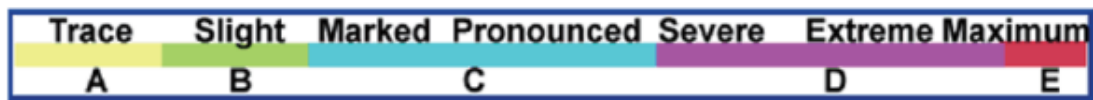


Figure 2-2 Relation between the scale of evidence and intensity scores taken from [50]

The terms A, B, C, D and E in Figure 2-2 respectively describe the intensity as classified from visible to maximum intensity of each AU [50]. However, the work in this thesis only considers the highest intensity of selected action units when modelling each action unit. For instance, the AU4E indicates AU4 with an intensity of E, which means that the brow was pulled together or lowered with maximum intensity.

2.3 Geometry based facial animation

This section will survey facial animation techniques that are based on facial geometry. Geometry based facial modelling and animation techniques include key-framing and blend shape interpolation, direct parameterization, and

muscle based animation ([58],[42]). For a full survey of facial animation method, a book by [30] can be referred to as it has reviewed all the complete techniques for facial animation.

2.3.1 Key framing and blendshape interpolation

A key-framing method with interpolation is by far the simplest and oldest approach [11], but still frequently used by animators because of its intuitive and flexible controls [59]. In fact, the pioneer work for facial animation demonstrated the use of this approach by Parke [10]. In particular, Parke describes a facial geometry in at least two different expressions and uses linear interpolation for manipulating the face from one expression to the other [43]. He used relatively small numbers of polygons to represent the face (as shown in Figure 2-3) and explicitly key framed the data of animation [11]. Even using a small number of parameters, Parke's face model can be rendered smoothly and facial features (eyes, nostrils, teeth, eyelashes and eyebrows) were added for a realistic head model as illustrated in Figure 2-3. However, this approach is not practical for complex facial expressions and needs a lot of storage to store the animation data [60].

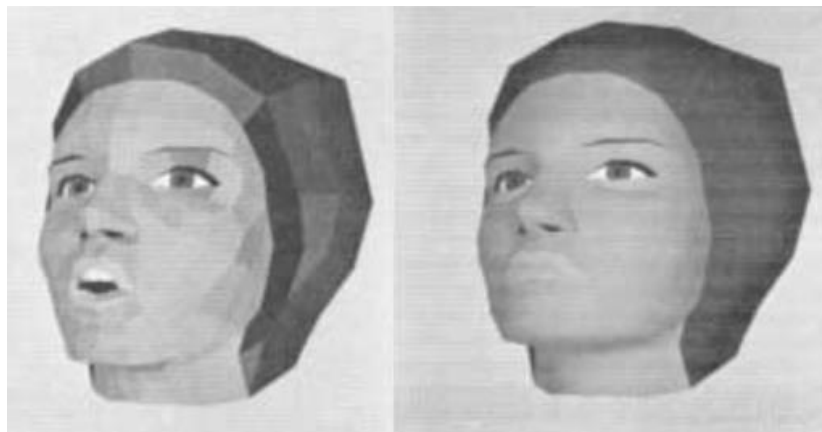


Figure 2-3 Parke's face model of the same person in different shading, left was rendered using polygonal shading and right was rendered using Gouraud's smooth shading. The images were taken from [10].

Although more sophisticated animation techniques have been developed to date, key-framing and blendshape interpolation remain popular among animators and are commonly used in commercial animation software packages such as Maya and Studio Max [35]. More discussion on blendshapes will be covered in Chapter 6.

2.3.2 Direct parameterisation

In order to overcome the limitations of key framing and blendshapes, Parke derived a method called direct parameterisation [38], where facial expressions can be generated and manipulated by specifying appropriate sets of parameters [61]. The facial expressions are represented by a small parameter even when the face is represented using polygonal surface [11]. Thus, a wide range of facial expressions can be animated by combining related parameters with a relatively low cost of computation [62].

There are some limitations inherited from a direct parameterisation animation technique. This animation technique is not suitable for a complex facial model that consequently produces a very large number of vertices [42]. The choice of the parameters set depends on the facial topology, it is impossible to generate a complete set of facial expressions [48]. Thus, this limitation will lead to unnatural expressions especially when manipulating a vertex that is represented by the same parameter [63]. Furthermore, a new face model requires a manual manipulation task in order to adapt the generic face to an existing parameterisation animation scheme [42].

However, the work by Sheng et al [47] eliminated some limitations derived from a parameterisation animation technique. They demonstrated that

the PDE method can produce and animate a 3D face using a small number of parameters. They successfully animated six facial expressions as given in Figure 2-4.



Figure 2-4 PDE face with six facial expression namely (from left to right) joy, sadness, anger, fear, disgust and surprise. The face of expressions were taken from [47].

The facial animation above uses a direct parameterisation scheme by associating PDE boundary curves with MPEG-4 Facial Animation Parameters (FAPs).

2.3.3 Muscle based animation

A muscle based approach simulates real human face muscles for animation purposes [63]. The earliest approach towards a muscle based model was proposed by Platt and Badler [64], where animation was applied by a set of facial action units using a contraction algorithm. They constructed an abstract muscle fibre structure that was connected by spring for connection of skin, muscles and bone nodes in their system [65]. Later, Water [31] presented an animation of a human face that used FACS for describing the muscle movement and representing facial expression. Water's muscle models as shown in Figure 2-5 are the basis principle of widely used physic-based models [42].

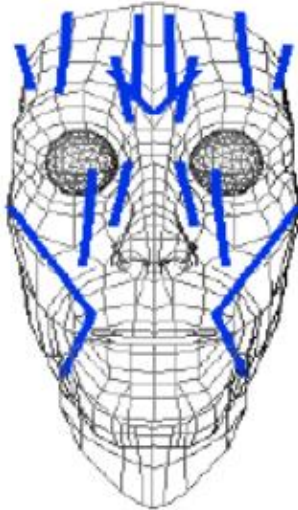


Figure 2-5 Water's facial muscles distribution taken from [39]

The physic-based model approach is divided into three categories [48]:

- mass spring systems. These spring systems are applied on a face to force simulation of skin deformation.
- layered spring meshes. They extend mass spring muscles into three connected mesh layers for realistic facial behaviour.
- vector muscles. They define a vector field as muscle for contraction of mesh vertex.

Among these three approaches, a vector muscles model uses FACS for animating human emotions such as anger, fear, surprise, joy and happiness by using orbicularis oris muscles and linear muscles (blue line in Figure 2-5). The facial expression that was generated by Waters work [31] is given in Figure 2-6.



Figure 2-6 Facial expressions generated by Water's namely neutral, happy, anger, fear, surprise and disgust (from left to right). All facial expressions are extracted from [31].

Animation of facial expression in Figure 2-6 is achieved by changing the contraction of muscle parameters, where these parameters are embedded under the face mesh with an assumption that the muscles are placed correctly under the face mesh [66]. However, finding correct positions to anatomically locate the position of vector muscles is a daunting task [48] as it is not intuitive or consistent from one model to another face model.

2.4 FACS-based animation

Based on animation techniques discussed in the previous section, there exists some facial models that utilized FACS as a basis for control parameters as shown in [64] and [31]. To be exact, muscle based animation employs action units in the face models for animating facial expression by changing the relevant parameters. Since then, more research has adopted FACS in the face model for generating realistic facial animation. This section will review some of related works that employ FACS as parameter controls in facial animation.

The work in [61] introduced an anatomically-based arrangement of muscle model, along with a physically-based tissue model. The tissue model is developed by considering the biomedical of tissue mechanics for more realistic surface deformation [11]. The face model also assimilates a muscle

model that was introduced in [31] through FACS abstraction in order to provide a more convenient interface to the face model for representing facial expressions. This physical-based tissue model contains a total of 960 polygons with approximately 6500 spring units. Hence, a lot of storage is needed to store the face data and it needs a high speed computer for simulation of facial expressions.

Later, a face model based on non-linear spring frames is proposed in [44], that can simulate the dynamic of real skin. This 3D physical based model is also based on human facial anatomy and has the 3D structure of muscle and skin. Facial muscles are modelled as forces for deforming the spring mesh. They use ten major functional facial muscles from [31] which is based on FACS and simulates the muscle contraction for generating a facial expression. By determining the facial muscles parameters, this non-linear face model can analyse the relationship between the facial skin deformation and inside the mesh face. In contrast, this work also inherits the same problems as previous work in [61], which it also requires a big amount of storage.

A 3D model of a specific person is presented in [37] for producing natural looking animations where facial geometry is reconstructed individually from a laser scan range. Three kind of facial muscle models are developed for simulating facial muscle contraction. They do not use all 46 AUs in FACS but simplify the model by selecting 18 action units that strongly influence the facial expression. Facial expression is parameterised by FACS, where a simple linear combination of action units approximates the muscle contraction. However, simulation of facial expression of such a model increases computing time and speed due to multiple layers of the face model.

All facial models discussed above use Water's muscle model as shown in Figure 2-5. It is shown that using FACS in a muscle based model, the facial expression can be animated by combining one or more action units. Instead of using muscle-based modelling, the work in this thesis will adopt Water's muscle model to a parametric surface in order to overcome the problems with storage capacity and computing time. For this purpose, parameterisation of a human face will use the PDE method, by embedding Water's muscle model to the boundary curves. More discussion on the PDE method will be presented in Chapter 3. Then in Chapter 4, a detailed methodology of PDE face parameterisation is given. However, the goal of this work is not to develop a highly realistic looking facial model, but a model that would be able to be realistic in a natural way.

2.5 Summary

In this chapter, the discussion begins with a detailed explanation about FACS, which includes both FACS classifications; upper face action units and lower face action units. Facial muscles corresponding to each action unit are also given for both categories as a reference of Chapter 5. A knowledge of muscle distribution is needed when extracting boundary curves for representing the action units.

The second section of this chapter has examined three different techniques of geometry facial modelling and animation: key-framing and blendshapes interpolation, direct parameterisation and muscle based animation. After reviewing those modelling techniques, it is shown that facial

modelling is based on anatomy and a physics based model successfully represents a realistic facial animation of expressions based on FACS. This is supported by some literatures] given in the last section.

Thus, the approach that has been used in this research is an integrated approach of all three techniques discussed above. In particular, our method adapted muscle movement from the model-based approach by Waters [31] when locating the facial features that are related to FACS. Then, a face model is parameterised using the PDE method as the PDE method only stores a small amount of parameters for representing the face and can be manipulated by changing the related parameters. For animation purposes, this work will employ key-framing and blendshape interpolation as it is the simplest animation technique for animating basic facial expressions.

3. Conventional methods of surface modelling and PDE-based face modelling

3.1 Introduction

In the previous chapter, various animation techniques in geometric modelling have been discussed including key-framing and blend shape animation, direct parameterization and muscle-based animation. The goal of these animation techniques is to control the geometry of the face until the rendered facial geometry has the desired expressions and textures in each frame of the animated sequence [30]. Furthermore, the face model that is used in the modelling stage will determine the animation potential [42]. Hence, the correct choice for a facial modelling method is important so that the face model can be easily manipulated for desired expressions and animated in realistic movement.

After reviewing different approaches for facial animation, blend shapes interpolation suits the nature of this research considering its popularity, efficiency, and intuitive controllability. Here popularity refers to a highly preferred method by the animator [67], as blend shapes are a standard component of commercial animation packages such as Maya and Softimage [68]. However, a traditional blend shapes method which is based on polygonal models has disadvantages when animating desired animation due to tedious and labouring work [67] for correcting vertex positions manually. Considering this disadvantage, some works used parametric models for more intuitive surface interpolation and manipulation. For example, work by Deng et. al [59]

used a NURBS face model composed of 46 blend shapes while Li and Deng [69] used spline base curves for smooth animation.

In this thesis, a face model is based on a parametric surface, to the exact PDE parametric surface. The purpose of this chapter is to review various modelling techniques that have been used in computer graphics for generating parametric surfaces. It begins with a general discussion about conventional method for generating parametric surface including Bezier, B-Spline and NURBS. Then, the mathematical concepts of PDE method for surface generation are discussed in detail as the PDE method is the main approach of this research. Later, an overview of previous approaches for modelling faces using PDE surface generation is given.

3.2 Conventional methods of surface representation

In geometric modelling, a variety of surfaces are used by researchers for representing surfaces. The most common methods are implicit equations and parametric functions.

The implicit surface is represented in general form as $f(x, y, z) = C$ in 3D-space. Note that x , y and z are Cartesian coordinates and C takes any constant. If the equation is linear and it is given by $ax + by + cz = 0$, then it is a plane. If the equation is in second order of a quadratic surface, the equation being $x^2 + y^2 + z^2 = r^2$, then it represents a surface of sphere. These two examples of equations are implicit because one can test directly either if any given points lie on the surface or not. If the chosen points satisfy $f(x, y, z) = 0$,

then it is concluded that the points lay on the surface. The implicit surface is rendered using various techniques such as polygonization, ray-tracing and contours [70].

A parametric surface is more popular and known as polynomial based method. This polynomial based method can be used to generate complex geometries that are defined by a set of control points. The popularity of parametric surface representation is due to the fact that they can be easily manipulated. The work in this thesis used parametric surfaces for face modelling. Thus, a detailed discussion of a parametric surface is outlined in the next subsection consisting of various methods of conventional surface. These conventional methods refer to Bezier surface, B-Spline surface, and NURBS surface. At the end of this chapter, an alternative method to conventional surface that is known as PDE based surface generation is presented.

3.3.1 Parametric surface

Parametric surface representation consists of mapping two parameters u and w , into three function $x = x(u, w)$, $y = y(u, w)$ and $z = z(u, w)$. In general, a parametric surface takes the form:

$$\underline{X}(u, w) = \begin{pmatrix} x(u, w) \\ y(u, w) \\ z(u, w) \end{pmatrix}. \quad (3.1)$$

It is basically a mapping of a domain $\Omega \subset R^2$ to R^3 . Figure 3-1 (extracted from [71]) shows an example of a parametric patch. Some of the most popular techniques for parametric surface generation are Bezier surfaces, B-Spline

surfaces and Non-uniform rational B-Spline (NURBS), which are discussed in detail in the following section.

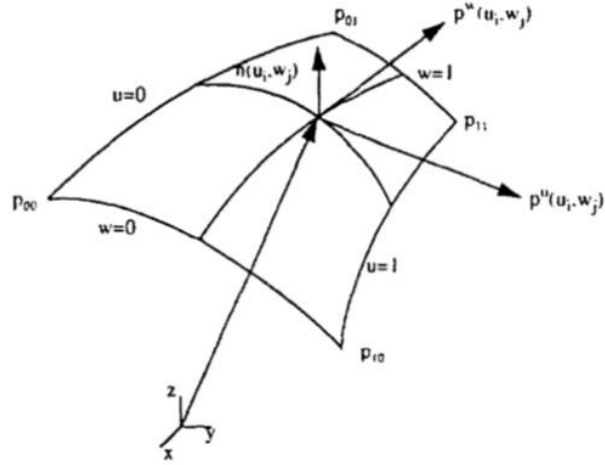


Figure 3-1 Parametric surface patch taken from [71].

3.2.2 Bezier surface

A Bezier surface is defined by a two-dimensional set of control vertices and Bernstein polynomial basis function [72]. If the Bezier surface of order $(m \times n)$ is denoted by $\underline{X}(u, w)$, then the equation of Bezier surface is given by

$$\underline{X}(u, w) = \sum_{i=0}^m \sum_{j=0}^n \underline{p}_{i,j} B_{i,m}(u) B_{j,n}(w) \quad (3.2)$$

where

- $\underline{p}_{i,j}$ denotes the control points. $\underline{p}_{i,j}$ are the vertices of control polygon that form an $(m+1)$ by $(n+1)$.
- $B_{i,m}(u)$ is a Bernstein polynomial in u -direction. The Bernstein polynomial is defined explicitly by

$$B_{i,m}(u) = \binom{m}{i} u^i (1-u)^{m-i} \quad (3.3)$$

where the binomial coefficients are given by

$$\binom{m}{i} = \begin{cases} \frac{m!}{i!(m-i)!} & ; \quad 0 \leq i \leq m \\ 0 & ; \quad otherwise \end{cases} \quad (3.4)$$

- Similarly, $B_{j,n}(w)$ is Bernstein polynomial in w -direction.

These Bernstein polynomials, $B_{i,m}(u)$ and $B_{j,n}(w)$, are also known as Bezier basis function. The Bezier basis functions are non-negative over the interval (0,1) and their summation in interval (0,1) is one ($\sum_{i=0}^m B_{i,m}(u) = 1$ and

$$\sum_{j=0}^n B_{j,n}(w) = 1).$$

Bezier surfaces possess convex hull property, which means the surface remains within the convex hull of control points. Bezier surface also offers an intuitive way of creating and manipulating a generated surface by manipulating the control points [71]. However, the object represented with a massive number of control points tends to be impractical, time consuming and would require a vast amount of storage space [73].

Another drawback of Bezier surface is that their control points have global control. Global control means the adjustment of a small set of control points have global control. Global control means the adjustment of small set of control points will affect the overall shape of the object. The problem occurs when one cannot make minor adjustments to create a desired shape. Local control brings up piecewise triangular polynomials or bi-polynomials to define

a certain surface. The local control techniques include B-Spline and NURBS surfaces [74].

3.2.3 B-Spline surface

B-Spline surfaces play an important role in geometric modelling. Mathematically, a B-Spline surface is a product of control vertices and B-Spline basis function that can be written as follows

$$\underline{X}(u, w) = \sum_{i=0}^m \sum_{j=0}^n p_{i,j} N_{i,m}(u) N_{j,n}(w) . \quad (3.5)$$

It is assumed that one knot sequence is in the u -direction and one in the v -direction. The term $N_{i,m}(u)$ is a B-Spline basis functions that is given by

$$N_{i,m}(u) = \frac{u - u_i}{u_{i+m} - u_i} N_{i,m-1}(u) + \frac{u_{i+m+1} - u}{u_{i+m+1} - u_{i+1}} N_{i+m,m-1}(u) \quad (3.6)$$

where

$$N_{i,0}(u) = \begin{cases} 1 & ; \quad u_i \leq u \leq u_{i+1} \\ 0 & ; \quad otherwise \end{cases} \quad (3.7)$$

The term $N_{i,m}(u)$ in Equation (3.6) is called a normalized B-Spline of degree m . The interval $[u_i, u_{i+1}]$ is called the i -th knot span and the knot vector is defined as $\underline{U} = \{u_0, u_1, \dots, u_m\}$.

In geometric design, there are two types of knots which are used; uniform (with equally spaced knots) and non-uniform. In addition, if the first and the last knots are repeated $(m+1)$ times then the knot vector is non-uniform and non-periodic. Non-uniform knot vectors provide great flexibility for

various design applications [74] and give the advantage of manipulating only a small part of the surface accordingly.

The main difference between B-Spline and Bezier surfaces is that in the case of Bezier surfaces, the control polygon uniquely defines the surface, whereas B-Spline surface requires the knot vector in addition to the control polygon. This property of B-Spline is called knot insertion. By adding extra knots to the control polygon, the approximation of the generated surface is greatly improved. Therefore, by adding more knots, one can easily limit the area affected by a control point modification otherwise known as local modification of the surface [73].

3.2.4 NURBS surface

The most popular surface in geometric modelling is NURBS surface. It offers a common mathematical form for representing and designing free form surfaces. NURBS curves represent a natural generalization of B-Spline and Bezier curves. Bivariate NURBS surfaces are the proper generalization of tensor product B-Spline and Bezier surfaces [74].

Extending the formulation of B-Spline in (u, v) space, the formulation of NURBS is given by:

$$\underline{X}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m \omega_{i,j} \underline{p}_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^n \sum_{j=0}^m \omega_{i,j} N_{i,p}(u) N_{j,q}(v)} \quad (3.8)$$

where

- $\omega_{i,j}$ denotes the weight of every control and $\omega_i > 0$.

- $\underline{p}_{i,j}$ forms a control polygon. The control polygon is defined over the knot vectors as follows:

$$\underline{U} = \{u_1, u_2, \dots, u_{n+p+1}\} \quad (3.9)$$

$$\underline{V} = \{v_1, v_2, \dots, v_{m+q+1}\} \quad (3.10)$$

- $N_{i,p}$ and $N_{j,q}$ are normalized B-Splines of degree p and q respectively in the u and v directions. Normally, $u \in [0,1]$ and the surface is undefined outside the domain.

NURBS surfaces are useful because they are invariant under rotation, scaling, translation and perspective transformation of the control points [2]. The most useful properties of NURBS surface is localness where one can manipulate the NURBS surface by repositioning the control points or altering the value of surface weight [75]. This characteristic gives NURBS surface a flexible way to design a large variety of shapes. However, the NURBS surface needs a lot of control points to define a complex surface such as a human face, thus it requires a bigger storage capacity. This disadvantage is not particular to NURBS. Other polynomial based surface generations also exhibit the same problem.

3.3 PDE surfaces

Given that conventional methods need huge amounts of storage to define complex geometry, Bloor and Wilson have introduced a PDE method, which is based on the use of partial differential equations with suitable chosen

boundary conditions to generate any given object. The PDE method is able to parameterize complex surfaces using a small set of design parameters compared to conventional methods, which require hundreds of control points.

The PDE-based method was primarily introduced in surface blending [7] and followed by free-form surfaces [8]. In recent years, the PDE-based method has widened its application to graphics and modelling including; modelling of wing geometries [76], interactive design [77], vase design [78], aircraft geometry [6], pharmaceutical tablets [79], among others.

3.3.1 PDE surface patches

The PDE method produces parametric surface patches, $\underline{X}(u, v)$, defined by two parameters, u and v , in finite region $\Omega \subset R^2$ with a specified boundary, $\partial\Omega$. Here $\underline{X}(u, v)$ is regarded as a mapping point in Ω to a point in Cartesian space such that $R^2(\Omega) \rightarrow E^3$. In order to find $\underline{X}(u, v)$, it is required to find a solution to the partial differential equation given by:

$$D_{u,v}^m(\underline{X}) = \underline{F}(u, v) \quad (3.11)$$

where

- $D_{u,v}^m$ is a partial differential operator of order m in independent variables u and v .
- \underline{F} is a vector-valued function of u and v .

The function \underline{F} is included for completeness and usually the value is zero [80].

Usually, the choice of the operator $D_{u,v}^m$ is elliptic as elliptic PDE can produce

smooth surfaces for boundary-value problems. The choice of order m depends on the level of surface control and continuity required for the shape of the surface. If a second order PDE is used, there are no tangential boundary conditions therefore the designer loses the ability for controlling the shape of the surface. In general, the more flexible and powerful surface is archived by higher order PDE [81]. For this research, the fourth-order PDE is used, where four boundary conditions are needed for each $\underline{X}(u, v)$ patch.

The general form of a fourth-order elliptic PDE over 2D parametric domain is given by:

$$\left(\frac{\partial^2}{\partial u^2} + a^2 \frac{\partial^2}{\partial v^2} \right)^2 \underline{X}(u, v) = 0. \quad (3.12)$$

where

- $\underline{X}(u, v)$ is a PDE surface in a domain Ω . The boundary conditions are imposed on the functions $\underline{X}(u, v)$ and its normal derivatives ($\underline{X}_{,u}$ and $\underline{X}_{,v}$) at the edge of the surface patch.
- a is a smoothing parameter and $a \neq 0$. This special design parameter controls the relative smoothing of the surface in the u and v directions.

Note that Equation (3.12) is known as Biharmonic equation if $a=1$. The Biharmonic equation can model some phenomena related to solid and fluid mechanics. It is also useful in engineering analysis that relates with stress/strain analysis problems. Hence, there are different methods that have been developed for solving Equation (3.12) ranging from analytical solution techniques to sophisticated numerical methods [82].

3.3.2 Solution to the PDE method

As mentioned above, there are various techniques that can be used to find the solution of Equation (3.12). The work described in this thesis is based on spectral approximation [83] where the approximation solution of the PDE surfaces is calculated using Fourier analysis and gives the solution in closed form including a remainder term.

Equation (3.12) is solved over the region Ω of a parameter plane, subject to a set of boundary conditions on the solution \underline{X} , \underline{X}_u and \underline{X}_v around the boundaries of specified region, where $0 \leq u \leq 1$ and $0 \leq v \leq 2\pi$. Boundary conditions on \underline{X} are imposed for defining the shape of the trim lines in physical space in terms of u and v , and the first derivative continuity, \underline{X}_u and \underline{X}_v , is required for specifying the surface normal ($\underline{X}_u \times \underline{X}_v$) along the trim lines.

The boundary conditions that are used to solve Equation (3.12) are specified as follows:

$$\underline{X}(0, v) = \underline{P}_0(v) \quad (3.13)$$

$$\underline{X}(1, v) = \underline{P}_1(v) \quad (3.14)$$

$$\underline{X}_u(0, v) = \underline{d}_0(v) \quad (3.15)$$

$$\underline{X}_u(1, v) = \underline{d}_1(v) \quad (3.16)$$

Note that $\underline{P}_0(v)$ defines the edge of the surface patch with derivative conditions $\underline{d}_0(v)$ at $u=0$ and $\underline{P}_1(v)$ defines the edge of the surface patch with derivative conditions $\underline{d}_1(v)$ at $u=1$ as shown in Figure 3-2.

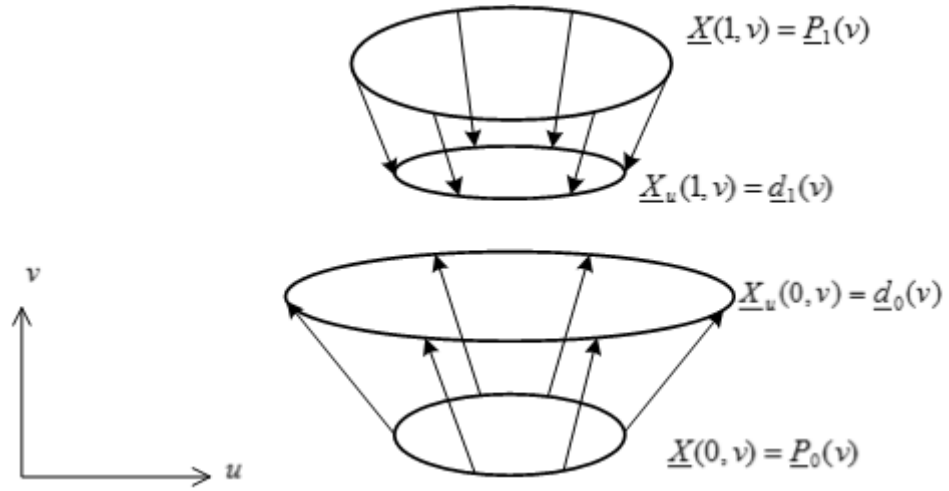


Figure 3-2 Illustration of position vectors and derivative vectors

When the functions $\underline{P}_0(v)$, $\underline{P}_1(v)$, $\underline{d}_0(v)$ and $\underline{d}_1(v)$ are specified, the solution of Equation (3.12) can be found using the separation of variable method, which can be written as:

$$\underline{X}(u, v) = \underline{A}_0(u) + \sum_{n=1}^{\infty} [\underline{A}_n(u) \cos(nv) + \underline{B}_n(u) \sin(nv)] \quad (3.17)$$

where

$$\underline{A}_0 = \underline{a}_{00} + \underline{a}_{01}u + \underline{a}_{02}u^2 + \underline{a}_{03}u^3 \quad (3.18)$$

$$\underline{A}_n = (\underline{a}_{n1} + \underline{a}_{n3}u)e^{anu} + (\underline{a}_{n2} + \underline{a}_{n4}u)e^{-anu} \quad (3.19)$$

$$\underline{B}_n = (\underline{b}_{n1} + \underline{b}_{n3}u)e^{anu} + (\underline{b}_{n2} + \underline{b}_{n4}u)e^{-anu} \quad (3.20)$$

and \underline{a}_{n1} , \underline{a}_{n2} , \underline{a}_{n3} , \underline{a}_{n4} , \underline{b}_{n1} , \underline{b}_{n2} , \underline{b}_{n3} and \underline{b}_{n4} are constant vectors which can be found by imposing the boundary conditions in Equation (3.13 – 3.16). Then the boundary conditions need to be expressed in terms of Fourier series in order to identify all the unknown Fourier coefficients. If all the functions $\underline{P}_0(v)$, $\underline{P}_1(v)$,

$\underline{d}_0(v)$ and $\underline{d}_1(v)$ can be expressed in terms of finite Fourier series, the solution to Equation (3.17) is also finite.

However, when the functions at boundaries cannot be expressed in terms of finite Fourier mode, therefore, the solution is infinite. In this case, an approximate technique is often used and is based on the sum of the first few Fourier modes and a remainder term as follows:

$$\underline{X}(u, v) = \underline{A}_0(u) + \sum_{n=1}^N [\underline{A}_n(u) \cos(nv) + \underline{B}_n(u) \sin(nv)] + \underline{R}(u, v) \quad (3.21)$$

where N is finite (usually $N \leq 5$) and $\underline{R}(u, v)$ is a remainder function. The coefficient functions $\underline{A}_n(u)$ and $\underline{B}_n(u)$ are given by Equation (3.19) and (3.20), which can be determined from the amplitudes of the first N Fourier modes in boundary conditions. For example, a Fourier analysis of the boundary conditions allow \underline{P}_0 to be written as:

$$\underline{P}_0(v) = \underline{a}_0 + \sum_{n=1}^N [\underline{a}_n \cos(nv) + \underline{b}_n \sin(nv)]. \quad (3.22)$$

The remainder function $\underline{R}(u, v)$ represents the contribution of high frequency mode where it can be written as

$$\underline{R}(u, v) = [\underline{r}_1(v) + \underline{r}_2(v)u]e^{ou} + [\underline{r}_3(v) + \underline{r}_4(v)u]e^{-ou} \quad (3.23)$$

and the $\underline{r}_1, \underline{r}_2, \underline{r}_3$ and \underline{r}_4 are obtained from the Fourier analysis by considering the difference between the original boundary conditions and the boundary conditions satisfied by the function

$$\underline{F}(u, v) = \underline{A}_0(u) + \sum_{n=1}^N [\underline{A}_n(u) \cos(nv) + \underline{B}_n(u) \sin(nv)] \quad (3.24)$$

The solution outlined above is considerably faster than looking for the numerical solution (either finite element or finite difference) to Equation (3.12). It is important to highlight that the solution using approximation above also guarantees that the chosen boundary conditions are satisfied by the function $\underline{F}(u, v)$ [84].

3.3.3 Boundary conditions modification

The solution given in the previous section defines the position and derivative boundary conditions at the edge of $u = 0$ and $u = 1$. The solution from chosen boundary conditions produces a parametric surface patch. The generated parametric surface can be controlled by:

- the position vectors, which are defined by the position boundary conditions ($\underline{P}_0(v)$ and $\underline{P}_1(v)$).
- the derivative vectors, which are defined by the derivative boundary conditions ($\underline{d}_0(v)$ and $\underline{d}_1(v)$).

In a case where no derivative vectors are required for manipulating the shape of the surface, one can change the way to define the boundary conditions. It would be useful for designing a surface patch, which all boundary conditions pass through on the surface. This small modification has more practical advantage as the surface can be manipulate in a more intuitive way.

This slight adjustment can be done by defining the periodic boundary conditions using four positional conditions in parametric region, $0 \leq u \leq 1$ and $0 \leq v \leq 2\pi$. Thus, the position boundary conditions, are given by:

$$\underline{X}(0, v) = \underline{P}_1(v) \tag{3.25}$$

$$\underline{X}_a(a, v) = \underline{P}_a(v) \quad (3.26)$$

$$\underline{X}_b(b, v) = \underline{P}_b(v) \quad (3.27)$$

$$\underline{X}(1, v) = \underline{P}_2(v) \quad (3.28)$$

where $a, b \subseteq [0, 1]$. It is worth pointing out that this definition of boundary condition expressions does not affect the core of PDE formulation [85]. Without loss of generality, these positional boundary curves can be used for generating PDE surface patch using the method as described in Section 3.3.2. For example, a cylinder can be generated using four boundary curves for generating a PDE surface patch as shown in Figure 3-3(a) and the corresponding generated cylinder shown in Figure 3-3(b).

In addition, the PDE generated cylinder can be added to a texture for more a sophisticated surface as illustrated in Figure 3-3(c). Furthermore, the shape of the cylinder can be controlled by varying the boundary curves. Figure 3-4 shows that the different boundary curves will change the shape of the surface.

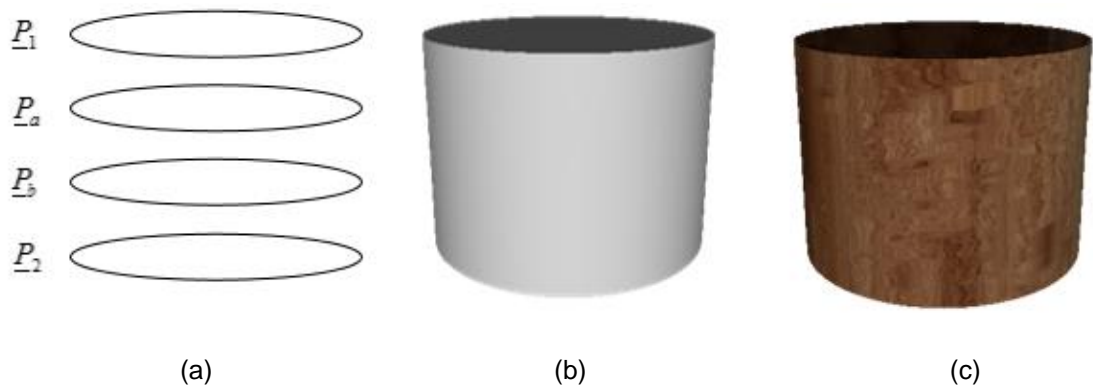


Figure 3-3 A cylinder generated by four boundary curves. (a) boundary curves (b) PDE generated cylinder (c) PDE cylinder with texture

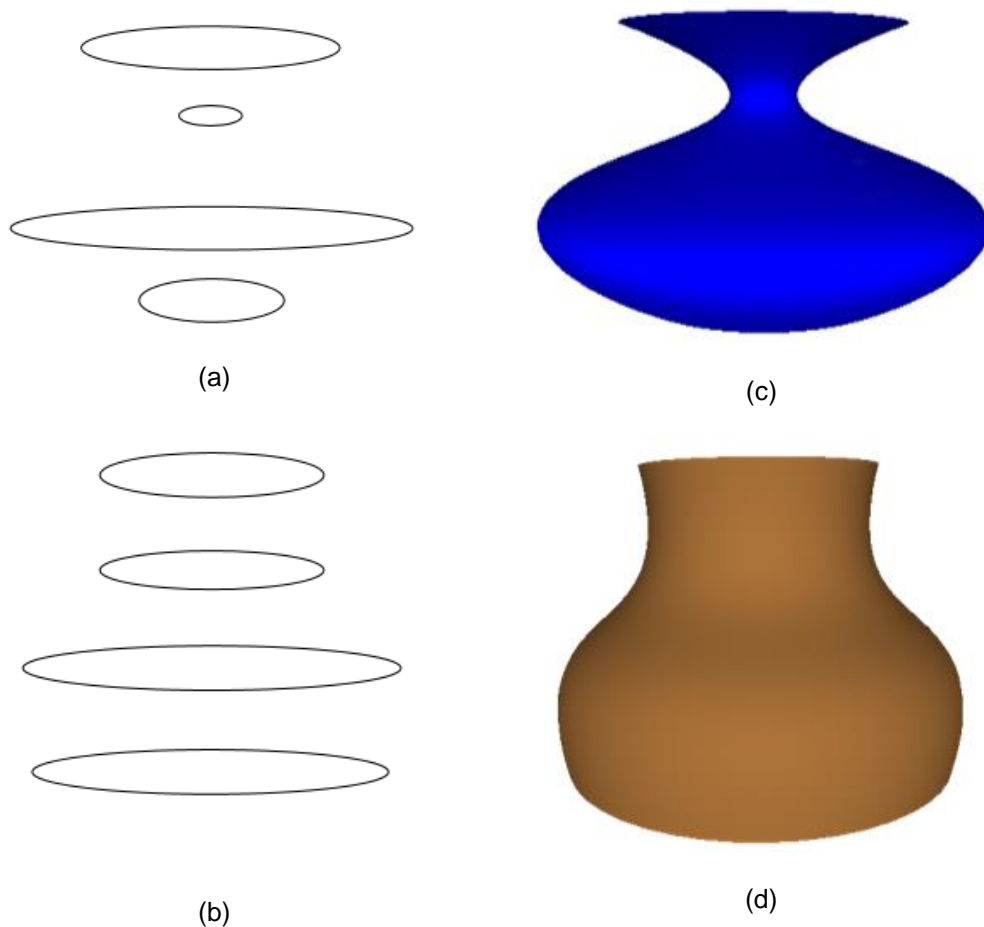


Figure 3-4 Shape manipulation by changing the positional curves: (a) and (b) are boundary curves and (c) and (d) are corresponding PDE surface patch respectively.

In practical application, more than one surface patch is needed for modelling complex objects. For this purpose, an adjacent PDE patch shares a common boundary curve to ensure smooth positional continuity across all PDE patches. For example, the object shown in Figure 3-5 is generated using six different PDE patches. These six different patches use 16 boundary curves as shown in Figure 3-6 whereas using four consecutive boundary curves accordingly means the adjacent PDE patches share a common boundary curve (i.e. 4, 7, 10, 13 and 16).

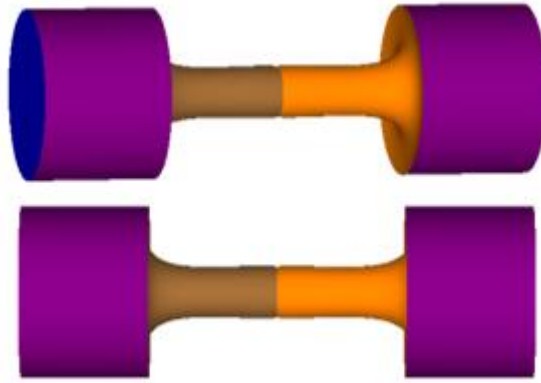


Figure 3-5 An example of complex object of PDE surface with two different views.

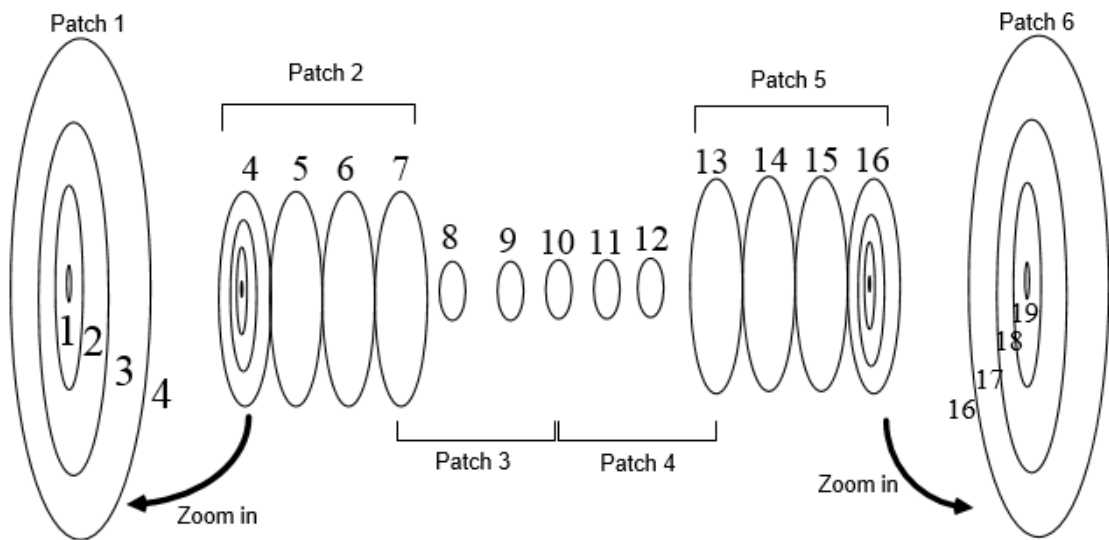


Figure 3-6 Boundary curves for surface generation of six patches object

For a more complex object, more boundary curves are required to make sure the generated surface has a smooth surface.

3.3.4 Advantages of the PDE method

Compared to conventional surface generation in Section 3.2, the PDE surface generation offers the following advantages:

- A PDE surface is a powerful surface generation technique, as it requires a small number of parameters compared to the conventional method. Hence, a PDE surface is easier to manipulate for getting desired shape [86].
- Surface generation technique is based on PDE that automatically guarantees some degree of intrinsic smoothness. This is very useful when blending two or more surfaces together [82].
- A PDE surface is more intuitive as the designers do not need to have mathematical knowledge in order to manipulate the shape given that the boundary conditions are explicitly represented in the solution [84].
- A PDE-based modelling method is physically based. For example, a fourth order PDE can be derived from the theory of bending thin elastic plates. Thus the coefficients of such a PDE are closely related to the physical properties of the surface which it represents [87].
- The PDE modelling methods provide alternative means to the control-points based methods such as Bezier or splines. The surface shape can be manipulated by changing only a few vector-valued shape parameters or by altering the position, the tangent and the curvature of the boundary conditions of the PDE [87].

Considering the advantages offered by the PDE method, this research is taking the full potential of a PDE surface for surface generation of the human faces in a fast and realistic way. As mention in [61], facial modelling that is based on anatomy and physics is more powerful compared to a conventional modelling method. The PDE-based modelling itself facilitates the physical properties of the generic face. Thus, in this thesis, an additional feature is

added to the PDE-based model by integrating the muscle movement on the face model, which is known as action units for animation purposes. Further discussion on the implementation of action units on the PDE face model will be explained in Chapter 5.

3.4 PDE-based facial model

In recent years, there has been considerable interest in modelling and animating the human face using the PDE facial model, or known as PDE-based face. The work in [46] demonstrates a method for the construction of 3D geometry of human faces using the PDE method. The proposed method is an efficient parametric representation of a face model and the generated PDE faces is very close to its original data set. This pioneer work on PDE faces uses 28 cross-section curves across the face, where nine PDE patches are composed to a complete geometry of a PDE face as shown in Figure 3-7.

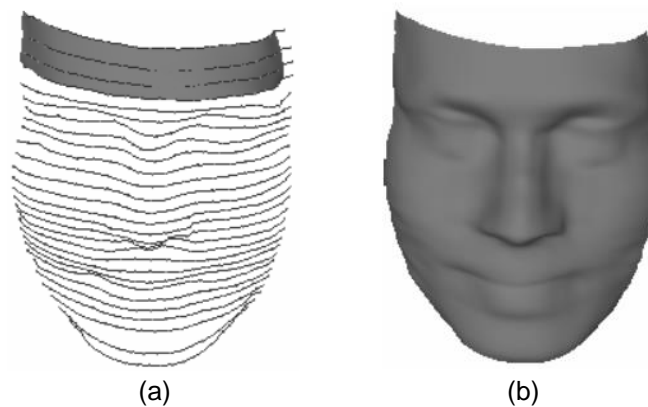


Figure 3-7 PDE face model taken from [46]. (a) represent 28 boundary curves (b) the resulting PDE face

The textured 3D PDE face model presented in [85] introduced a uv -mesh texture mapping method, which provides a solution to the face modelling problem especially to model representation and texture. The generated PDE

face can be textured and its gives advantages over the existing texturing approaches as the specific face can be textured without the need for the facial expression in the texture map to match exactly.

The research on PDE faces was expanded for generating a PDE face with expression by manipulating the original boundary curves as presented in [88]. It presented a technique for parameterizing different facial expressions by adjusting specific regions of the boundary curves using the PDE method. After the boundary curves and the points in these curves that are affected by each expression were identified, a suitable mathematical transformation can be applied to get the desired result. In their works, they successfully represented some facial expressions such as smiling, frowning, mouth opening, and eyebrow raising.

A PDE face model also gives some advantages in facial animation as the work presented in [89]. Direct parameterization in facial animation is achieved by manipulating the position of boundary curves. It adopts low level MPEG-4 Facial Animation Parameters (FAPS) to three facial features. They are eyebrows, eyes and mouth. This successfully developed a FAP-s driven animation scheme for four facial expressions; joy, sadness, anger and fear. However, this animation cannot implement the animation involved in the opening and closing of the mouth.

Based on the previous facial model generated using the PDE method, the method concentrates on generating generic facial models and uses 28 boundary curves to define a PDE face. In this research, the chosen boundary curves are based on FACS where the boundary curves are placed in areas

that correspond to the action units of FACS. The detail on the process to embed the facial action in PDE-based face will be discussed further in Chapter 5.

3.5 Animation of facial expressions based on FACS using conventional surface generation methods

There is some research based on FACS that has been done for modelling and animating of facial expressions using conventional methods such as B-Spline and NURBS. The control points of spline methods are positioned according to facial anatomy and spline curves are geometrically associated with polygon mesh, which represents the face.

The work in [21] introduces a technique known as the canonical model when fitting the scan data for the animation of the facial expression. They had chosen four action units that are sufficient to generate simple facial expressions; these are AU1, AU2, AU4 and AU12. The control points that are associated with action units are positioned correctly in the region where the respective action units have an influence. For this purpose, it manually determines four characteristic points after the surface fitting of each face, which is not practical for complex facial animation.

Some studies have been done in modelling and animating facial expressions using NURBS such as the work described in [75] and [90]. They position the control polygon of NURBS based on facial anatomy and primarily decide by location where the simulated muscles are attached to facial bones. A fuzzy set is then used to assign vertices of the facial mesh as a membership

function based on the different areas. The happiness, anger and sadness expressions are achieved by simulating the weight functions while repositioning the control points simulates fear, surprise and disgust. However, finding the correct position to place the control polygon on a mesh model is not an easy task.

There is also a realistic facial expression animation developed using a NURBS-based vector muscle system [63]. It presents a system to simulate linear muscle, sphincter muscles and the fatty tissue. By modifying the weight of different control points, this method can simulate a particular facial expression. Twenty-two muscles are implemented where the action units are based on the facial anatomy. As each action unit represents a single muscle, facial expressions are created using different combinations of action unit. The NURBS-based vector muscles are extended by [91] for simulating and animating the talking mouth. From a video of a talking person, the parameters of the lip shape were extracted and used for animating the talking mouth. Although this approach is able to produce realistic results, capturing a real person's lip movement can be a time-consuming and tedious task.

Based on the literature discussed above, there is some weakness that needs to be improved in modelling and animating the facial expression based on FACS. Previous models use the polygon based method for generic face models and manipulated the face for desired expressions using control points. This type of approach is very time-consuming and needs huge storage for storing the facial data and expressions. Furthermore, there is still an open question on how to represent the facial expression in appropriate parameters of parameterised face models [38].

Therefore, this research aims to address the problems discussed above that arise by conventional surface generation methods. In order to reduce the storage capacity, this work adopted the PDE method for generating parametric face models as it only stores small parameters of facial data (further discussion in Chapter 4). Moreover, this work also describes the way for efficiently storing the parameters for defining action units, where the combination of these action units will form an expression on a generic PDE face. This approach of defining parameters gives an intuitive way for manipulating the PDE face in a fast and realistic way (more discussion in Chapter 5).

3.6 Summary

In this chapter, the discussion begins with surface generation techniques that are popular in geometric modelling including Bezier surface, B-Spline surface and NURBS surface. These surface generation techniques are referred to as conventional methods for easy reference. This conventional modelling method is based on a control polygon for manipulating the generic object. As the human face has a complex surface, it is difficult and time-consuming to model and animate facial expressions since they consist of a large number of control points.

On the other hand, this research uses another parametric surface representation for modelling a face known as the PDE method. The PDE-based face is generated as a solution to elliptical PDE where face generation is treated as a boundary value problem. This boundary value problem is imposed around the edges of the surface patches known as boundary

conditions. Boundary conditions are important for determining the shape of surface. Changing the boundary conditions will change the shape of the surface. In the context of this work, changing the parameter in boundary curves will create human faces with different action units and expressions.

As mentioned in Section 3.4, the approach adopted in this research is different from the approach used by others for generating a PDE-based face. In particular, this work used FACS for representing the facial movements of the face, where facial parameters corresponding to the actions units are extracted in terms of Fourier coefficients. Using such an approach, the deformations of facial models for making expressions is straightforward that is by adding the coefficients of action units to the generic model. Thus, the approach in this work gives advantages in terms of storage capacity and reduces time for modelling and animating facial expressions.

4. PDE-based Facial Modelling for Animation

4.1 Introduction

This chapter offers detail on the methodology that has been used in this research. In the next section, the database that is used in this thesis is introduced. Then the detailed explanation on how a 3D human face is generated by utilising the PDE method is discussed including how to texture the generic PDE face for a more realistic look. Later, results as well as the accuracy of the methodology is presented. Finally, a summary of the methodology is given in the last section.

4.2 Database of 3D faces

For the purpose of this research, a database of dynamic 3D FACS [92] is utilized. The database was developed at the Department of Computer Science at the University of Bath. This database contains 10 different people, consisting of four males and six females. All subjects in the experiment have been performed between 19 and 97 different action units; both individually and in combination. In total, there are a total of 519 action unit sequences. The peak expression frame of each sequence has been manually coded by certified FACS experts.

The 3D face data in the database are available in OBJ file format for the 3D surface meshes and consists of the two pod-half face images joined together in BMP file format, for each frame of the dynamic data. An OBJ file is

a powerful representation of the surface of a 3D object in a simple data-format [93] and supported by 3D modelling software such as Maya, 3DS Max, Blender etc. The OBJ data file taken from the database consists of:

- the position of each vertex, denoted by “v”
- the vertex normal, denoted by “vn”
- the UV position of each texture coordinate vertex, denoted by “vt”
- the faces that make each polygon defined as a list of vertices, denoted by “f”.

If an original OBJ file taken from the database is rendered using Meshlab, the rendered face is shown in Figure 4-1, which is rendered as untextured face.



Figure 4-1 Example of 3D image taken from database

This is due to the fact that an OBJ file only represents the geometric information of the face without colour or texture information. Any information on texture is stored independently in the Material Template Library (MTL) files [94]. Thus, an OBJ file need information about texture which is stored separately in a MTL file. The MTL file contains definitions of various material types used for realistic rendering together with texture images (PNG, BMP,

JPEG etc). In order to put a texture on the face from the database, the face model is rendered by specifying the OBJ file, MTL file and texture image accordingly. The example of texture image (BMP file) of Eva's neutral face taken from the database is illustrated in Figure 4-2(a) and the corresponding face with texture is given in Figure 4-2(b).

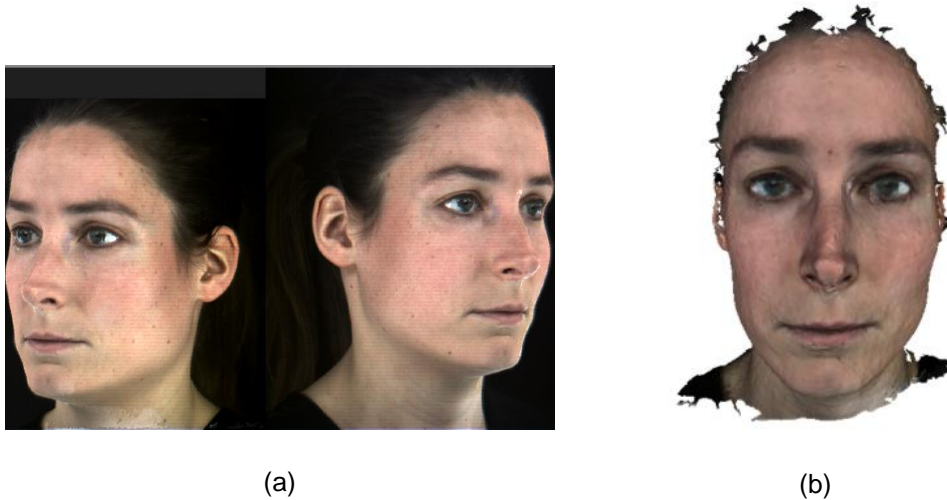


Figure 4-2 Example of face model taken from database. (a) two pod half-face image (b) face model with texture

It is worth highlighting that a database of this nature has been highly useful in this research to develop and verify the works that have been done. In particular, this database is utilized for extracting the boundary curves in order to realistically describe the face and develop the PDE based action units.

4.3 Boundary curves extraction

As mentioned above, there are ten people demonstrating the action unit in either single or combination. From the ten subjects, Eva's dataset has been used throughout this thesis as she performed the most action units compared

to the other subjects. Some of the single action unit that she performed are shown in Figure 4-3 that is extracted from [95].

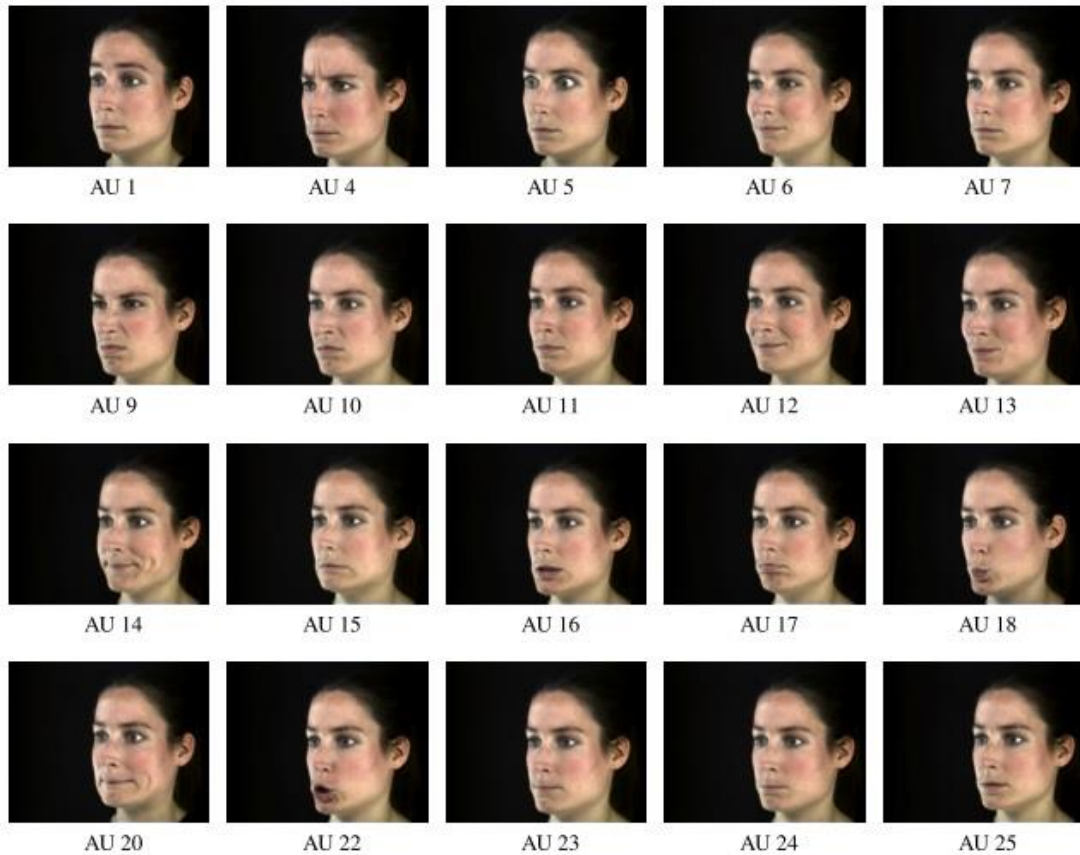


Figure 4-3 Images of Eva in 3D FACS database

From the sequences of 3D mesh dataset by Eva, the neutral mesh face of Eva was chosen and textured as shown in Figure 4-2(b). The chosen neutral configuration acts as a template of the PDE-based face of Eva. Then, the template is used for extracting boundary curves for a given action unit that will be discussed in detail in Section 4.5.

4.3.1 Pre-processing

The obtained 3D mesh face data from the database would require pre-processing prior to the boundary curves extraction procedure. A raw surface

mesh may have a different alignment and orientation. This pre-processing step is important to have a standardized and normalized mesh. The procedure is performed using Autodesk MAYA and its scripting language, MEL script.

Initially, the 3D mesh data is scaled to a normalized coordinate system and then rotated around to put the face in the correct orientation. Here correct orientation means that the facial mesh should face the xy -Cartesian coordinate plane. When the tip nose is located, the mesh is translated so the nose tip is located at origin. Next, the nose bridge and the centre of the bottom are located manually. Based on the nose tip, nose bridge and the nose bottom, the facial mesh can be rotated by aligning the mesh parallel to yz -Cartesian coordinate plane. The standardized and normalized face mesh is shown in Figure 4-4.

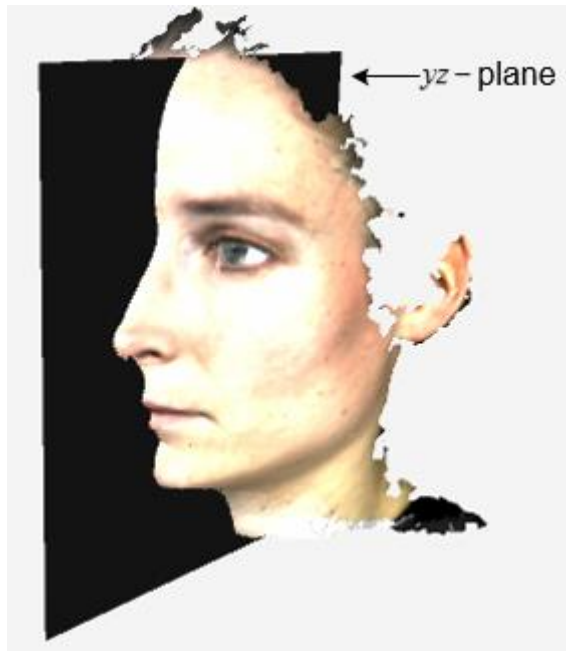


Figure 4-4 Standardized and normalized face mesh

This task is performed automatically within Maya using a MEL script specifically written for this purpose.

4.3.2 Curve extraction

The most important task when shape modelling using the PDE formulation is to make available a set of suitable boundary curves for representing the shape of the object. For the purpose of human face representation, it is paramount to make sure that all extracted curves accurately represent important features of the face. These facial features include mouth, eyebrows and eyes as the appearance of these facial features is changing when the facial muscles contract [96].

Previous PDE-based models by E. Elyan [46], and Y. Sheng et al ([88], [89]) have used 28 boundary curves across the face to accurately represent the face. This formulation of boundary curves produced nine different PDE surface patches where each of the surface patches is generated by four consecutive boundary curves. Note that all nine PDE patches share common boundary curves to guarantee positional continuity between all the generated patches as discussed in detail in section 3.3.3.

In this thesis, a new approach for extracting the curve has been formulated by referring to the previous work on the PDE face representation. This new curve extraction procedure is necessary due to the fact that the goal in this research is not just to represent the face, but also underlying muscle structure of the face for representing facial expressions suitable for animation. Thus, the boundary curves are based on the facial anatomy representing the geometric aspect of facial muscle distribution and the movement of the muscles of the face.

In order to position points to form boundary curves, a set of characteristic points (vertices) that correspond to feature points corresponding to the muscles is determined as defined by FACS. For this purpose, the features points selection are based on the 19-feature points that is adapted from [96].

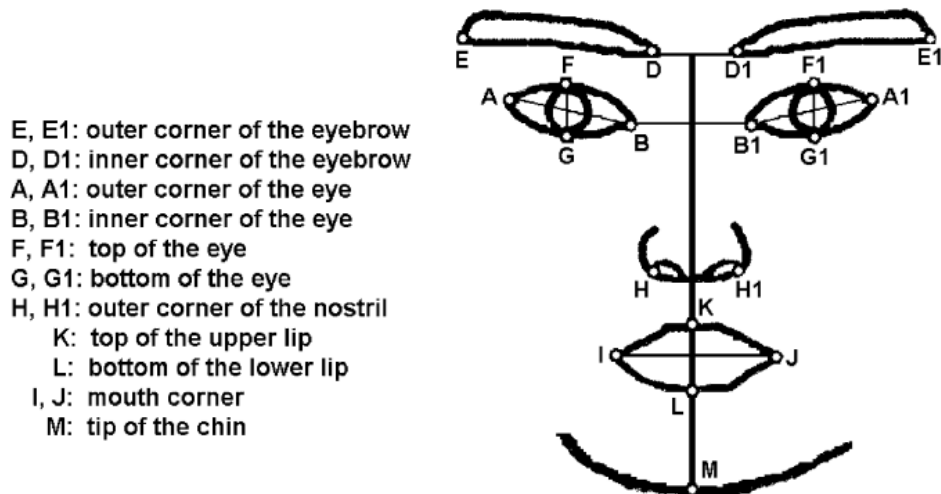


Figure 4-5 Feature points extracted from [96]

These feature points are important for extracting action units and aligning procedure that is described further in Chapter 5.

In order to locate feature points, a simple experiment is carried out using trial-and-error method. At the beginning of the experiment, the selected points on the face are chosen randomly. These points are a representation of 19 boundary curves. Then, all the points are exported to 19 different OBJ files. Note that for each boundary curve, 37 points are stored in each OBJ files. C# programming language is used to run and process these files (this step will be discussed in detail in Section 4.4). The visualization result of PDE generic face based on randomly chosen points (37 points) and 19 boundary curves is shown in Figure 4-6(a).

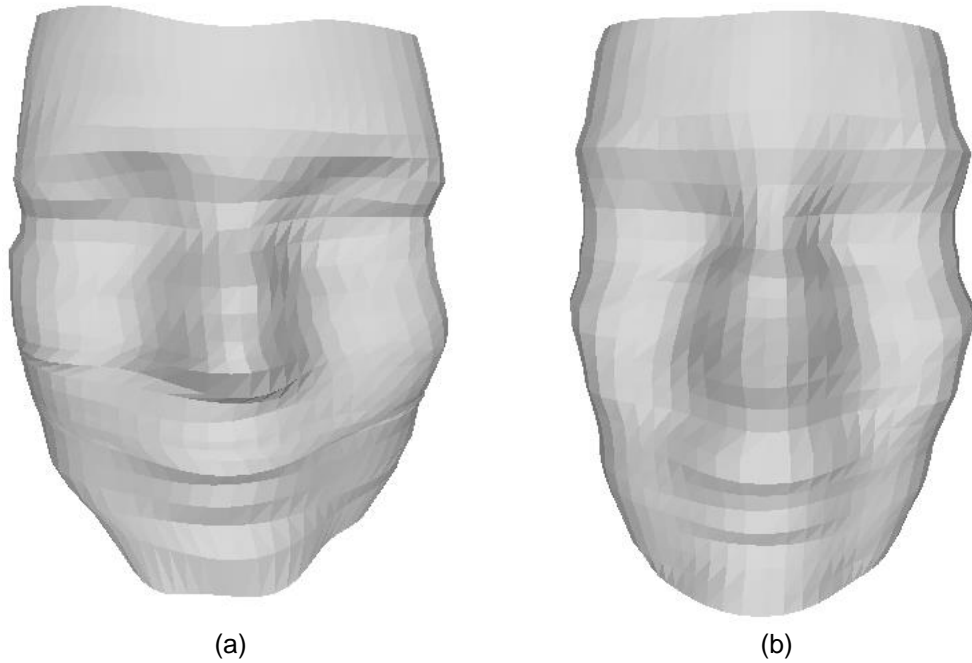


Figure 4-6 PDE face generated using 19 boundary curves, (a) randomly chosen points, and (b) uniformly chosen points

It is clear that using randomly chosen points with 19 boundary curves is not sufficient enough to represent correct facial features. Due to this issue, the experiment is continued by refining the distributed points into a uniform way, but maintaining the number of boundary curves. The result is depicted in Figure 4-6(b). However, the problem occur when modelling the AU (will be discussed in detail in Chapter 5). It turned out to be a bit complicated to represent certain AU using 19 boundary curves. Therefore, the experiment is repeated by refining the boundary curves iteratively with 3 boundary curves are added each time. The experiment stopped when the author found out that a total of 31 boundary curves is sufficient to accurately represent a face and facial muscle movement as shown in Figure 4-7. The boundary curves extracted for a typical face used in the rest of this thesis are shown in Figure 4-7(a) and the position of boundary curves located throughout the face is shown in Figure 4-7(b).

It is worth mentioning that all 31 extracted boundary curves must be in closed form to ensure that the correct PDE face is produced. The generating boundary curves shown in Figure 4-7(a) are not in closed form as the mesh image from the database only contains the image of the front face. Thus, the MEL script is used to close the curves by reading the OBJ file and adding imaginary points by reflecting the coordinate to the opposite side.

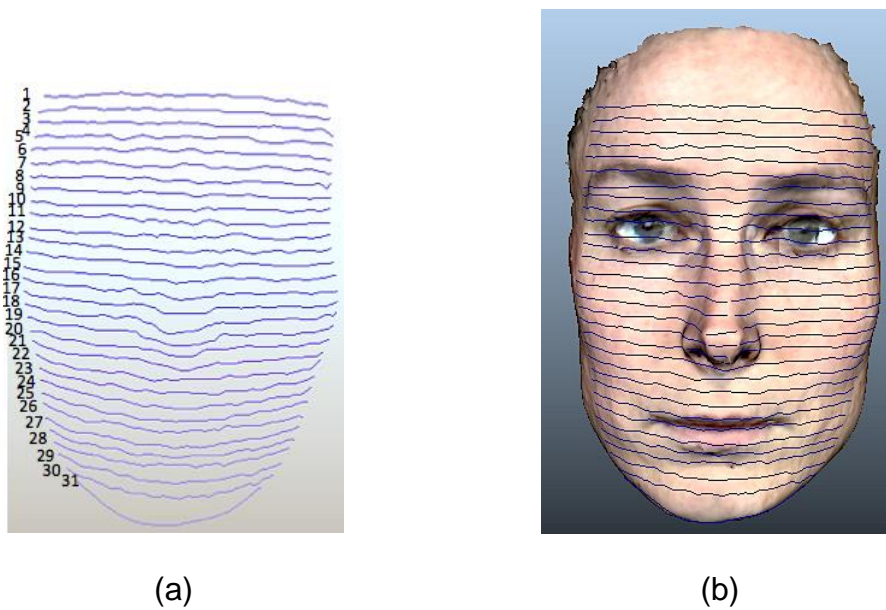


Figure 4-7 A neutral face and position of extracted curves, (a) generating boundary curves, (b) a face from the FACS database

4.4 PDE-based face generation

Once all the necessary curves have been extracted, these curves are saved in external OBJ files. It is important to point out that all 31 boundary curves extracted in the previous section are represented in terms of their corresponding Fourier series as discussed in the next sub section. Then in section 4.4.2, a method for texturing the generic PDE face is discussed.

4.4.1 Fourier coefficients of facial curves

As mentioned previously, the points of the curves are represented in terms of their corresponding Fourier series. The approximation of the boundary curves can be written in terms of finite Fourier series as:

$$f_i(v) = \underline{a}_{ou} + \sum_{n=1}^N [\underline{A}_{ni} \cos(nv) + \underline{B}_{ni} \sin(nv)] \quad (4.1)$$

where i is the number of boundary curves and N is the mode of Fourier series. Note that $N \geq 5$.

This procedure of storing facial information in the form of Fourier coefficients corresponding to the curves representing the face gives advantages in terms of storage. From Equation (4.1), a facial information of all boundary curves can be written in this form:

$$M_f = \begin{bmatrix} a_{0,1} & a_{1,1} & \cdots & a_{N,1} & b_{1,1} & \cdots & b_{N,1} \\ a_{0,2} & a_{1,1} & \cdots & a_{N,2} & b_{1,2} & \cdots & b_{N,2} \\ a_{0,3} & a_{1,1} & \cdots & a_{N,3} & b_{1,3} & \cdots & b_{N,3} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{0,i-2} & a_{1,i-2} & \cdots & a_{N,i-2} & b_{1,i-2} & \cdots & b_{N,i-2} \\ a_{0,i-1} & a_{1,i-1} & \cdots & a_{N,i-1} & b_{1,i-1} & \cdots & b_{N,i-1} \\ a_{0,i} & a_{1,i} & \cdots & a_{N,i} & b_{1,i} & \cdots & b_{N,i} \end{bmatrix} \quad (4.2)$$

As mentioned previously, this work used 31 boundary curves ($i = 1, 2, 3, \dots, 31$) for representing a complete human face using five modes of Fourier series. An example of the coefficients that are stored in given form when $N=5$ is shown in **Appendix A**. With formulation in Equation (4.2), a full PDE-based face is generated by four sets of consecutive boundary curves resulting in ten continuous facial surface patches as shown in Figure 4-8(a), where different

colours represent different PDE patches. The corresponding PDE face in wireframe mode is shown in Figure 4-8(b).

Furthermore, the resolution of the PDE face is flexible due to its parametric representation [47]. Using the formulation given in Equation (4.2), a reconstruction of a face model in different resolution is very straightforward. In a case where very smooth surface is needed, a higher resolution can be used by changing the uv – mesh resolution. In this thesis, the face model was generated using 31–by–35 resolution.

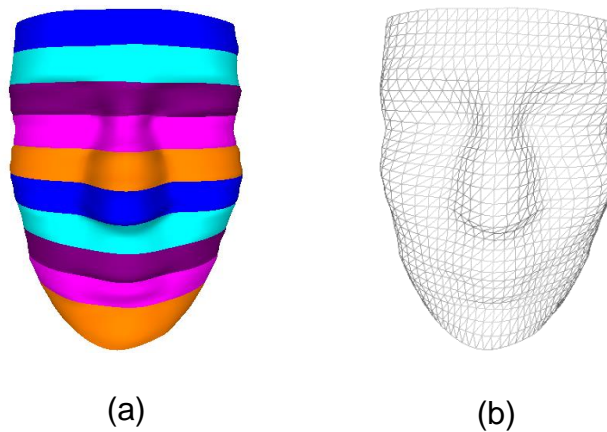


Figure 4-8 PDE generated neutral face, (a) ten patches of continuous PDE surface (b) wireframe of full PDE face

The PDE method that has been discussed in Section 3.3 is implemented using C# by reading the boundary curves that define the shape of the face and then produces the solution for each set of boundary curves. It is noted that the geometry of the generic PDE face is saved in OBJ file format. This format of storing the geometry of a face gives an advantage as the process of texturing the generic PDE face is easier. The texturing process of a PDE face is covered in the next sub section.

4.4.2 Textured face models

For a more realistic face model, a texture can be added to the facial model. Texture mapping is a popular technique to enhance the detail and quality of a face model. The details of the face models are important for rendering realistic results [10]. Previous works in ([89], [85], [47]) have used a method called *uv*-mesh texture mapping for texturing facial geometry of a PDE-based model. They developed a texturing method by associating texels of the texture map with the vertices of the *uv*-mesh which is more simple compared to the conventional method of texture mapping [85]. Another work in [88] is more straightforward where they take advantage of a PDE-based method since 2D parametric domains are not only used to find the solution to the PDE but also used for determining the coordinates of the texture mapping.

However, the texturing method used in this research is simpler compared to the previous texturing techniques mentioned above. The face data from 3D FACS give an advantage for texturing process of 3D PDE face. As mentioned in Section 4.2, the OBJ files of face geometry taken from the database have the information for vertices position, vertices normal and vertices texture coordinates. For utilizing the given information, a procedure using a MEL script has been written to copy all the information from the original face mesh to the OBJ files that contains the geometry of PDE. The procedure is done in four simple steps as follows:

- First, overlap the PDE mesh to the original mesh. This process is shown in Figure 4-9(a).
- Next, find the closest point for each vertices in the PDE mesh corresponding to the original mesh using MEL script. Based on the

nearest point of selected vertex in the original mesh (illustrated by yellow colour in Figure 4-9(b)), MEL script copy the vertex normal and vertex texture coordinate to the new file. The pseudo code used in this step is attached as **Appendix C**.

- Finally, the OBJ file of PDE mesh is combined with the new file written by the MEL script, together with the MTL file. A combination of those three files produces a geometry mesh of PDE face complete with texture image. Figure 4-9(c) is illustrates the textured PDE face.

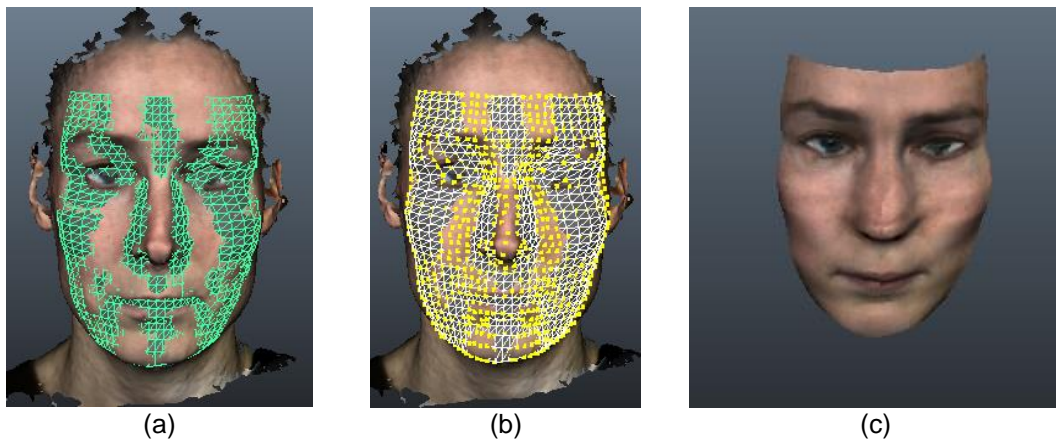


Figure 4-9 Procedure for texturing PDE face, (a) two mesh overlapped together, where green mesh is PDE face (b) the closest mark with yellow vertices, and (c) PDE face with texture

4.5 Results and discussions

So far, this chapter has focussed on the methodology of PDE surface generation for human face by utilizing a solution to a fourth order PDE subject to a suitable set of boundary conditions. The PDE face has been implemented using C# and MEL script. In this section, the discussion will continue with some important results that have been achieved using a PDE method for generating a face.

As mentioned in the previous chapter, modelling a face using PDE method gives an advantage in terms of storage capacity. In Section 4.4.1, it is shown that the PDE formulation is able to store the face information in terms of M_f (see Equation (4.2)). This is an efficient way for storing the data of human faces compared to defining the face geometry in polygon mesh. For example, the face image taken from the 3D FACS database range in mesh density between 15 K and 16 K vertices and the size of file, varies from 3 MB to 5 MB. However, the PDE formulation is able to store the facial data efficiently by only storing small amounts of parametric data in terms of Fourier coefficients, where only 93 KB of storage is needed to store a full geometry of human face.

To demonstrate how accurate the PDE face is generated in this work, it can be shown by comparing the geometry of the face taken from the database. For this purpose, the PDE mesh and the original mesh is overlaid to each other as shown in Figure 4-10. The green surface represents the PDE generated face, which is placed on the corresponding facial data from the 3D FACS database. In addition, three cut-planes are created through the facial data where:

- Cut-plane 1 and cut-plane 2 is randomly placed where the aim is to measure the accuracy of the data the PDE face has produced.
- Cut-plane 3 is a vertical plane that lies from head to the chin. This vertical cut-plane is not located at the centre of the face, as the centre of the face does not affect the action unit movements.

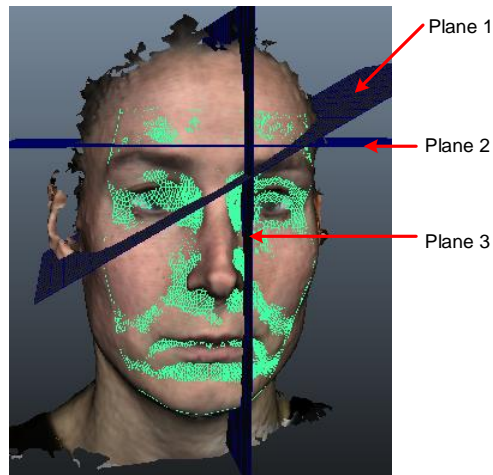
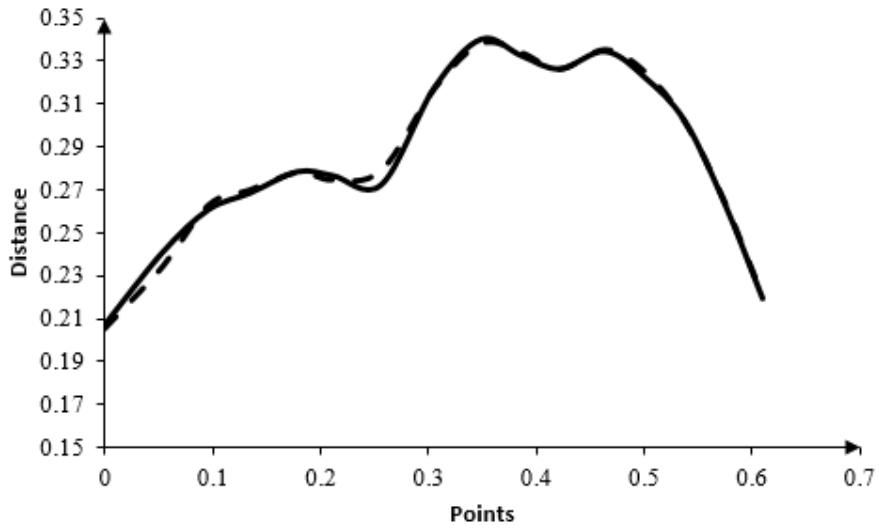


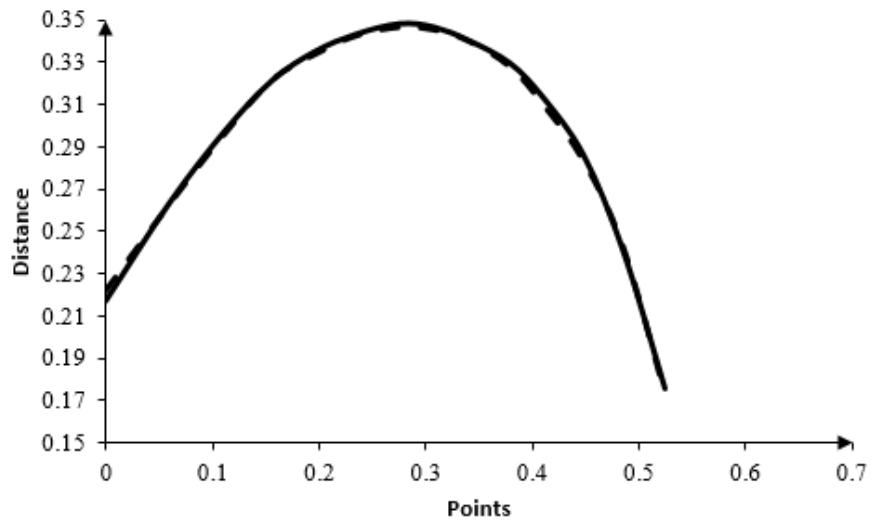
Figure 4-10 Two meshes, PDE face (green mesh) and original face mesh overlapped to each other with three different cut-planes

For the purpose of measuring accuracy, these are the steps taken:

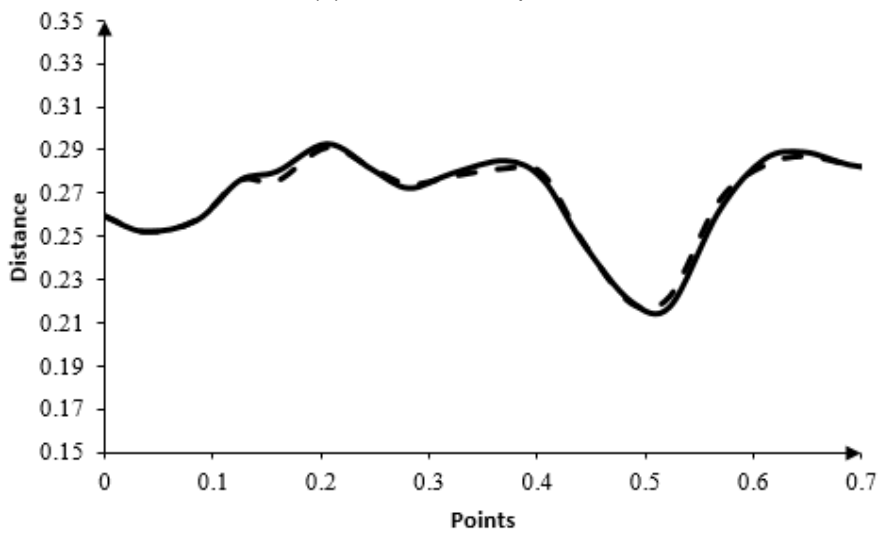
1. The plane is placed on the selected position as shown in Figure 4-10.
2. Then 20 points on the plane is chosen uniformly.
3. A simple algorithm is used to seek nearest points on the original mesh and the corresponding points from Step 2.
4. The first point is taken as the initial point. From this initial point, the distance of each corresponding points are calculated.
5. All the points from original mesh taken from Step 3 are represented as x-axis.
6. The distance of each corresponding points from Step 4 are represented as y-axis.
7. Step 3 until Step 6 are repeated for points on PDE face mesh.
8. The results for each plane are plotted in the same graph for both original and PDE face meshes using Microsoft Excel as illustrated in the Figure 4-11.



(a) diagonal cut-plane



(b) horizontal cut-plane



(c) vertical cut-plane

Figure 4-11 Comparison of the accuracy between original mesh and PDE mesh for different cut-plane.

Note that the solid line represents the original mesh from the database and the dotted line is the mesh of the generic PDE face. Figure 4-11(a), Figure 4-11(b) and Figure 4-11(c) respectively represent diagonal cut-plane (Plane 1), horizontal cut-plane (Plane 2) and vertical cut-plane (Plane 3). From Figure 4-11, it is clear that a face generated by PDE method has close geometry to the original mesh taken from the database.

4.6 Summary

In this chapter, the first objective of this research has been achieved. The methodology used in this chapter has successfully utilized a PDE-based formulation for generating a smooth 3D face. The generated PDE face efficiently represents the human face based on PDE formulation using small sets of parameters. It is also demonstrated that the PDE face generated in this work only requires a small amount of storage and accurately represents the face model relative to the original mesh from the database. The texturing method of the PDE face is also explained in this chapter. The work presented in this chapter is summarized graphically in Figure 4-12 for easy reference.

PDE face parameterisation scheme

Output

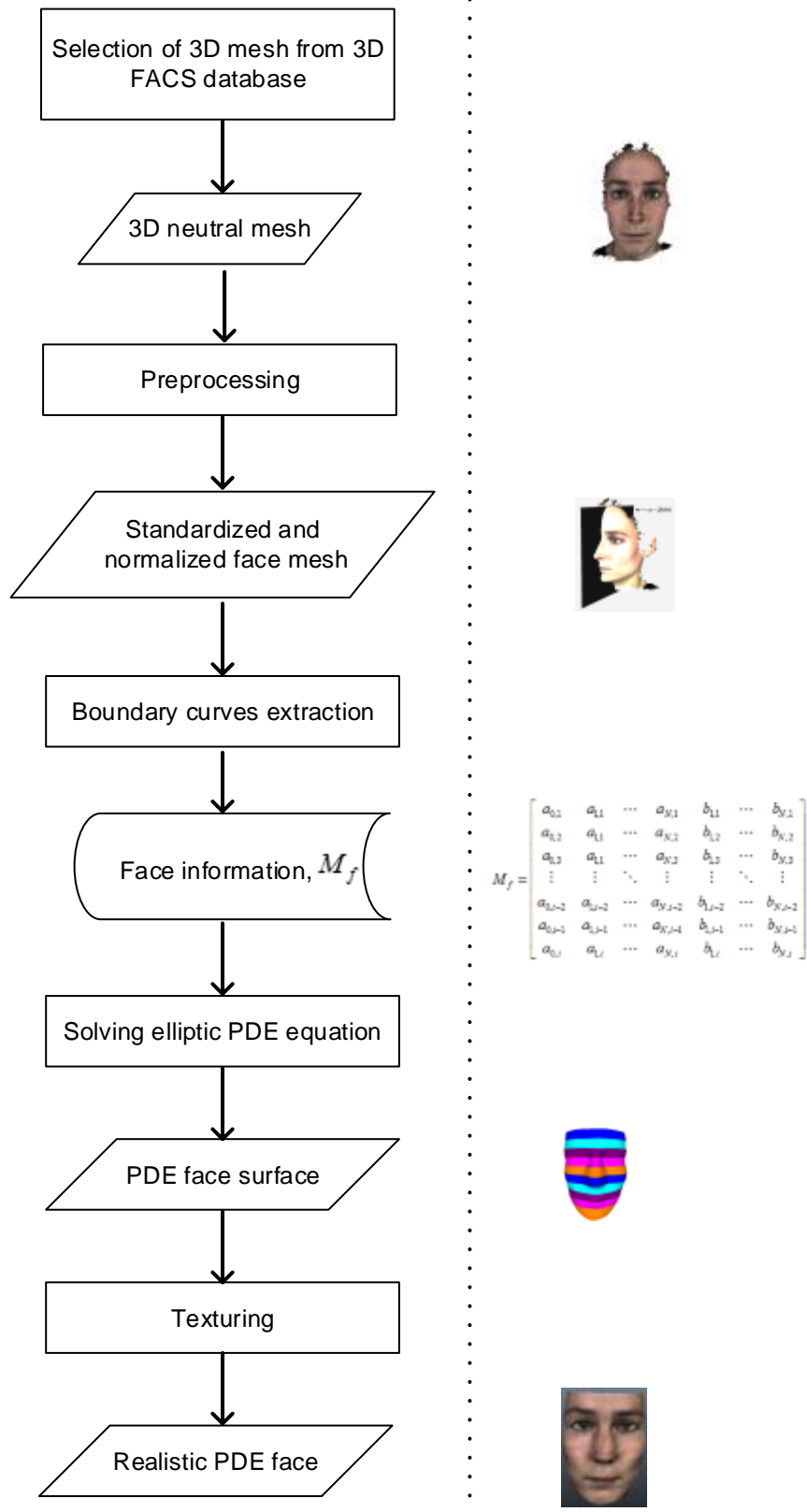


Figure 4-12 Flowchart shows methodology used for modelling PDE face

5. PDE-based Facial Expressions

5.1 Introduction

The methodology of a parameterisation of a face for generating a PDE face has been discussed in Chapter 4. A parameterised model is a powerful tool for facial animation as it only needs a small number of parameters for manipulating the shape of the model to create a sequence of images [30]. A good parameterised face model should allow any given face to be specified by only selecting the appropriate values of parameters for creating desired expressions [97]. In other words, a variety of facial expressions can be obtained by manipulating merely few parameters [21].

This chapter continues with a methodology for manipulating the PDE face for creating desired facial expressions. The next section examines common basic expressions in computer animation: happy, disgust, surprise, fear, sad and anger, and a combination of action units that affect given expressions given. Then the methodology for extracting the boundary curves of a given action unit is described and a PDE descriptor is introduced. The PDE descriptor is an efficient way to represent facial movements and acts as parameter controller of a given action unit. Using the combination of a PDE descriptor, the four facial expressions are generated.

5.2 Facial expressions based on FACS

Facial expressions result from contraction of face muscles [41]. FACS that has been developed by Ekman and Friesen [57], has become a standard for measuring facial expressions. They describe the muscle movement in terms of action units (AUs). Thus, combinations of AUs can represent a facial expression.

In this section, the combination of action units that represents basic expressions is determined. The basic expressions refers to happy, sad, disgust, surprise, anger and fear [98]. Work in [99] has listed some action units that are related with six basic expressions as shown in Figure 5-1.



















<p>AU1</p>  <p>Inner Brow Raiser</p>	<p>AU2</p>  <p>Outer Brow Raiser</p>	<p>AU4</p>  <p>Brow Lowerer</p>	<p>AU5</p>  <p>Upper Lid Raiser</p>	<p>AU6</p>  <p>Cheek Raiser</p>	<p>AU7</p>  <p>Lid Tightener</p>
<p>AU9</p>  <p>Nose Wrinkler</p>	<p>AU10</p>  <p>Upper Lip Raiser</p>	<p>AU12</p>  <p>Lip Corner Puller</p>	<p>AU15</p>  <p>Lip Corner Depressor</p>	<p>AU16</p>  <p>Lower Lip Depressor</p>	<p>AU17</p>  <p>Chin Raiser</p>
<p>AU20</p>  <p>Lip Stretcher</p>	<p>AU23</p>  <p>Lip Tightener</p>	<p>AU24</p>  <p>Lip Pressor</p>	<p>AU25</p>  <p>Lips part</p>	<p>AU26</p>  <p>Jaw Drop</p>	<p>AU27</p>  <p>Mouth Stretch</p>

Figure 5-1 Some action units that relate to six basic expressions. This figure is reproduced from [99].

However, there are no unique combinations of AUs that can represent given facial expressions. It is difficult to determine a clear comprehension of facial action and subsequent expressions [35]. There are many ways researchers combine AUs for representing basic facial expressions. For example, to get a

sad expression, work in [100] uses the combination (AU1+AU2+AU4+AU15), while work in [26] uses the combination (AU1+AU4+AU15+AU23). On the other hand, work from [99] grouped AUs of facial expressions as primary AUs and auxiliary AUs is given in Table 5-1.

Table 5-1 AU combinations based on primary and auxiliary AUs taken from [99]

Expression	Primary					Auxiliary				
	AU	AU	AU	AU	AU	AU	AU	AU	AU	AU
Happy	6	12				25	26	16		
Sad	1	15	17			4	7	25	26	
Disgust	9	10				17	25	26		
Surprise	5	26	27	1+2						
Angry	2	4	7	23	24	17	25	26	16	
Fear	20	1+5	5+7			4	5	7	25	26

Primary AUs mean that the AU combinations are strongly relevant to one of six expressions without ambiguity while auxiliary AUs can be added to primary AUs to provide supplementary support to the facial expression combination [99]. In addition, combining the primary AU from the same expression gives a more realistic expression. Another rule of emotion description is given by [101] where they list the qualifying criteria as shown in Table 5-2.

Table 5-2 Expression description adapted from [101]

Expression	Description criteria
Happy	AU12 must be present
Sad	Either AU1+AU4+AU15 or AU11 must be present. An exception is AU6+AU15
Disgust	Either AU9 or AU10 must be present
Surprise	Either AU1+AU2 or AU5 must be present and the intensity of AU5 must not be stronger than B
Angry	AU23 and AU24 must be present in the AU combination
Fear	AU combination of AU1+AU2+AU4 must be present, unless AU5 is of intensity E then AU4 can be absent

There are many more combinations of action units as given in ([41], [100], [22], [26]). The combinations vary from simple combinations to more complicated combinations to form basic expressions. However, the simplest action units used for modelling simple facial expressions are given by [21], where they only

modelled AU1, AU2, AU4 and AU12 for generating three different facial expressions, i.e. friendly, angry and sad.

After reviewing different approaches in combining the AUs, the work in this thesis was carried out by combining the works from [99] and [101]. Hence, the formulation for combining the AUs is given in Table 5-3.

Table 5-3 Proposed AU combinations for facial expression

Expression	Main		Supplement					
	AU	AU	AU	AU	AU	AU	AU	AU
Happy	12		6	16	25	26		
Sad	1	15	4	17	7	25	26	
Disgust	9		10	17	25	26		
Surprise	1+2		5	26	27			
Angry	23	24	2	4	9	16	17	25 26
Fear	1+2+4		5	7	20	25	26	

From Table 5-3, the classification of AUs falls into two categories; main AU/AUs and supplementary AUs. It is believed that the main AU is sufficient to represent certain facial expressions. For example, AU12 (Lip corner puller) can represent a happy expression without a combination of other AUs. Similarly, disgust expression is described by AU9 (Nose wrinkler). However, for surprise expression, it needs the combination of AU1 (Inner brow raiser) and AU2 (Outer brow raiser).

In the latest research by Jack et al [102], they claimed that the basic facial expressions only distinguish four categories; happy, sad, fear/surprise and disgust/anger. Their experimental data shows that surprise and fear shared the common sign (i.e. the wide-open eyes) while disgust and anger shared the common AU9 (nose wrinkler). Work by [22] also mentioned that they have difficulty in differentiating the disgust expression to anger expression, while work in [18] put fear and surprise in the same category.

Hence, to avoid ambiguity of generated expression, the work in this thesis only implemented four basic expressions, i.e happy, sad, disgust and fear.

5.3 Facial action units extraction

Previous work G.G. Castro et al in [88] used mathematical transformations for creating generic facial expressions by identifying the boundary curves and points in these curves that are affected by each expression. They have successfully generated different facial expressions; smiling, frowning, mouth opening, lateral shift of the lower lip, eyebrow raising and eye closing.

The approach used in this work is quite similar with the approach in [88]. Instead of using mathematical formulation, this research used the FACS formulation as facial expression controls. In particular, FACS is adopted to control the modification of the neutral PDE face to a generic PDE face with a given action unit. By adjusting the position of the boundary curves related to a given action unit, the generated PDE face is created with that given action unit.

5.3.1 Identification of facial muscles

From the Table 5-3, there are 16 action units that relates to the four basic expressions that have been determined. The next step is to identify the facial muscles that contract to produce a given action unit. This step is important for two reasons:

- to recognize the facial features that are affected by AU contraction.
- to identify the number of boundary curves that are affected by action units.

As mentioned in Section 4.3.1, this research adopted the facial features from [96] as given by Figure 4-5. These 19 feature points are used as a reference when aligning the neutral mesh and the mesh with an action unit. This aligning process will be explained in detail in the next subsection.

Table 5-4 Action unit related with facial expressions and their corresponding facial muscles adapted from [51] and [57].

Action unit	Facial muscles
AU1 (Inner brow raiser)	Occipito-Frontalis
AU2 (Outer brow raiser)	
AU4 (Brow lowerer)	Procerus
AU5 (Upper lid raiser)	
AU6 (Cheek raiser)	Orbicularis Oculi
AU7 (Lid tightener)	
AU9 (Nose wrinkle)	Levator Labii Superioris, Alaeque Nasi
AU10 (Upper lip raiser)	
AU12 (Lip corner puller)	Zygomaticus minor and major
AU15 (Lip corner depressor)	Depressor Anguli Oris
AU16 (Lower lip depressor)	Depressor Labii Inferioris
AU17 (Chin raiser)	Mentalis
AU25 (Lips part)	Depressor Labii Inferioris, Orbicularis Oris
AU26 (Jaw drop)	Masseter

This subsection focuses on identifying the boundary curves that are affected by a corresponding action unit. In order to identify the affected boundary curves, some knowledge about facial muscles is needed. For this purpose, the facial muscles anatomy taken from [51] is used as a reference. Figure 5-2 shows a total of 17 muscles that responsible for contraction of action units. According to Table 5-3, the action units and their respective contracted facial muscles is determined and summarized in Table 5-4. The summary in Table 5.4 is adapted from [51] and [57].

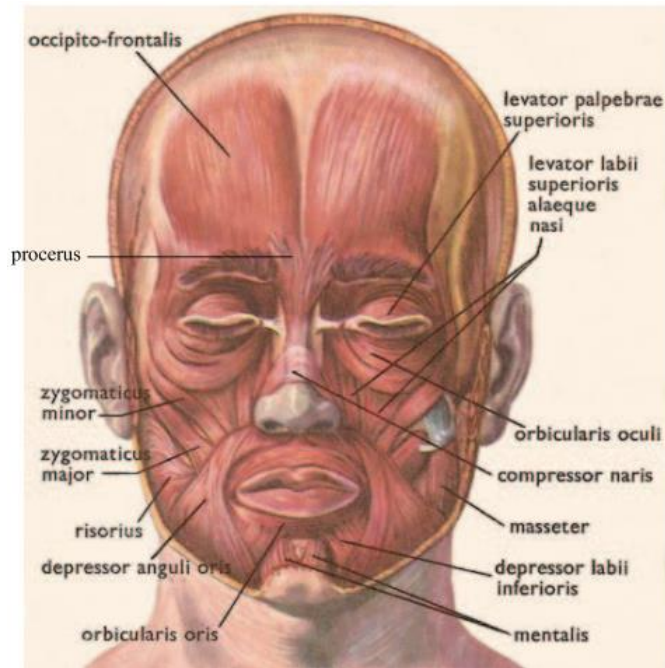


Figure 5-2 Facial muscle anatomy taken from [51]

However, all 17 facial muscles, as illustrated in Figure 5-2, do not clearly describe the facial movement of corresponding action units. For more clear identification, the work from [16] is referred to as they have specified 31 muscles on their face model. They adapted the model by Water's [31], which incorporates the FACS representation and named the model as facial musculature. These 31 muscles include:

- three sphincter muscles (two Orbicularis Oculi and a Orbicularis Oris).
- eleven pairs of linear muscles (symmetrical on both side of face).
- three linear muscles on each cheek, i.e. Cheek Sup, Cheek Center and Check Inf.

Figure 5-3(a) shows all 31 muscles that lie on face model extracted from [16] where the yellow circles are sphincter muscles and the red lines are linear muscles.

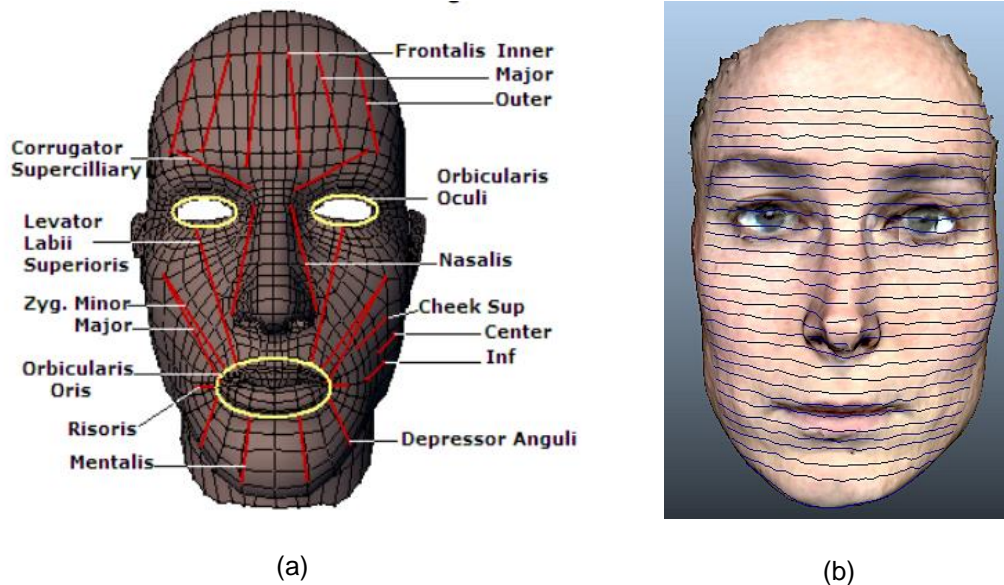


Figure 5-3 Facial musculature (extracted from [16]) given in (a) is compared to boundary curves in (b).

Note that the boundary curves in Figure 5-3(b) are neutral facial expressions that have been extracted in Section 4.3.2. Here, neutral facial expression means eyes, brow, cheek and lips that are relaxed [103]. These boundary curves of a neutral face act as a template for extracting various action units in the next subsection.

By comparing the facial musculature in Figure 5-3(a) to boundary curves in Figure 5-3(b) with the information given in Table 5.4, it is easier to determine the boundary curves that are affected by action units and the result is given in Table 5-5. The “X” mark in Table 5-5 represents the affected boundary curves corresponding to a given action unit. For example, the AU4 (brow lowerer) movement affected the boundary curves 4 to 8, while AU9 (nose wrinkle) movement affected the boundary curves 8 to 18. It is worth pointing out that the action units in this thesis are controlled by a group of boundary curves. Changing the boundary curves that correspond to any given action unit will manipulate the PDE face correspondingly. Then in Section 5.4,

a PDE descriptor will be introduced as a controlling parameter of an action unit.

Table 5-5 Action units and the corresponding boundary curves affected

Curve No	Action unit													
	1	2	4	5	6	7	9	10	12	15	16	17	25	26
1	X	X												
2	X	X												
3	X	X												
4	X	X	X	X										
5	X	X	X	X										
6	X	X	X	X										
7			X	X										
8			X	X	X	X	X							
9					X	X	X							
10					X	X	X							
11					X	X	X							
12					X	X	X	X						
13							X	X						
14							X	X						
15							X	X	X					
16							X	X	X					
17							X	X	X					
18							X	X	X					
19								X	X					
20								X	X	X				
21								X	X	X				
22								X	X	X			X	X
23								X	X	X			X	X
24								X	X	X			X	X
25									X	X	X	X	X	X
26										X	X	X	X	X
27										X	X	X	X	X
28										X	X	X	X	X
29											X	X		
30											X			
31														

5.3.2 Curve extraction

As mentioned previously, the boundary curves that have been extracted in Chapter 4 are used as a template for extracting the boundary curves of a given action unit. This subsection will discuss the process for extracting the boundary curves for a given action unit.

Prior to boundary curves extraction of a given action unit, the 3D mesh face data with a given action unit is obtained from the database. Then, the selected mesh of a given action unit is performed using the pre-processing

procedure as described in Section 4.3.1. After the pre-processing procedure, the selected mesh should have a standardized and normalized mesh. Apart from standardized and normalized mesh, the face data with a given action unit also must have the same coordinates on their 19 facial feature points as the neutral mesh data as shown in Figure 5-4.

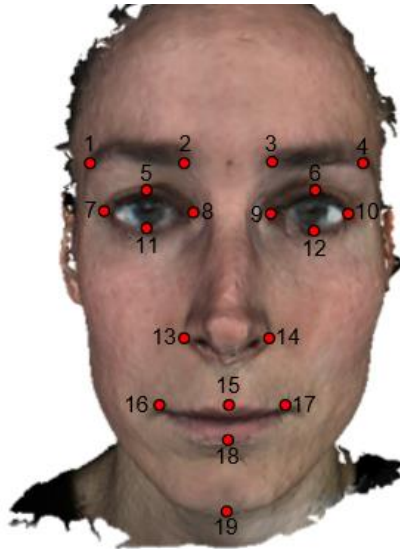


Figure 5-4 Facial feature points on Eva's face

In order to correctly place the facial feature points of neutral mesh and the mesh with a given action unit, the process is carried out by aligning the neutral mesh and the action unit mesh in the same position. For example, Figure 5-5 shows two meshes (neutral mesh and AU1 (Inner brow raiser)) that overlapped each other. It is important to mention that the facial feature points of the two meshes must be positioned so that they nearly overlap for facilitating the correct correspondence between two surfaces, except for the facial feature points that are affected by a correspondence action unit.



Figure 5-5 Two meshes overlapped; neutral mesh and AU1 mesh

For example, when aligning the AU1 mesh to neutral mesh as shown in Figure 5-5, facial feature point number 2 and number 3 are affected by large muscles in the scalp and forehead to raise the eyebrows, where AU1 muscles pull the inner part of the eyebrows upward [57]. The remaining facial feature points must be positioned in the same coordinates as the neutral mesh. For reference, the corresponding facial feature points that are affected by action units are given in Table 5-6, where "x" marks the affected facial points of a given action unit.

Table 5-6 Action units and corresponding facial feature points affected

Facial feature point	Action unit													
	1	2	4	5	6	7	9	10	12	15	16	17	25	26
1		x												
2	x	x	x	x										
3	x	x	x	x										
4		x												
5				x										
6				x	x	x								
7								x						
8								x						
9				x	x	x								
10				x										
11					x	x	x							
12					x	x	x							
13								x	x	x				
14								x	x	x				
15									x	x	x	x	x	x
16									x	x	x	x	x	x
17									x	x	x	x	x	x
18										x	x	x	x	x
19														x

Once the mesh for a given action unit is properly aligned, then MEL script reads the boundary curves that have been stored in OBJ files to the mesh as shown in Figure 5-6(a). Note the different mesh can be spotted on the forehead area when AU1 mesh is imposed to the neutral mesh as illustrated in Figure 5-6(b). Next, the MEL script is used for finding the closest point of all sets of points in the boundary curves to the AU1 mesh. The closest points are indicated by the yellow points in Figure 5-6(c). Then the MEL script reads the points coordinate of selected points on AU1 mesh and exports those points to OBJ files after closing the boundary curves. The process was repeated for all 31 boundary curves on the AU1 mesh.

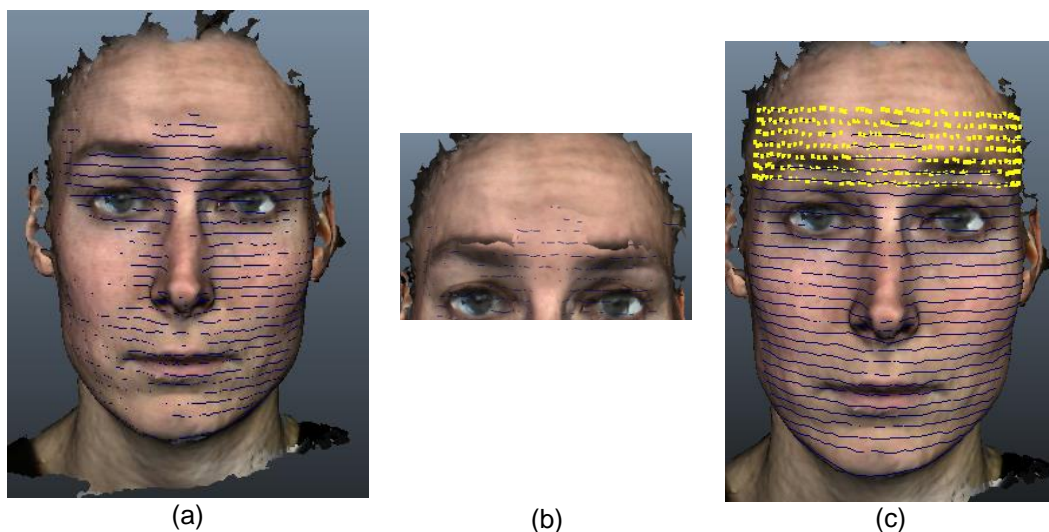


Figure 5-6 Boundary curves extraction process for AU1 (a) boundary curves from neutral face, (b) different mesh on forehead area of AU1 and (c) closest points indicated by yellow points

The above examples explain the process of boundary curves extraction for AU1. The other action units boundary curves extraction process also used the same methodology as described above. An algorithm of MEL script for extracting procedure of action unit is given in **Appendix D** as a reference.

As a results from this section, each action unit is stored in terms of boundary conditions in 31 OBJ files. Different action units have different

boundary curves. In the next section, the discussion will be continued to show the novelty of this research, where the boundary curves for representing action units are stored in efficient way, which is known as a PDE descriptor.

5.4 PDE descriptor for an action unit

Let us recall the next step in the methodology after finishing extracting all the boundary curves. In Chapter 4, it has been mentioned that all the boundary curves are approximated to finite Fourier series as given in Equation (4.1) and all 31 boundary curves are represented in matrix of Fourier coefficients, M_f as given in Equation (4.2). This matrix M_f is actually representing the natural pose of the face in terms of Fourier coefficients.

For all 31 boundary curves of action units that have been extracted in the previous section, the Fourier coefficients are also stored in a similar way and lets name it as M_g which is given by Equation (5.1) below:

$$M_g = \begin{bmatrix} a_{0,1} & a_{1,1} & \cdots & a_{N,1} & b_{1,1} & \cdots & b_{N,1} \\ a_{0,2} & a_{1,1} & \cdots & a_{N,2} & b_{1,2} & \cdots & b_{N,2} \\ a_{0,3} & a_{1,1} & \cdots & a_{N,3} & b_{1,3} & \cdots & b_{N,3} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{0,i-2} & a_{1,i-2} & \cdots & a_{N,i-2} & b_{1,i-2} & \cdots & b_{N,i-2} \\ a_{0,i-1} & a_{1,i-1} & \cdots & a_{N,i-1} & b_{1,i-1} & \cdots & b_{N,i-1} \\ a_{0,i} & a_{1,i} & \cdots & a_{N,i} & b_{1,i} & \cdots & b_{N,i} \end{bmatrix} \quad (5.1)$$

Even though M_g represents the facial data for a given action unit, this is not an efficient way for storing the coefficients and not practical for manipulating the PDE face model. For a more convenient approach, the action unit data is stored in terms of matrix M_i where

$$M_i = M_f - M_g \quad (5.2)$$

and i is the number of AUs that it represents. For example, M_1 represents the coefficients of AU1. For an easy reference, M_i is referred to as a PDE descriptor. An example of a PDE descriptor for AU1 coefficients is given in **Appendix B**. From **Appendix B**, it is noticed that only the affected boundary curves have the coefficients value (in this case the boundary number 1 to 7), while the other coefficients have a NULL value.

Using the PDE descriptor, a PDE face that has been generated in Chapter 4 can be manipulated to generate a PDE face with an action unit. In order to generate a PDE face with a given action unit, the PDE descriptor is added to neutral Fourier coefficients, M_f . Then Biharmonic equation (Equation 3.12) is solved to generate the PDE face with the given action unit. After the PDE face with a given action unit is generated, then a generic PDE face is textured. Figure 5-7 shows some examples of generic PDE faces with action units. Figure 5-7(a), Figure 5-7(b), Figure 5-7(c) and Figure 5-7(d) respectively represent AU4 (Brow lowerer), AU12 (Lip corner puller), AU14 (Dimpler) and AU17 (Chin raiser).

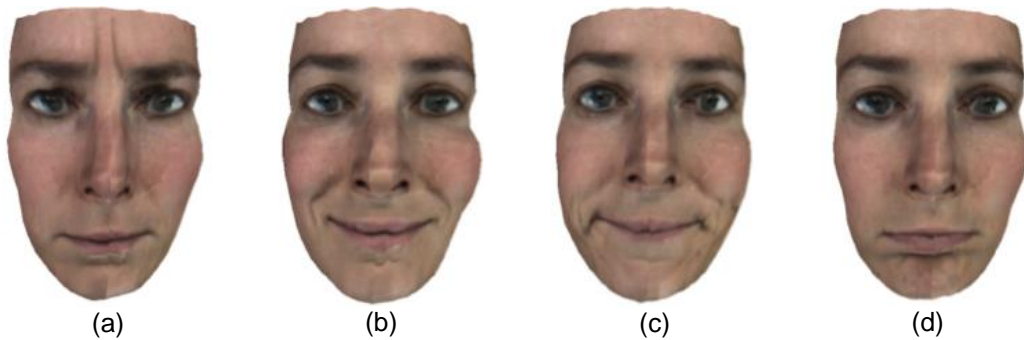


Figure 5-7 PDE generated facial action units (a) AU4 (b) AU12 (c) AU14 and (d) AU17

5.5 Facial expression using PDE descriptor

In this section, the PDE descriptor, which has been formulated in the previous section, is used for manipulating the neutral PDE face to have an expression. As mentioned in the earlier section, a combination of action units will generate a facial expression.

For example, happy expressions can be generated by combining the AU6 (Cheek raiser) and AU12 (Lip corner puller) while sad expressions are generated by combining AU1 (Inner brow raiser), AU4 (Lip corner depressor) and AU15 (Lip corner depressor). Thus, to manipulate a neutral PDE face to a happy expression, the PDE descriptors M_6 and M_{12} are added to M_f . Similarly, the PDE descriptors M_1, M_4 and M_{15} were added to M_f for generating a PDE face with sad expression. Then the Biharmonic equation is solved to generate the PDE face with expression. After that, the texturing process is performed. Four basic expressions are illustrated in Figure 5-8. Figure 5-8(a), Figure 5-8(b), Figure 5-8(c), and Figure 5-8(d) respectively representing happy, sad, fear and disgust expressions.

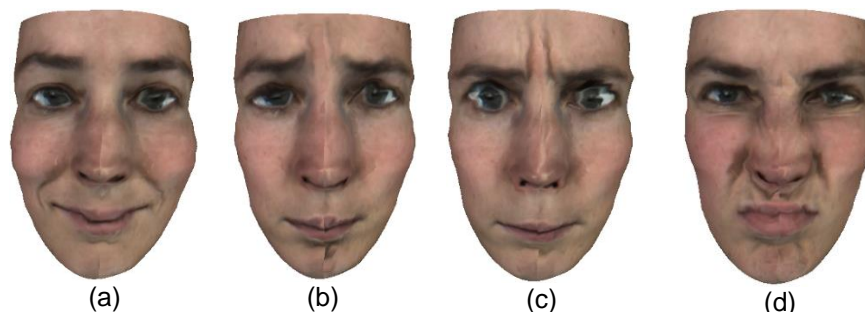


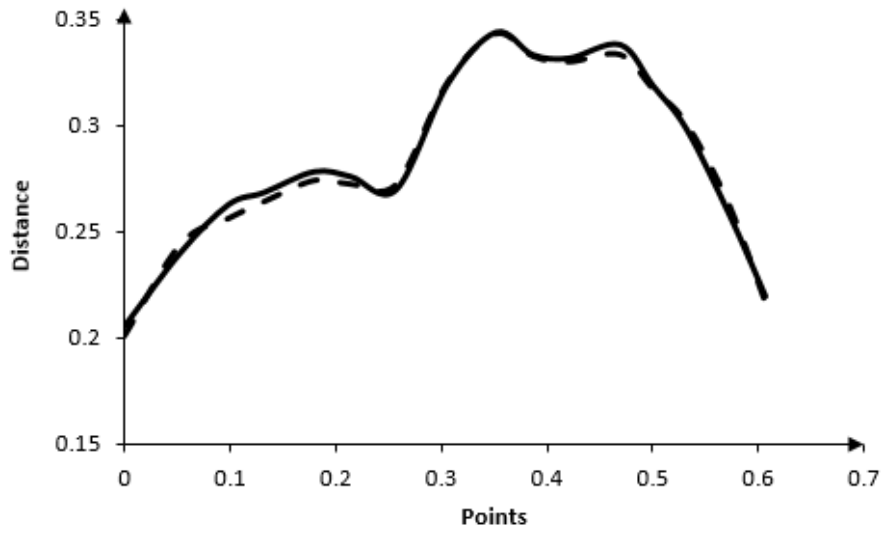
Figure 5-8 PDE generated facial expressions (a) happy (b) sad (c) fear (d) disgust

Note that the fear expression is generated using the combination of M_1 , M_2 and M_4 , while combining the PDE descriptors of M_9 and M_{17} produces disgust expression.

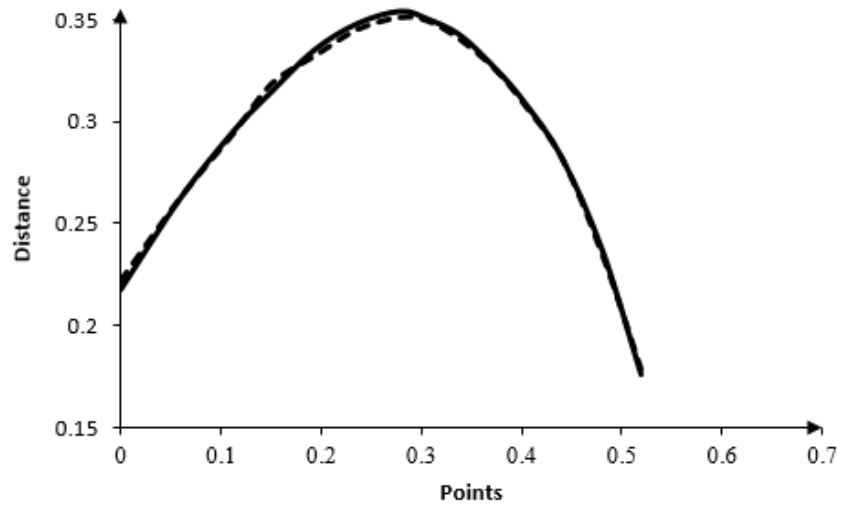
5.6 Results and discussions

Similarly, as with Chapter 4, this results and discussion section will discuss some important results that have been achieved in this chapter. In particular, the comparison of a generic PDE face with a given action unit and original mesh taken from the database is given graphically. Furthermore, the discussion also includes a comparison of storage based on an original facial data and a generic PDE face.

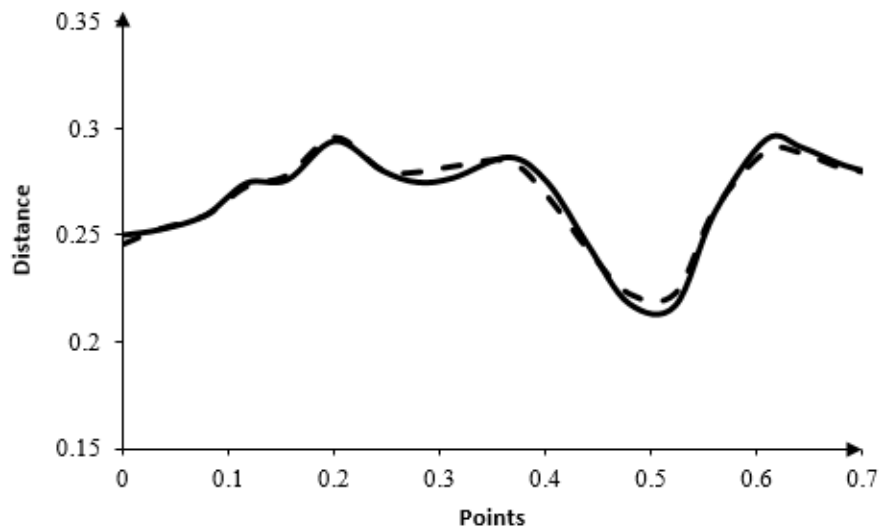
Using the same comparison method for comparing accuracy of PDE generated meshes as described in Section 4.5, the results are shown in Figure 5-9 for AU4 mesh and Figure 5-10 for disgust expression. Note that the solid line represents the original mesh from the database and the dotted line is the mesh of the generic PDE face.



(a) diagonal cut-plane

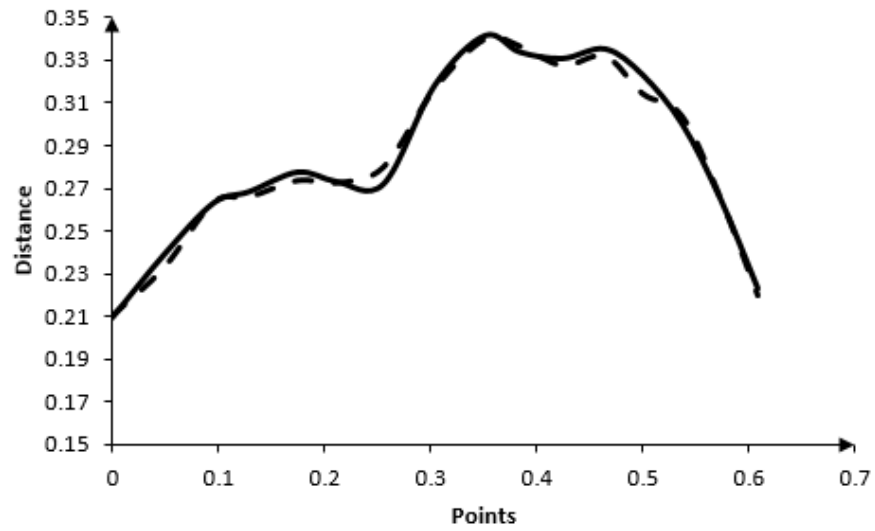


(b) horizontal cut-plane

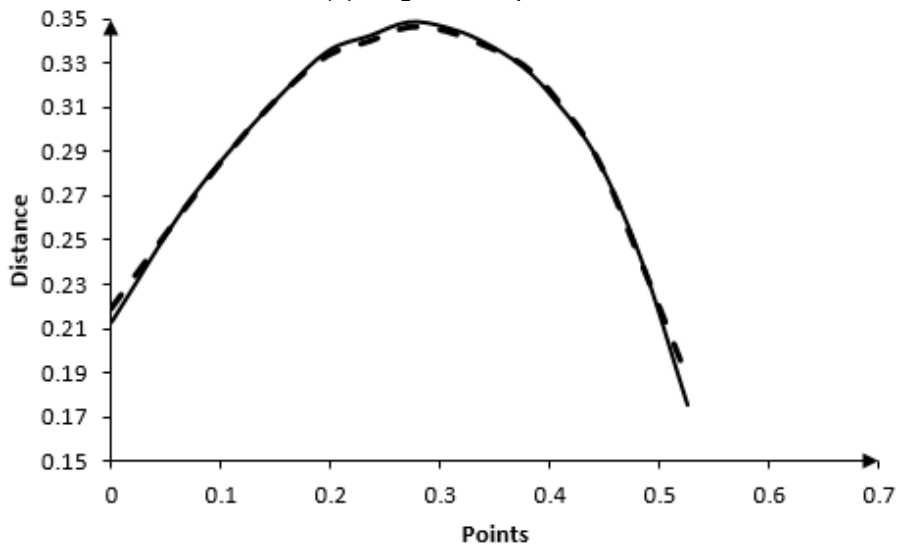


(c) vertical cut-plane

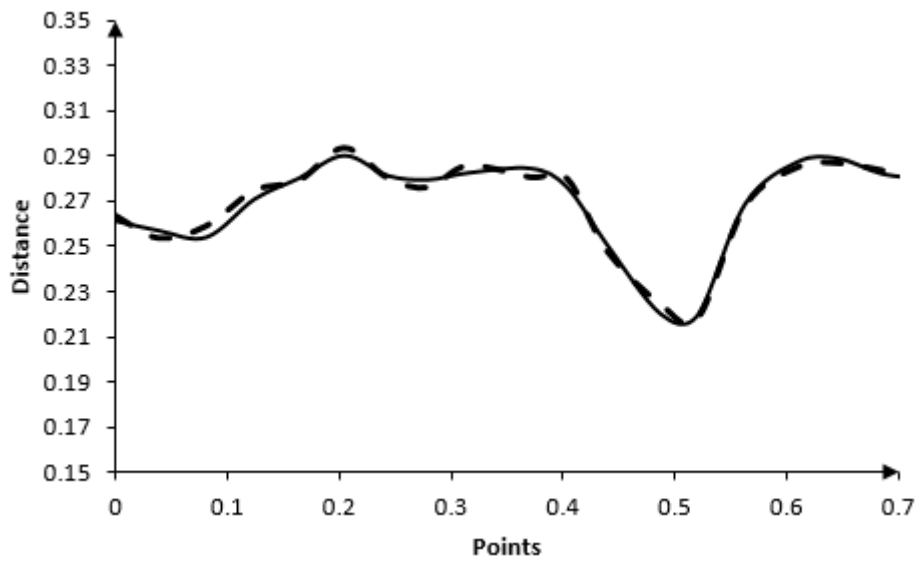
Figure 5-9 Comparison of the accuracy between original mesh of AU4 and PDE mesh of AU4 for three different cut-planes.



(a) diagonal cut-plane



(b) horizontal cut-plane



(c) vertical cut-plane

Figure 5-10 Comparison of the accuracy between original mesh of disgust and PDE mesh of disgust expression for three different cut-planes.

From the graphs shown above, it is noted that the PDE generated face of AU4 and disgust expression are in close agreement with the real action unit/expression found in the database. All other action units and expressions generated in this works also have the same agreement.

It is mentioned in Section 4.5 that 3 MB of storage is needed for storing full geometry of a human face. Note that for each boundary curves that represents the Fourier coefficients, an amount of 3 KB of storage is needed. Since there are 31 boundary curves stored for a full face, then the total storage for generating a neutral PDE face is 93KB. For example, AU1 contains boundary curves ranging from number 1 to 6, which means that 18 KB is needed for storing the PDE descriptor. Hence, the total of storage needed for storing a PDE face with AU1 will be only 111 KB. Worse case, even if all boundary curves affected, regardless which AUs involved, a maximum of 186KB will be used to store them. This clearly indicates that using this new approach promotes storage efficiency up to 93.8%. Figure 5-11 demonstrates further support on the author’s claim.

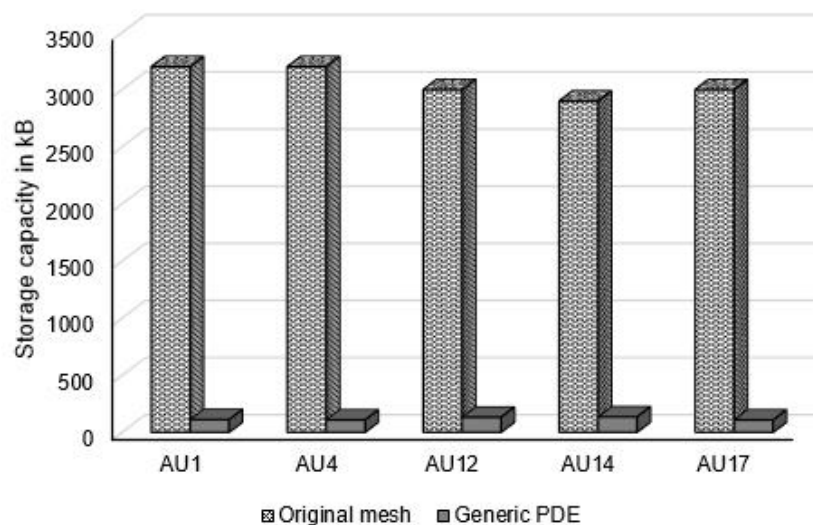


Figure 5-11 Comparison of five different AUs using original mesh and generic PDE.

However, it is worth mentioning that the work in this thesis is not considering the action units for mouth opening such as AU25 (Lip part) and AU26 (Jaw drops). The work by Sheng *et al* [47] has shown that they successfully generated a PDE face with an open mouth by introducing two invisible boundary curves on the upper and lower inner lip contours. So, it is possible to present the AU25 and AU26 in terms of PDE descriptors for future work. In addition, the research regarding the lips movement is another interest in facial animation as shown in [91] and [104]. Computer animation for facial expressions are categorized into two; expressions for illustration of a person's emotions and expressions for generating synthetic visual speech [19]. The former is the focus of this study, while the latter is beyond the scope of this thesis.

PDE face parameterisation scheme

PDE face manipulation scheme

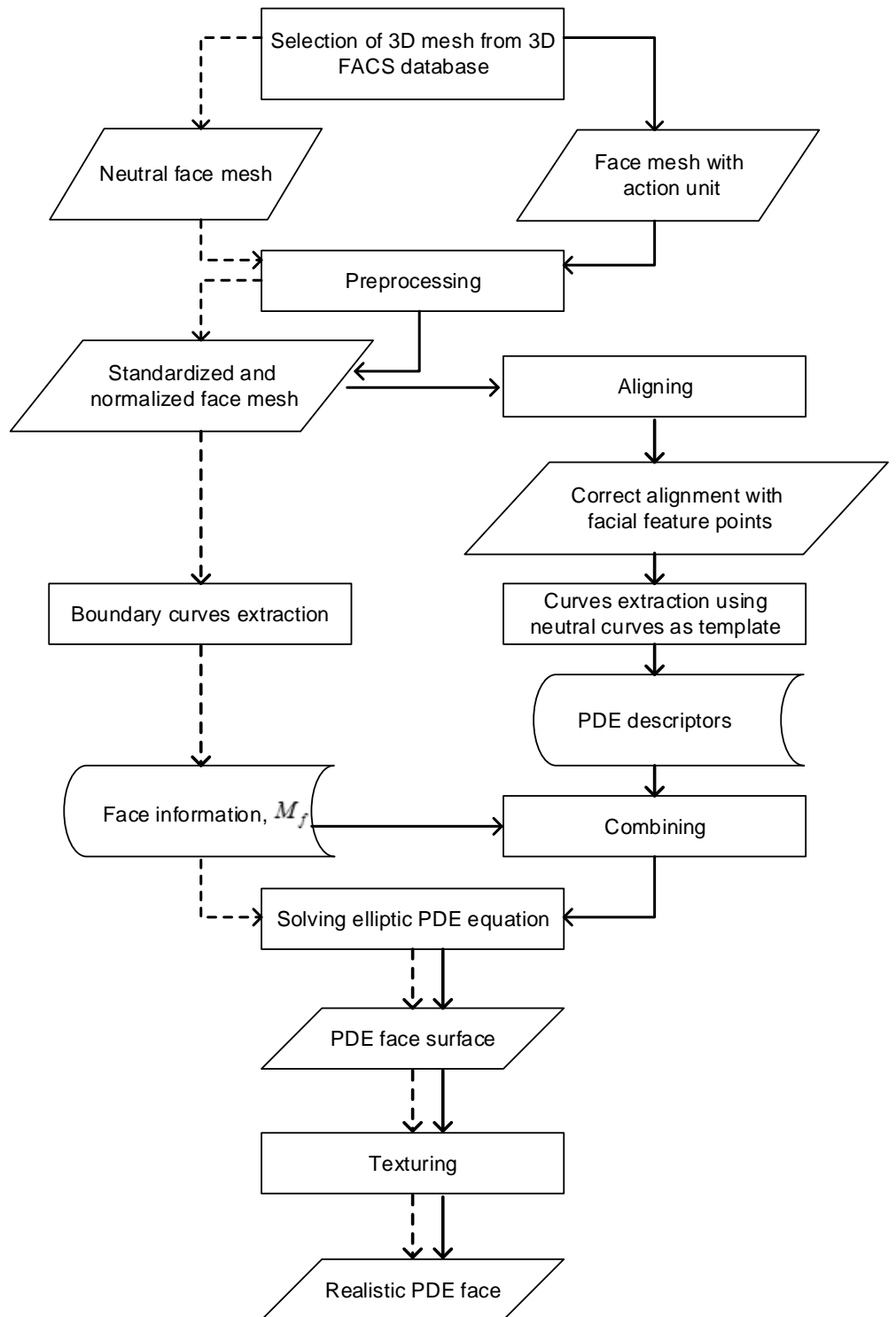


Figure 5-12 Flowchart shows methodology used for modelling and manipulating a PDE face

5.7 Summary

A methodology for manipulating a generic PDE face of neutral expression to a face with expression has been described in this chapter. Three objectives have been achieved in this chapter. Firstly, it has demonstrated the formulation for storing the action unit data in terms of Fourier coefficients or known as PDE descriptors. The formulation of PDE descriptors is based on the anatomy of the human face and produced more natural facial motions using a relatively small number of parameters based on facial muscle structure. A generic PDE face in this work is an anatomically based model, which incorporate FACS as a controller of parameters. Thus, the second objectives of this chapter are fulfilled by manipulating the generic PDE face using the PDE descriptor. The PDE descriptor uniquely represents an action unit and can be added to a neutral PDE face for generating a PDE face with a given action unit. It is also shown that synthesizing four different facial expressions becomes relatively straightforward using PDE descriptors. The combination of PDE descriptors produces different facial expressions of a generic PDE face hence, the third objective has been achieved.

To this end, the summary of the work that has been done in Chapter 4 and Chapter 5 is given in Figure 5-12. The arrows with dotted lines represent the methodology described in Chapter 4 while the arrows with solid lines represent the methodology for manipulating the generic PDE face as described in this chapter. From Figure 5-12, it is illustrated that the parameterised model of a PDE face, developed in this thesis, can create facial expressions by specifying an appropriate set of PDE descriptors.

6. Animation of PDE-based Faces

6.1 Introduction

There are three steps involved for animating a human face: designing a facial model, modifying the facial expression and animating the facial images sequences [43]. An overview of animation framework used in this thesis is given by Figure 6-1.

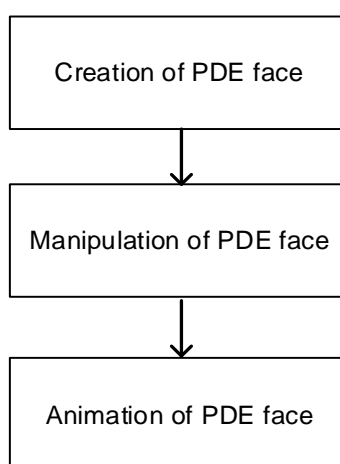


Figure 6-1 Overview of PDE face animation process

Figure 6-1 shows an overview of the animation process, which involves three major stages: creating the PDE-based face model, creating a PDE-based face model with expressions, animating the PDE-based model.

In Chapter 4, the methodology of modelling the PDE face has been described in detail. Later in Chapter 5, the manipulation of a PDE face has been demonstrated, where PDE descriptions are used for making a PDE face with expressions. In this chapter, the discussion will continue with the final

stage of the animation process, where blendshapes technique is used for animating the PDE face facial expressions.

In the first section of this chapter, an introduction and some literature of facial animation using the blendshapes technique is given. Then the discussion focuses on the technique of blendshapes for animating the generic PDE face for making different facial expressions. Some results and discussion are given in the later section.

6.2 Facial animation using blendshapes

Blendshapes (which are also known as shape interpolation) are commonly used in commercial animation software packages and have become one of the popular animation technique [35]. The term “blendshape” was introduced by the computer graphics industry [105]. The blendshapes technique is popular due to its simple linear representation [106] and in terms of control, this technique has intuitive and flexible controls [59]. The blendshapes method was introduced by Parke [10] and the efficiency of this method interests animators in the entertainment industry. In film projects such as *Lord of the Rings*, *Stuart Little*, *King Kong* and *Star Wars*, the animators used blendshapes for animating the facial models [68].

Although animating using blendshapes is a popular technique among animators, it is very time consuming and takes considerable labour costs [36]. For example, an animator must remember the function of 50 to 100 commonly used sliders for efficiently using a blendshape model and the process to locate

the desired slider is not a trivial task [106]. An example of the blendshape slider interface is shown in Figure 6-2.

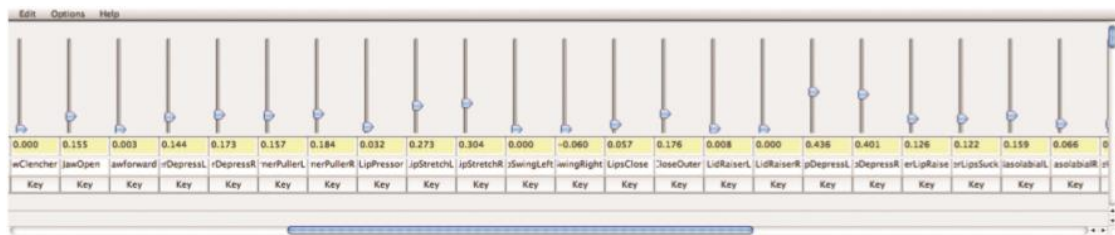


Figure 6-2 A portion of blendshape slider interface taken from [106]

Figure 6-2 shows a part of the blendshape slider interface, which is usually used by professional animators for manipulating a face model. The model actually has more than 100 sliders, where the complete set of sliders doesn't fit on a computer display. Therefore, to model a rough face of expression, a professional digital artist needs one hour for producing one second of animation [106].

The considerable time taken when animating using blendshapes held a particular interest for the computer graphics community to improve the blendshape's performance. Work by [107] has segmented the face geometry for several regions, in order to guarantee the modification of a specific part of the surface, by designing an automatic techniques that extracts a set of parameters. Segmenting the face region only hinders the problem of interference across the segment, but does not improving performance of the blendshape technique [68]. Then a work in [59] animating the blendshape face models from facial motion capture data to reduce considerable efforts, by introducing a semi-automatic technique, which means manual work is still required in this process. The work by Lewis and Anjo [106] received positive feedback from professional animators when they introduced a direct-

manipulation approach for blendshapes. They built a user interface using a commercial animation software package for animators, so that animators can select vertices and adjust to desired locations [106]. They came to the conclusion that the combination of direct manipulation and slider control, gives advantages and saves significant time as an animator does not need to recreate an expression each time it changes. Considering the work by [106], this work adapted the animation technique using blendshapes in order to save time when animating facial expressions using parametric PDE face.

6.3 PDE-based face animation using blendshapes

Previous work by Yun Shen et al [47] is based on direct parameterisation for animating a high-resolution of a PDE-based face with a relatively small number of parameters. They developed an algorithm to associate the Facial Animation Parameters (FAPs) with some feature points on the boundary curves for making facial animation independent of the reconstructed PDE face. Thus, a fixed animation scheme is used for animating a multi-resolution PDE face model. However, this approach inherits the drawbacks of direct parameterisation, where a new PDE face needs to be reconstructed when there is a changes of the boundary curves. Thus, this work is applied to blendshape animation in order to reuse the PDE face meshes that have been generated from the previous chapters.

In the previous chapter, it has been shown that four different expressions can be generated using PDE descriptors. Then all these

expressions will be animated using the blendshape. The following expression represents the blendshape face model:

$$\underline{V} = \sum_{i=0}^n \alpha_i \underline{V}_i \quad (6.1)$$

where:

- α_i are the weight and $0 \leq \alpha_i \leq 1$.
- \underline{V}_i are vertices of face base.
- \underline{V} are vertices of face target (resulting face)
- n is the total number of vertices in both faces (base and target).

Note that, both face base and face target must have the same number of vertices, same topology and the same vertex order. From Equation (6.1), the blendshape can be considered as adding a scalar to the vertices in the face base. Also note that α_0 represents the neutral face. When the value of α_i increases until α_n , the base face is changes from a neutral face to peak expression depending on the weighting value that is given. Therefore, the target face quality largely depends on the base face and the given blending weight. Figure 6-3 shows the base face and the target faces that are used in this work. From the Figure 6-3, the target faces is generated as a linear combination of a weighting value and the base face. Different weights of the linear combination will expressed a different range of facial expression.

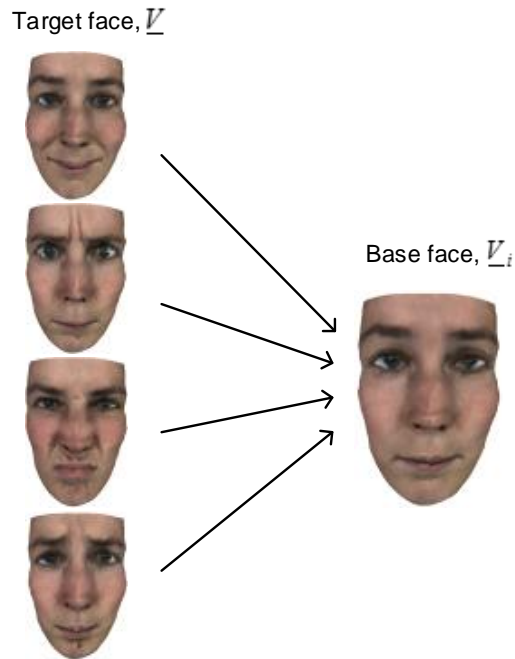


Figure 6-3 Target face and base face

Although the concept of blendshape interpolation looks simple, it is not easy to develop a blendshape face model because it is a large and labour intensive effort [105]. In traditional blendshape approach, the animators have to create large libraries of blendshape targets [107]. For instance, the character of Gollum in the film *Lord of the Rings: The Two Towers* had 675 targets [107]. The tedious work occurs when digital artists need to refine the model in many iteration processes. However, studies in [36] have shown that it is possible to automate the iterative process, where they have developed an iterative scheme by finding the combination of base model that is involved in the facial expression and their corresponding blending weight using an optimization procedure.

On the other hand, the work in this thesis is uses a base face and the target faces that have been generated using a boundary-value problem in the previous chapter as shown in Figure 6-3. As mentioned previously, the base face and the target face must have the same number of vertices and the

generated PDE face fulfilled this requirement. The approach using a generic PDE mesh for blendshape gives advantages in terms of times, as the face model and the manipulation of the face shape are not constructed from scratch.

Then the base face and target face are loaded into the animation software, Maya. In Maya, there is a features known as a deformer. This feature enables an animator to change the shape of geometry in the scene by blending, inflating, clustering, etc. A blend shape is one of the tools in the deformer category and a very powerful and useful tool for animators. Usually the animator duplicates the object and then change the shape of object by manually adjust the vertices. The original shape is known as a base object, while the adjusted shape is regarded as a target object. Then, the animation is applied by animating the target mesh to the base mesh. Changing the weight of the blendshape will change the base object in a natural-looking manner. In this work, the neutral PDE face is assign as a base face, while the PDE face with an expression is regarded as a target face.

Next, a blendshape for happy expression is generated by selecting the neutral PDE as a base mesh and a happy face of PDE as a target mesh. This procedure targets the blendshape to the neutral mesh and changes the neutral mesh to have a happy expression. To vary the weighting value, this is done by using a slider in the Maya user interface. An example of a happy expression blendshape model is given in Figure 6-4.

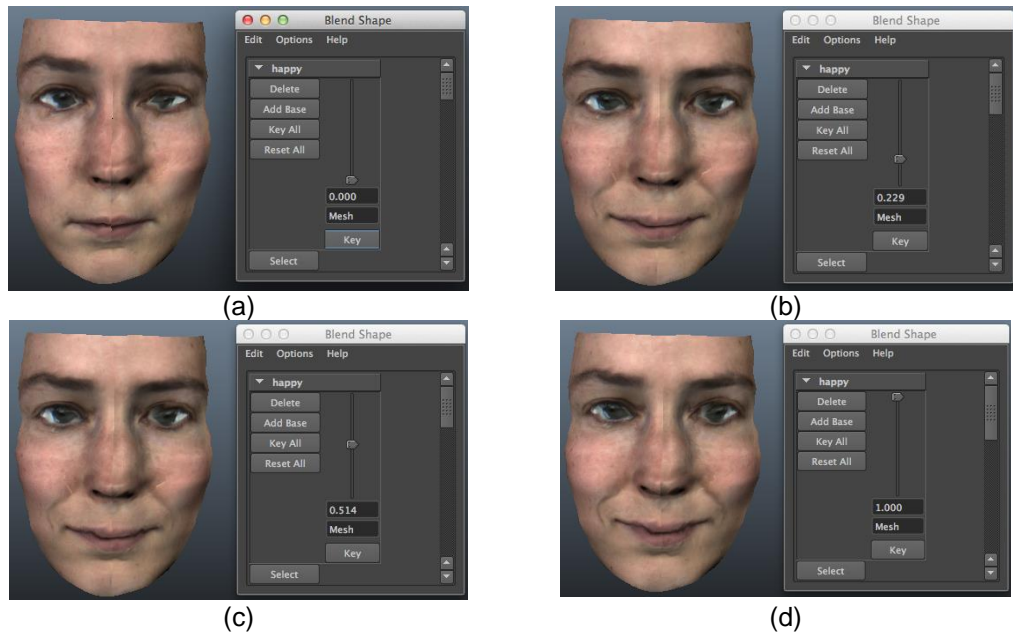


Figure 6-4 Varying weighting value for happy expression from (a) 0 (b) 0.229 (c) 0.534 (d) 1

From Figure 6-4, it is shown that from a neutral configuration of a PDE face, the expression changes to a peak expression of happiness. For other expressions, the same step is applied and one slider uniquely represents the given expressions. The examples of another two facial expressions are given in Figure 6-5. Figure 6-5(a) represents the fear expression when $\alpha = 0.633$, Figure 6-5(b) represents the disgust expression when $\alpha = 0.165$ and Figure 6-5 (c) represents the sad expression when $\alpha = 1$. It is worth mentioning that in the Maya blendshape interface, the weights value allows a range [0,1] to make sure the animators have not overridden the limit [105].

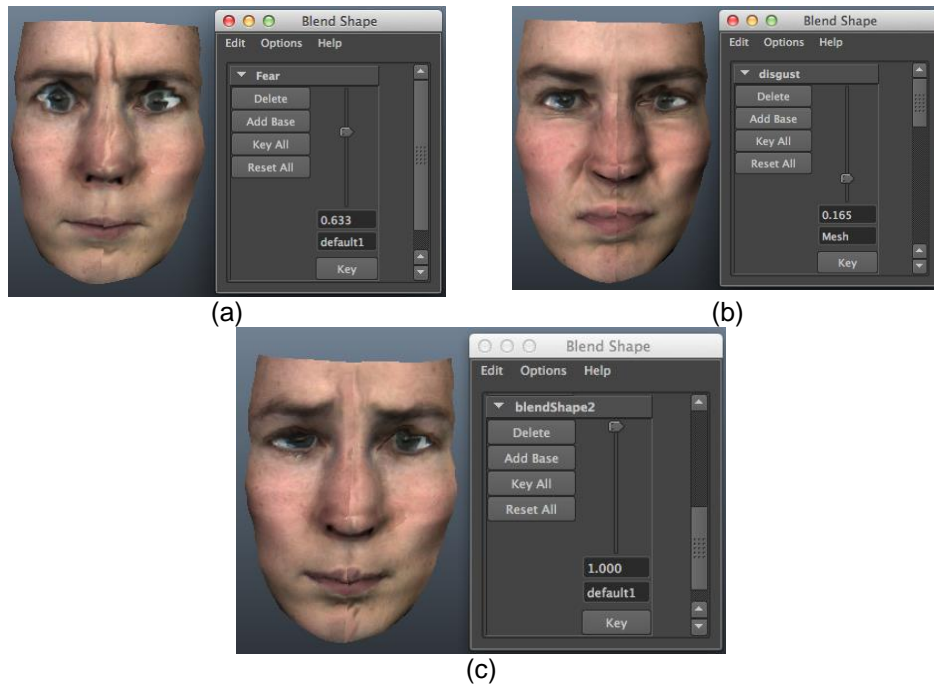


Figure 6-5 Blendshape sliders for fear, disgust and sad face expressions

Traditionally, blendshape are animated with key framing or motion capture [108]. This work employed key framing for animating the face model and Mel script used as scripting language. The result of key framing animation is given in the next section.

6.4 Results and discussions

In this section, the results of the key framing animation technique using a blendshape face are given. As mentioned previously, this task is carried out using Mel script that has been written for this purpose. Instead of using the sliders, MEL script has been used for illustration purposes of animation sequences. The animation sequences for happy and disgust expressions are given in Figure 6-6 and Figure 6-7 respectively.



Figure 6-6 Animation sequences for happy expression



Figure 6-7 Animation sequences for disgust expression

In this work, the key framing animation of blendshapes is applied separately for each expression, where from a neutral face the blendshape “happy” will blend the neutral face to a face with a happy expression by increasing the weight value. Then the weight value of “happy” blendshape decreases to get the neutral face. After that, the animation continues to the next expression, disgust, by repeating the same process for “disgust” blendshapes. The process for fear and sad expressions was conducted in the same way. The main limitation when animating the facial expressions described above is a blendshape, which is only able to animate the face from neutral to a face with expression or vice-versa.

However, the problem occurred when the author tried to animate blendshapes from a face with expression to another expression. The problem occurred is known as blend shape interference, where blendshape targets naturally have overlapping effects [105]. Animator has expecting extra cost when using blend shape approach caused by interference as it cannot be

entirely eliminated. Consequently, an animator need to manually refining the blend shape when interference is found.

The animation techniques presented in this chapter are far from a complete animation process, but sufficient to demonstrate that the PDE face models generated in this work are capable for animating facial expressions. By using the generic PDE faces as a base face and a target face, a tedious task for adjusting the vertices of face to make an expression can be eliminated.

6.5 Summary

In this chapter, the animation technique using blendshapes and key framing has been described. The animation approach uses the generic PDE face for base mesh and target mesh. This approach is time saving as the designer can use the generated model without manually adjusting to get the desired facial expression.

7. Conclusions

7.1 Summary

This thesis deals with the utilization of PDE method for modelling facial action units, creating expressions and making animation. The PDE-based face is generated as a solution to elliptical PDEs, where face generation is treated as a boundary value problem. This boundary value is imposed around the edge of the surface patches known as boundary conditions. To represent important features of human face, data in 3D FACS database were extracted to get boundary curves, which then determine PDE boundary condition.

In fact, a new approach based on facial anatomy for extracting the curves have been formulated. This formulation represents the geometric aspect of facial muscle distribution and the movement of the muscles of the face. The movement of facial muscles is determined by the information encoded by FACS. As the key player that play a prominent role, integrating boundary curves and FACS provides promising results for modelling and animation of facial expressions.

To promote storage efficiency, boundary curves are converted into Fourier coefficients in the form of matrix, M_f . This form of matrix denotes a full PDE-face in natural pose that is generated by four sets of consecutive boundary curves producing continuous facial surface patches. The application of M_f allows for a very straightforward reconstruction of PDE model, by the same token provide PDE face resolution flexibility.

PDE descriptors in a form of matrix, M_i , stores the differences between the neutral pose of the face and a given action unit. In creating a PDE face with a given action unit, the PDE descriptors is added to the matrix of the neutral coefficients, M_f . The utilization of this Fourier coefficients matrices leads to swift and accurate generation of a PDE face.

The work in this thesis has successfully modelled four different basic expressions namely happy, sad, fear and disgust. In modelling the PDE face with facial expressions, the PDE descriptors manipulates the neutral PDE face to generate any given basic expressions. A different combination of action unit will generate a different facial expression.

The blendshapes animation technique is used for animating the PDE face with facial expressions. As the PDE face with basic expressions has been generated in the modelling stage, all generic PDE faces are used for animation purposes. Applying this technique, a base face and the target faces are reused for time efficiency, without the need to manually adjust the vertices for making faces with expressions as implemented in the traditional blendshapes approach. The animation that has been carried out in this work gives an acceptable results, where the slider allows the transformation of the face expressions (happy, sad, fear and disgust) from a neutral face based on the weight of the blendshapes.

7.2 Contributions

The contributions that have been achieved with this research are given as follows:

- **Facial features extraction based on anatomy of human face.**

The PDE method that has been applied in this work is based on the anatomy of a human face when extracting the boundary curves, where 19 important facial features are taken into account during the extraction procedure.

- **New technique for texturing the PDE-based face.**

The texturing technique proposed in this work is very simple and gives realistic results. This work utilised the information taken from 3D FACS database such as vertices position, vertices normal and vertices texture coordinates that are given in OBJ files of face geometry.

- **PDE descriptors for facial movement.**

In this research, facial movements in modelling procedures are coded using FACS that uses action units as its measurement unit. The chosen boundary curves are based on action unit muscles, where the boundary curves are placed in areas that correspond to the action unit movement. PDE descriptor gives advantages in terms of storage and time efficiency in modelling and animating facial expressions.

- **Combination of PDE descriptors for facial expressions.**

A combination of action units has generated four different basic expressions namely happy, sad, fear and disgust. Thus, this work has successfully utilised the PDE descriptors combination for making a PDE face with expressions.

- **Blendshapes as the technique for animating a PDE-based face.**

Since the PDE-based face are compatible with blendshapes animation, this method might give solution to animators, to eliminate tedious work when generating a face with basic expressions.

7.3 Research limitations

There are several limitations of the study that are highlighted as follows:

- **Limited action unit data.**

All work in this thesis uses the data from a 3D FACS database. A limited number of action units are generated based on the facial data that is available from the database. However, it is sufficient to demonstrate that the proposed method is able to model any given action unit taken from the database in an accurate and realistic way.

- **Limited to model the PDE face with mouth closed.**

As mentioned in Section 5.6, the work in this thesis is not considering the action unit for mouth opening, as all the surface patch is continuous from chin to head. However, most of the action unit muscles are related to the upper part of the face for describing facial expressions compared to the lower part of the face. Hence, lack of action unit modelling on the mouth area is not given significant effect to the resulting PDE-based face using the proposed method when modelling and animating the facial expressions.

- **Limited to animate the PDE face from neutral face to expressive face.**

As mention in Section 6.4, the blendshapes animation generated in this work is only able to animate the PDE face from neutral to a face with expression but is unable to animate blendshapes from a face with an expression to another expression. This type of problem is known as blendshape interference and is common problem in blendshape animation techniques when blendshape targets have overlapping effects.

7.4 Future works

The limitations previously discussed can be improved in the future with regard to continuation of this research domain, in particular, facial action unit modelling using the PDE method. Some works that can be further extended from this research are outlined as follows:

- **Source of data.**

Instead of using a data form database, an attempt can be made by capturing the facial expressions in real time, such as performance-driven or facial motion data driven and adopting the PDE method for modelling of facial expressions.

- **PDE face with mouth open.**

Further investigation is proposed to generate a PDE-based face that can animate the opening and closing of the mouth in a smooth way. In order to achieve the proposed work, one can choose to integrate the modelling face using a combination of two popular parametric models, MPEG-4 Facial Animation Parameters (FAPs) and FACS. FAPs can be

used for defining the facial parameter around the mouth and lower part of the face, while FACS is used for defining the muscles in the upper part of the face.

- **Eliminate blendshape interference.**

As blendshape interference is common among the animators who used blendshape techniques in their animations, most of the animators usually manually tuning the shape of the polygon to get the desired expression. However, the work in [107] has segmented the face into smaller regions to allow local control of the blendshapes by physically-motivated segmentation. This physically-motivated segmentation as proposed in [107] can also be adopted to PDE-base facial modelling by identifying boundary curves correspondingly for segmentation of the face region.

Bibliography

- [1] S. Coquillart, "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *Comput. Graph. (ACM)*., vol. 24, no. 4, pp. 187–196, 1990.
- [2] G. Farin, *Curves and surfaces for CAGD, a practical design*, Fifth. Morgan Kaufman, San Diego, CA, 2002.
- [3] K. Gorowara, "On Bezier curves and surfaces," *Aerosp. Electron. Conf. 1988. NAECON 1988., Proc. IEEE 1988 Natl.*, pp. 754–756, 1988.
- [4] C. A. D. Cam, D. F. Rogers, S. G. Satterfield, and F. A. Rodriguez, "Ship Hulls, B-Spline Surfaces, and CAD/CAM," *Comput. Graph. Appl. IEEE*, vol. 3, no. 9, pp. 37–45, 1983.
- [5] L. Piegl, "On NURBS: a survey," *IEEE Comput. Graph. Appl.*, no. August 1981, 1991.
- [6] M. Athanasopoulos, H. Ugail, and G. G. Castro, "Parametric design of aircraft geometry using partial differential equations," *Adv. Eng. Softw.*, vol. 40, no. 7, pp. 479–486, Jul. 2009.
- [7] M. I. G. Bloor and M. J. Wilson, "Generating blend surfaces using partial differential equations," *Comput. Aided Des.*, vol. 21, no. 3, pp. 165–171, 1989.
- [8] M. I. G. Bloor and M. J. Wilson, "Using partial differential equations to generate free form surfaces," *Comput. Aided Des.*, vol. 22, pp. 341–212, 1990.
- [9] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, "Synthesizing Realistic Facial Expressions from Photographs," in *SigGraph 98 Conference Proceeding*, 1998, pp. 75–84.
- [10] F. I. Parke, "Computer generated animation of Faces," in *ACM '72 Proceedings of the ACM annual conference - Volume 1*, 1972, pp. 451–457.
- [11] M. Radovan and L. Pretorius, "Facial animation in a nutshell: past,

present and future,” *Proc. SAICST 2006*, pp. 71–79, 2006.

- [12] Y. Sun, J. Dong, M. Jian, and L. Qi, “Fast 3D face reconstruction based on uncalibrated photometric stereo,” *Multimed. Tools Appl.*, vol. 74, no. 11, pp. 3635–3650, 2015.
- [13] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin, “Static and dynamic 3D facial expression recognition: A comprehensive survey,” *Image Vis. Comput.*, vol. 30, no. 10, pp. 683–697, Oct. 2012.
- [14] A. Wojdel and L. Rothkrantz, “Automatic assessment of facial expressions using fiducial points,” *Proc. 13th Int. Conf. Comput. Syst. Technol. - CompSysTech '12*, pp. 290–297, 2012.
- [15] A. S. Agianpuye and J.-L. Minoi, “Synthesizing Neutral Facial Expression on 3D Faces using Active Shape Models,” in *IEEE Regent 10 Symposium*, 2014, pp. 593–598.
- [16] Y. Bouzid, O. El Ghoul, and M. Jemni, “Synthesizing Facial Expressions for Signing Avatars using MPEG4 Feature Points,” in *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*, 2013, pp. 1–6.
- [17] A. Wojdel and L. J. M. Rothkrantz, “Parametric Generation of Facial Expressions Based on FACS,” *Comput. Graph. Forum*, vol. 24, no. 4, pp. 743–757, Dec. 2005.
- [18] L. Zalewski and S. Gong, “2D statistical models of facial expressions for realistic 3D avatar animation,” *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 217–222, 2005.
- [19] W. Mattheyses and W. Verhelst, “Audiovisual speech synthesis: An overview of the state-of-the-art,” *Speech Commun.*, vol. 66, pp. 182–217, 2015.
- [20] D. DeCarlo, D. Metaxas, and M. Stone, “An anthropometric face model using variational techniques,” in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 67–74.
- [21] M. Hoch, G. Fleischmann, and B. Girod, “Modeling and animation of facial expressions based on B-Splines,” *Vis. Comput.*, vol. 11, no. 2, pp. 87–95, 1994.

- [22] C.-L. Huang and Y.-M. Huang, "Facial Expression Recognition Using Model-Based Feature Extraction and Action Parameters Classification," *J. Vis. Commun. Image Represent.*, vol. 8, no. 3, pp. 278–290, 1997.
- [23] N. Patel and M. Zaveri, "3D facial model reconstruction, expressions synthesis and animation using single frontal face image," *Signal, Image Video Process.*, vol. 7, no. 5, pp. 889–897, 2013.
- [24] P. S. Aleksic and a. K. Katsaggelos, "Automatic facial expression recognition using facial animation parameters and multistream HMMs," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 1, pp. 3–11, 2006.
- [25] Y. Sheng, A. H. Sadka, and A. M. Kondoz, "Automatic single view-based 3-D face synthesis for unsupervised multimedia applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 961–974, 2008.
- [26] M. H. Alkawaz and A. H. Basori, "The Effect of Emotional Colour on Creating Realistic Expression of Avatar," in *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2012, vol. 1, no. 212, pp. 143–152.
- [27] S. Villagrasa and A. S. Sánchez, "Face! 3d facial animation system based on faces," *IV Ibero-American Symp. Com- Puter Graph.*, 2009.
- [28] V. Orvalho, P. Bastos, F. Parke, B. Oliveira, and X. Alvarez, "A Facial Rigging Survey," *Proc. 33rd Annu. Conf. Eur. Assoc. Comput. Graph. - Eurographics*, vol. 51, pp. 10–32, 2012.
- [29] M. H. Alkawaz, D. Mohamad, A. H. Basori, and T. Saba, "Blend Shape Interpolation and FACS for Realistic Avatar," *3D Res.*, vol. 6, no. 6, pp. 1–10, 2015.
- [30] F. I. Parke and K. Waters, *Computer Facial Animation*, 2nd ed. Wellesley, Massachusetts, 1996.
- [31] K. Waters, "A muscle model for animation three-dimensional facial expression," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 17–24, 1987.
- [32] W. Lee, J. Gu, and N. Magnenat-Thalmann, "Generating Animatable 3D

Virtual Humans from Photographs,” *Comput. Graph. Forum*, vol. 19, no. 3, pp. 1–10, Sep. 2000.

- [33] N. Ersotelos and F. Dong, “Building highly realistic facial modeling and animation: A survey,” *Vis. Comput.*, vol. 24, no. 1, pp. 13–30, 2008.
- [34] T. Weise, S. Bouaziz, H. Li, and M. Pauly, “Realtime performance-based facial animation,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 77(1–9), 2011.
- [35] M. H. Alkawaz, D. Mohamad, and A. H. Basori, “A Crucial Investigation of Facial Skin Colour Research Trend and Direction,” *Int. J. Multimed. Ubiquitous Eng.*, vol. 10, no. 1, pp. 295–316, 2015.
- [36] H. Yu and H. Liu, “Regression-based facial expression optimization,” *IEEE Trans. Human-Machine Syst.*, vol. 44, no. 3, pp. 386–394, 2014.
- [37] Y. Zhang, E. C. Prakash, and E. Sung, “Modeling and animation of individualized faces for 3D facial expression synthesis,” *Int. J. Imaging Syst. Technol.*, vol. 13, no. 1, pp. 42–64, Jun. 2003.
- [38] F. I. Parke, “Parameterized Models for Facial Animation,” *IEEE Comput. Graph. Appl.*, vol. 2, no. 9, pp. 61–68, 1982.
- [39] L. Li, Y. Liu, and H. Zhang, “A survey of computer facial animation techniques,” in *Proceedings - 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012*, 2012, vol. 3, no. 60675008, pp. 434–438.
- [40] T. Goto, M. Escher, C. Zanardi, and N. Magnenat-Thalmann, “MPEG-4 based animation with face feature tracking,” in *Eurographics Workshop on Computer Animation and Simulation 1999*, 1999, pp. 89–98.
- [41] Z. Li, J. Ma, and H. Feng, “Improved Radial Basis Function Based Parameterization for Facial Expression Animation,” in *The 9th International Conference on Computer Science & Education, ICCSE 2014*, 2014, pp. 353–358.
- [42] H. Y. Ping, L. N. Abdullah, P. S. Sulaiman, and A. A. Halin, “Computer Facial Animation: A Review,” *Int. J. Comput. Theory Eng.*, vol. 5, no. 4, pp. 658–662, 2013.
- [43] H. H. S. Ip and C. S. Chan, “Script-based facial gesture and speech

animation using a nurbs based face model,” *Comput. Graph.*, vol. 20, no. 6 SPEC. ISS., pp. 881–891, 1996.

- [44] Y. Zhang, E. C. Prakash, and E. Sung, “Real-time physically-based facial expression animation using mass-spring system,” in *Computer Graphics International 2001 Proceedings*, 2001, pp. 347–350.
- [45] M. H. Alkawaz, D. Mohamad, A. Rehman, and A. H. Basori, “Facial Animations: Future Research Directions & Challenges,” *3D Res.*, vol. 5, no. 2, p. 12, May 2014.
- [46] E. Elyan and H. Ugail, “Reconstruction of 3D human facial images using partial differential equations,” *J. Comput.*, vol. 2, no. 8, pp. 1–8, 2007.
- [47] Y. Sheng, P. Willis, G. G. Castro, and H. Ugail, “Facial geometry parameterisation based on Partial Differential Equations,” *Math. Comput. Model.*, vol. 54, no. 5–6, pp. 1536–1548, Sep. 2011.
- [48] J. Noh and U. Neumann, “A survey of facial modeling and animation techniques,” in *USC Technical Report*, 1998, pp. 99–705.
- [49] F. Pighin, R. Szeliski, and D. Salesin, “Modeling and animating realistic faces from images,” *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 143–169, 2002.
- [50] Y. Li, S. Mohammad Mavadati, M. H. Mahoor, and Q. Ji, “A unified probabilistic framework for measuring the intensity of spontaneous facial action units,” in *10th IEEE International Conference and Workshop in Automatic Face and Gesture Recognition (FG), 2013*, 2013, pp. 1–7.
- [51] Y. Li, J. Chen, Y. Zhao, and Q. Ji, “Data-free prior model for facial action unit recognition,” *IEEE Trans. Affect. Comput.*, vol. 4, no. 2, pp. 127–141, 2013.
- [52] Y. Tong, W. Liao, and Q. Ji, “Facial action unit recognition by exploiting their dynamic and semantic relationships,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1683–1699, 2007.
- [53] M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Measuring facial expressions by computer image analysis,” *Psychophysiology*, vol. 36, no. 2, pp. 253–263, 1999.

- [54] L. Y. L. Yin, X. C. X. Chen, Y. S. Y. Sun, T. Worm, and M. Reale, "A high-resolution 3D dynamic facial expression database," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008, pp. 1–6.
- [55] G. Donato, M. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying Facial Actions," *Pattern Anal. ...*, vol. 21, no. 1, pp. 974–989, 1999.
- [56] R. Amini and C. Lisetti, "HapFACS: An Open Source API/Software to Generate FACS-Based Expressions for ECAs Animation and for Corpus Generation," *2013 Hum. Assoc. Conf. Affect. Comput. Intell. Interact.*, pp. 270–275, Sep. 2013.
- [57] P. Ekman, W. V Friesen, and J. C. Hager, *Facial Action Coding System*, Second Edi. Salt Lake City: Research Nexus eBook, 2002.
- [58] N. Patel and Zaveri, "Parametric facial expression synthesis and animation," *Int. J. Comput. Appl.*, vol. 3, no. 4, pp. 34–40, 2010.
- [59] Z. Deng, P.-Y. Chiang, P. Fox, and U. Neumann, "Animating blendshape faces by cross-mapping motion capture data," in *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, 2006, vol. 1, no. March, p. 43.
- [60] A. Wojdel and L. J. M. Rothkrantz, "Intelligent System for Semiautomatic Facial Animation," in *Euromedia 2000*, 2000, pp. 133–137.
- [61] D. Terzopoulos and K. Waters, "Physically-based facial modeling, analysis , and animation," *J. Vis. an Comput. Animat.*, vol. 1, no. 2, pp. 73–80, 1990.
- [62] Z. Deng and U. Neumann, "Computer facial animation: A survey," in *Data-Driven 3D Facial Animation*, 2007, pp. 1–28.
- [63] S. Tang, H. Yan, and A. W.-C. Liew, "A NURBS-based vector muscle model for generating human facial expression," in *Proceeding of the 2003 joint conference of the Fourth International Conference on Information, Communications and Signal Processing*, 2003, pp. 758–762.
- [64] M. Platt and I. Badler, "Animating Facial Expressions," *Comput. Graph. (ACM)*, vol. 15, no. 3, pp. 245–252, 1981.

- [65] P. Havaladar, "Performance Driven Facial Animation," in *ACM SIGGRAPH 2006*, 2006, no. c, pp. 1–20.
- [66] J. 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.
- [67] X. Liu, T. Mao, S. Xia, Y. Yu, and Z. Wang, "Facial animation by optimized blendshapes from motion capture data," *Comput. Animat. Virtual Worlds*, vol. 19, no. August, pp. 235–245, 2008.
- [68] J. P. Lewis, J. Mooser, Z. Deng, and U. Neumann, "Reducing blendshape interference by selected motion attenuation," *Proc. 2005 Symp. Interact. 3D Graph. games - SI3D '05*, p. 25, 2005.
- [69] Q. Li and Z. Deng, "Orthogonal-blendshape-based editing system for facial motion capture data," *IEEE Comput. Graph. Appl.*, vol. 28, no. 6, pp. 76–82, 2008.
- [70] R. J. Balsys and K. G. Suffern, "Point based rendering of non-manifold surfaces with contours," in *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia (GRAPHITE '04)*, 2004, pp. 7–14.
- [71] P. . Chivate and A. . Jablokow, "Review of surface representations and fitting for reverse engineering," *Comput. Integr. Manuf. Syst.*, vol. 8, no. 3, pp. 193–204, 1995.
- [72] M. S. Muhammad, M. T. Mahamood, and T.-S. C. T.-S. Choi, "Approximating 3D shape using Bezier surface," *2009 IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 753–756, 2009.
- [73] C. W. Dekanski, "Design and analysis of propeller blade geometry using the PDE method," The University of Leeds, 1993.
- [74] E. Dimas and D. Briassoulis, "3D geometric modelling based on NURBS: a review," *Adv. Eng. Softw.*, vol. 30, pp. 741–751, 1999.
- [75] D. Huang and H. Yan, "Modeling and animation of human expressions using NURBS curves based on facial anatomy," *Signal Process. Image Commun.*, vol. 17, no. 2002, pp. 457–465, 2002.

- [76] M. Bloor and M. Wilson, "Generating parametrizations of wing geometries using partial differential equations," *Comput. methods Appl. Mech. ...*, vol. 148, pp. 125–138, 1997.
- [77] H. Ugail, M. I. G. Bloor, and M. J. Wilson, "Techniques for interactive design using the PDE method," *ACM Trans. Graph.*, vol. 18, no. 2, pp. 195–212, Apr. 1999.
- [78] J. J. Zhang and L. You, "PDE based surface representation—vase design," *Comput. Graph.*, vol. 26, no. 1, pp. 89–98, Feb. 2002.
- [79] N. Ahmat, H. Ugail, and G. G. Castro, "Method of modelling the compaction behaviour of cylindrical pharmaceutical tablets.," *Int. J. Pharm.*, vol. 405, no. 1–2, pp. 113–21, Feb. 2011.
- [80] H. Ugail, M. Robinson, M. G. Bloor, and M. Wilson, "Interactive Design of Complex Mechanical Parts using a Parametric Representation," in *The Mathematics of Surfaces IX SE - 11*, R. Cipolla and R. Martin, Eds. Springer London, 2000, pp. 169–179.
- [81] J. J. Zhang and L. You, "Surface representation using second, fourth, and mixed order partial differential equations," in *SMI '01 Proceedings of the International Conference on Shape Modelling & Applications*, 2001, p. 250.
- [82] G. G. Castro, H. Ugail, P. Willis, and I. Palmer, "A survey of partial differential equations in geometric design," *Vis. Comput.*, vol. 24, pp. 213–225, 2008.
- [83] M. I. G. Bloor and M. J. Wilson, "Spectral approximations surfaces to PDE," vol. 28, no. 2, pp. 145–152, 1996.
- [84] E. Elyan and H. Ugail, "Interactive surface design and manipulation using PDE-Method through Autodesk Maya Plug-in," in *International Conference on Cyber World*, 2009, pp. 119–125.
- [85] Y. Sheng, P. Willis, G. Castro, and H. Ugail, "PDE face: A novel 3d face model," *Proc. 8th IASTED Int. ...*, pp. 408–415, 2008.
- [86] H. Ugail, *Partial differential equation for geometric design*. Springer London Dordrecht Heidelberg New York, 2011.

- [87] L. H. You, P. Comninou, and J. J. Zhang, "PDE blending surfaces with C2 continuity," *Comput. Graph.*, vol. 28, no. 6, pp. 895–906, Dec. 2004.
- [88] G. G. Castro, H. Ugail, P. Willis, and Y. Sheng, "Parametric representation of facial expressions on PDE-based surfaces," in *Proc. of the 8th IASTED International Conference Visualization, Imaging, and Image Processing (VIIP 2008)*, 2008, pp. 402–407.
- [89] Y. Sheng, P. Willis, G. G. Castro, and H. Ugail, "PDE-Based Facial Animation: Making the Complex Simple," in *Lecture notes in Computer Science*, 2008, pp. 723–732.
- [90] D. Huang and H. Yan, "NURBS curve controlled modelling for facial animation," *Comput. Graph.*, vol. 27, no. 3, pp. 373–385, Jun. 2003.
- [91] S. Tang, A. Liew, and H. Yan, "Lip-sync in human face animation based on video analysis and spline models," in *Proceeding of the 10th International Multimedia Modelling Conference (MMM'04)*, 2004.
- [92] D. Cosker, E. Krumhuber, and A. Hilton, "A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling," *2011 Int. Conf. Comput. Vis.*, pp. 2296–2303, Nov. 2011.
- [93] S. K. Datta, P. Morrow, B. Scotney, and P. Morrow, "Facial Feature Extraction Using a 4D Stereo Camera System," *Soft Comput. Tech. Vis. Sci.*, vol. 395, pp. 209–218, 2012.
- [94] Y. Isoda, A. Tsukamoto, Y. Kosaka, T. Okumura, M. Sawai, K. Yano, S. Nakata, and S. Tanaka, "Reconstruction of Kyoto of the Edo era based on arts and historical documents: 3D urban model based on historical GIS data," *Int. J. Humanit. Arts Comput.*, vol. 3, no. 1–2, pp. 21–38, 2009.
- [95] D. Cosker, E. Krumhuber, and A. Hilton, "Perception of linear and nonlinear motion properties using a FACS validated 3D facial model," ... *Symp. Appl. Percept. ...*, vol. 1, no. 212, pp. 101–108, 2010.
- [96] M. Pantic and L. J. M. Rothkrantz, "Facial Action Recognition for Facial Expression Analysis From Static Face Images," *Ieee Trans. Syst. Man, Cybern. B Cybern.*, vol. 34, no. 3, pp. 1449–1461, 2004.

- [97] B. Ma, J. Yao, R. Yan, and B. Zhang, "The Improvement of Parameterized Face Model of Candide Based on MPEG-4 and FACS," *Int. J. Electr. Energy*, vol. 2, no. 2, pp. 99–102, 2014.
- [98] N. Patel and M. Zaveri, "3D facial model construction and expressions synthesis from a single frontal face image," *2010 Int. Conf. Comput. Commun. Technol. ICCCT-2010*, vol. 1, no. 1, pp. 652–657, 2010.
- [99] Y. Zhang and Q. Ji, "Active and dynamic information fusion for facial expression understanding from image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 699–714, 2005.
- [100] J. Spencer-smith, H. Wild, A. . Innes-ker, J. Townsend, C. Duffy, C. Edwards, K. Ervin, N. Merritt, and J. W. Paik, "Making faces: Creating three-dimensional parameterized models of facial expression," *Behav. Res. Methods, Instruments, Comput.*, vol. 33, no. 2, pp. 115–123, 2001.
- [101] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade dataset (CK+): A complete facial expression dataset for action unit and emotion specified expression," *IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 94–101, 2010.
- [102] R. E. Jack, O. G. B. Garrod, and P. G. Schyns, "Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time," *Curr. Biol.*, vol. 24, no. 2, pp. 187–192, 2014.
- [103] Y.-L. T. Y.-L. Tian, T. Kanada, and J. F. Cohn, "Recognizing upper face action units for facial expression analysis," in *IEEE Conference on Pattern Analysis and Machine Learning*, 2001, vol. 23, no. 2, pp. 97–115.
- [104] Y. Xu, A. W. Feng, S. Marsella, and A. Shapiro, "A Practical and Configurable Lip Sync Method for Games," *Proc. Motion Games - MIG '13*, pp. 131–140, 2013.
- [105] J. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng, "Practice and Theory of Blendshape Facial Models," *Eurographics 2014 - State Art Reports*, pp. 199–218, 2014.
- [106] J. P. Lewis and K. Anjyo, "Direct Manipulation Blendshapes," *Comput. Graph. Appl. IEEE*, vol. 30, no. 4, pp. 42–50, 2010.

- [107] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin, "Learning controls for blend shape based realistic facial animation," in *Proceedings of the SIGGRAPH 2003 conference on Sketches & applications in conjunction with the 30th annual conference on Computer graphics and interactive techniques - GRAPH '03*, 2003, p. 1.
- [108] Y. Seol, J. P. Lewis, J. Seo, B. Choi, K. Anjyo, and J. Noh, "Spacetime expression cloning for blendshapes," *ACM Trans. Graph.*, vol. 31, no. 2, pp. 1–12, 2012.

Appendix A

Data information of neutral face

Curve No	d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	b_0	b_1	b_2	b_3	b_4	b_5	b_6
1	0.03809274	0.1663846	0.05544525	0.00138079	0.00770672	0.01004672	0.126702	-0.01454962	0.0116259	0.0150041	0.00144417						
2	0.03777644	0.1698515	0.05616538	0.00043981	0.00905314	0.01027554	0.1285475	-0.01483806	0.00218215	0.01243211	0.00114658						
3	0.03756127	0.1731145	0.05623917	0.00049134	0.0089619	0.0102619	0.1297393	-0.0146828	0.00351547	0.01308741	0.00085239						
4	0.03699722	0.1759176	0.05536077	-9.37E-05	0.00891191	0.01002853	0.131346	-0.01400663	0.00441741	0.01302471	0.00070022						
5	0.03674003	0.1786196	0.05253005	-0.00098093	0.00970026	0.01021036	0.1365744	-0.01638211	0.00552938	0.0125875	0.0001245						
6	0.03608103	0.1825265	0.05212323	-0.00177231	0.00938047	0.00891772	0.1389372	-0.01755325	0.00922956	0.01189116	0.00055866						
7	0.03541613	0.1804375	0.05769901	-0.00193459	0.01213851	0.00620381	0.1362603	-0.01929088	0.0107471	0.01533902	0.00061233						
8	0.03375767	0.1746368	0.06348655	-0.00078156	0.01549784	0.00514351	0.1302718	-0.022261224	0.01005314	0.01764335	0.00213483						
9	0.03373364	0.1672488	0.06810722	-0.00115381	0.01494182	0.00432849	0.1254033	-0.0242054	0.00911411	0.01834665	-0.00017746						
10	0.0321284	0.1574537	0.07216101	0.00011116	0.01746673	0.00574719	0.117888	-0.02774804	0.00439148	0.01921304	4.74E-05						
11	0.03261713	0.1534894	0.07328939	-0.00033872	0.01585708	0.00517195	0.1148823	-0.02604791	0.00207268	0.01973174	-0.00093755						
12	0.03256914	0.1601436	0.06886216	-0.00040323	0.01436872	0.00669364	0.1187354	-0.02376434	0.00281148	0.01766501	-0.00099978						
13	0.03223198	0.1645422	0.06609114	0.00118804	0.0157	0.00906569	0.1214727	-0.02365186	0.00129687	0.0192291	-0.00075211						
14	0.03209821	0.1729878	0.06405623	0.00103302	0.0173764	0.0087216	0.126645	-0.02256194	0.00343005	0.02100036	-8.99E-05						
15	0.03186093	0.179219	0.06176132	-0.00030281	0.01771661	0.00826432	0.1309007	-0.02159838	0.00526835	0.02080603	-0.0006482						
16	0.03172289	0.183953	0.06027266	8.96E-05	0.0170252	0.00696033	0.1330163	-0.02050548	0.00804976	0.02029626	0.00046344						
17	0.0315195	0.1852569	0.05880916	-0.00051195	0.01604087	0.00587926	0.1330936	-0.01949699	0.00816457	0.01934514	0.00011278						
18	0.03103765	0.1855724	0.05577637	-0.00064929	0.01594886	0.00483688	0.1340397	-0.01917649	0.0067273	0.01851129	-3.06E-06						
19	0.02982532	0.1839316	0.05498653	-4.91E-05	0.01390081	0.00542263	0.131658	-0.01968793	0.00706556	0.01622968	0.00025506						
20	0.02879565	0.1809408	0.05484259	-0.0002719	0.01184667	0.00646138	0.1288199	-0.00721685	0.00721685	0.01351845	0.00045062						
21	0.02849503	0.1780785	0.05210769	-0.0002156	0.01178139	0.00737627	0.1273465	-0.00355736	0.00355736	0.01217788	7.37E-05						
22	0.02650806	0.1760723	0.04923287	0.00078967	0.0113604	0.00756306	0.1246746	-0.00021799	-0.00021799	0.01239228	-0.0011482						
23	0.02575089	0.1720069	0.04790263	0.0005296	0.01123162	0.00843665	0.1221516	-0.02081863	-0.00336572	0.01057788	-0.00114966						
24	0.02426033	0.1702445	0.04698459	0.00152947	0.01018125	0.00846064	0.1196117	-0.0214051	-0.00488541	0.01048066	-0.00182752						
25	0.02327249	0.1656954	0.04987682	0.00081519	0.00916547	0.00734838	0.116495	-0.02384022	-0.0029475	0.00919658	-0.00169099						
26	0.02324222	0.1621652	0.04935146	0.0012848	0.01010008	0.0071187	0.1144833	-0.02409419	-0.00599872	0.00972045	-0.00201079						
27	0.02166948	0.1582536	0.04878881	0.00138106	0.00838015	0.00914874	0.110146	-0.02430101	-0.00780677	0.0067818	-0.00194129						
28	0.0216373	0.1526258	0.05029868	0.00157883	0.00670974	0.00900853	0.1059132	-0.02421122	-0.0091458	0.00639458	-0.00232168						
29	0.01831295	0.1340569	0.03304948	-0.00130602	0.01030104	0.00778472	0.1009026	-0.01387224	0.002655216	0.01174878	-0.00181122						
30	0.01746629	0.13248	0.02880279	-0.0002592	0.00786339	0.00735539	0.09804223	-0.01166548	6.86E-05	0.00920522	-0.00144064						
31	0.01684559	0.123997	0.02816799	5.02E-05	0.00595365	0.00785317	0.09172541	-0.01187879	-0.00290637	0.00563363	0.00019694						

Appendix B

PDE descriptor for AU1

Curve No	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	
1	0.00877651	0.0083569	-0.00381557	-0.00192839	0.001407635	-0.00055185	-0.0107929	-0.0074059	-0.00022764	-0.00164024	-0.00346914	-0.00022764	-0.00022764	-0.00022764	-0.00022764	-0.00022764	-0.00022764	-0.00022764	-0.00022764	-0.00022764	-0.00022764
2	0.00931737	0.0089298	-0.00418992	-0.00188122	0.001726603	-0.00082591	-0.0109827	-0.00617936	-0.00034763	-0.00200352	-0.00358127	-0.00034763	-0.00034763	-0.00034763	-0.00034763	-0.00034763	-0.00034763	-0.00034763	-0.00034763	-0.00034763	-0.00034763
3	0.00908067	0.0095674	-0.00366714	-0.00223611	0.001380647	-0.00095697	-0.0107718	-0.00679543	0.001675283	-0.000312904	-0.00365603	-0.000679543	-0.000679543	-0.000679543	-0.000679543	-0.000679543	-0.000679543	-0.000679543	-0.000679543	-0.000679543	-0.000679543
4	0.00910896	0.0088962	-0.00431502	-0.00151652	0.001489765	-0.00145402	-0.0113127	-0.00680261	0.000865723	-0.000282234	-0.00350675	-0.000865723	-0.000865723	-0.000865723	-0.000865723	-0.000865723	-0.000865723	-0.000865723	-0.000865723	-0.000865723	-0.000865723
5	0.00873147	0.0094834	-0.00411651	-0.00179345	0.002446736	-0.00215232	-0.0119275	-0.00650017	0.00107577	-0.000262681	-0.00360787	-0.000650017	-0.000650017	-0.000650017	-0.000650017	-0.000650017	-0.000650017	-0.000650017	-0.000650017	-0.000650017	-0.000650017
6	0.00886105	0.0085049	-0.00398087	-0.00169289	0.002083498	-0.00201346	-0.0123231	-0.00674094	0.001106505	-0.0002816	-0.00376668	-0.000674094	-0.000674094	-0.000674094	-0.000674094	-0.000674094	-0.000674094	-0.000674094	-0.000674094	-0.000674094	-0.000674094
7	0.00881	0.0059968	-0.00229093	-0.00125616	0.00336172	-0.00215484	-0.0139646	-0.00773052	-0.00012506	-0.00127706	-0.00431321	-0.000773052	-0.000773052	-0.000773052	-0.000773052	-0.000773052	-0.000773052	-0.000773052	-0.000773052	-0.000773052	-0.000773052
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix C

Pseudo code for texturing

START

Count number of vertices in PDE mesh
Count number of vertices in original mesh

FOR all vertices in PDE mesh
 Find the smallest distance to the vertex in original mesh
 Store the index of vertex in an array
ENDFOR

Open a new blank file

FOR all element in index array
 Read vertex texture coordinate of original mesh
 Write the value to file
ENDFOR

FOR all element in index array
 Read vertex texture coordinate of original mesh
 Write the value to file
ENDFOR

Close the opened file

FINISH

Appendix D

An algorithm for extracting a boundary curve using MEL script

```
*****
int $i, $j, $MeshCount, $PointCount, $IndexMin[];
vector $Vmesh, $VCal, $VCurve1, $first, $distance, $CV;
float $small, $Dis[], $Min[];

//Define path
string $filePath = "/...path.../1.obj";

//Open file to write an OBJ file
$file = `fopen $filePath "w"`;

$PointCount=`polyEvaluate -v curve1`;
$MeshCount=`polyEvaluate -v AU1:Mesh`;

for($j=0; $j<$PointCount; $j++)
{
    $VCurve1 = `pointPosition curve1.cv[$j]`;
    $small = 1;

    for($i=0; $i<$MeshCount; $i++)
    {
        $VCal = `pointPosition AU1:Mesh.vtx[$i]`;
        $distance = $VCal - $VCurve1;
        $Dis[$i] = `mag <<$distance.x,$distance.y,$distance.z>>`;

        if ($Dis[$i]<0.05)
        {
            if($Dis[$i]<$small)
            {
                $small = $Dis[$i];
                $Min[$j]= $small;
                $IndexMin[$j] = $i;
            }

            }//end if small

        }//end if tolerance

    }//end for reading each points in mesh

    $CV = `pointPosition AU1:Mesh.vtx[$IndexMin[$j]]`;
    fprintf $file ("v "+$CV.x+" "+$CV.y+" "+$CV.z+"\n");

}

}

//end for each points in a boundary curve

//close an OBJ file
fclose $file;

*****
```