Method of Modelling Facial Action Units Using Partial Differential Equations

Hassan Ugail and Nur Baini Ismail

Abstract In this paper we discuss a novel method of mathematically modelling facial action units for accurate representation of human facial expressions in 3-dimensions. Our method utilizes the approach of Facial Action Coding System (FACS). It is based on a boundary-value approach, which utilizes a solution to a fourth order elliptic Partial Differential Equation (PDE) subject to a suitable set of boundary conditions. Here the PDE surface generation method for human facial expressions is utilized in order to generate a wide variety of facial expressions in an efficient and realistic way. For this purpose, we identify a set of boundary curves corresponding to the key features of the face which in turn define a given facial expression in 3-dimensions. The action units (AUs) relating to the FACS are then efficiently represented in terms of Fourier coefficients relating to the boundary curves which enables us to store both the face and the facial expressions in an efficient way.

1 Introduction

Human faces can be represented using both two-dimensional (2D) and three-dimensional (3D) geometry. It is true to say that in computer graphics, facial modelling and animation is one of the most challenging areas. This is due to the fact that the human face is an extremely complex geometric form [1–3].

The most common approaches in facial modelling and animation are geometric manipulation and those based on image manipulations [4]. Approaches based on geometric manipulations include key-frame expression interpolation [5], pseudo-muscle modelling including physics based modelling [6] and parametrisation techniques [7]. Image manipulations include image morphing between photographic images [8], texture manipulations [9] and image blending [10].

H. Ugail (✉) • N.B. Ismail
Centre for Visual Computing, School of Engineering and Informatics,
University of Bradford, Bradford BD7 1DP, UK
e-mail: h.ugail@bradford.ac.uk; N.B.Ismail@student.bradford.ac.uk

M. Kawulok et al. (eds.), *Advances in Face Detection and Facial Image Analysis*,
DOI 10.1007/978-3-319-25958-1_6

As stated previously, much work have been done in modelling human faces in 2D and 3D. One of the most popular models among 3D parametric face models is the 3D morphable model of Blanz and Vetter [11]. This parametric model has a low dimensional 3D face shape and texture. Further, the parametric description of the face is based on the principles that underlie in solving a mathematical optimization problem. Since Blanz and Vetter's pioneering work, a number of approaches have been adopted to develop sophisticated fitting algorithms of morphable models. For example, the work presented in [12] used a Lambert model for simplification of the optimization. Further work presented in [13] and [14] show improved optimization efficiency by using an input image. However, even with the existing methods, the iteration process for optimization in matching algorithms for morphable models is very time consuming and computational inefficient.

In computer animation, mainly six facial expressions are used for facial animation and facial simulations, i.e. anger, joy, fear, sad, disgust and surprise [15]. Further, there are some studies in this area whereby the Facial Action Coding Systems (FACS) based on the facial muscle movements are utilised e.g. [7, 16, 17].

Action units within the FACS are classified according to the location of the given muscle on the face and the type of action involved. Within the definitions of FACS the human facial muscles are divided into two parts namely those that are in the upper face and those that are in the lower face. The upper face comprises the muscles around eyebrows, forehead and eyelids. In contrast, muscles around the mouth and lips affect lower part of the face [16].

Although, FACS was initially designed for psychologists and behavioural scientists to understand facial expressions and behaviour, it has been recently adapted to visual communication, teleconferencing, computer graphics, etc. [18]. Thus, today FACS is widely acceptable for use in computer graphics and computer based simulations.

Conventional methods in Computer-Aided Design to generate parametric surface tend to be based on splines such as B-Splines and Non-Uniform Rational B-Splines (NURBS). Much work in modelling and animation of facial expressions has been based on B-Spline and NURBS. Hoch et al. [19], for example, used a canonical model for animating facial expressions by fitting splines to the scan data. They manually positioned the control points of the chosen splines on the geometry of the face accordingly so as to create the right action units. This process is usually quite tedious. Thus, it is not practical for complex facial animations. Meanwhile, Huang and Yan [20, 21] used NURBS for modelling and animating facial expressions by positioning a control polygon based on the facial anatomy. They used a fuzzy set formulation for simulating the features of facial muscles. Various expressions were simulated by changing the weights or positioning the control points. However, this approach requires a higher resolution face model for realistic animation which would in turn increase the computational cost.

The use of Partial Differential Equations (PDEs) for efficiently modelling human face and facial features have been discussed previously. PDEs can be used to generate complex geometry by posing the geometry generation as a mathematical boundary boundary-value problem. Due to the efficiency in parameterising complex geometry the use of PDEs for modelling faces is a plausible idea [2].

For example, the work in [22] used a 3D scan of a human face to efficiently represent and generate facial geometry. Later, the work in [23] applied simple mathematical transformations for generating PDE faces for various facial expressions. Given that the boundary conditions of the chosen PDE are used to describe the key feature points of the geometric object, it is thought that integrating FACS on these boundaries conditions would be a way forward for modelling facial expressions efficiently.

Thus, the aim of this work was to develop a PDE based mathematical formulation for representing action units for a human face. The approach we have adopted here for generating FACS is by mapping the boundary curves (in other words the boundary conditions of the chosen PDE) with key features of the face. There are many benefits in using a formulation of this nature. These include the generation of smooth PDE surface for the faces with associated FACS muscle structure, generation of facial expressions in real time as well as efficient storage of facial data.

The rest of the paper is organized as follows. In Sect. 2 we introduce the PDE method for surface generation. Then in Sect. 3, the process for modelling neutral face configuration using the PDE method is discussed. In Sect. 4, the methodology for generating PDE faces with given action units are discussed and this is illustrated with relevant examples. Results as well as the accuracy of the methodology we have presented in this paper are discussed in Sect. 5. Finally we conclude this work in Sect. 6.

2 The PDE Method for Parametric Surfaces

The PDE method for parametric surfaces is based upon by solving a suitably posed boundary-value problem where an elliptic PDE is solved subject to a given set of boundary conditions [2].

A PDE surface is a parametric surface patch $\underline{X}(u, v)$, defined as a function of two parameters u and v on a finite domain $\Omega \subset R^2$, by specifying boundary data around the edge region of $\partial\Omega$. Typically the boundary data are specified in the form of $\underline{X}(u, v)$ and a number of its derivatives on $\partial\Omega$. Moreover, this approach regards the coordinates of the (u, v) point as a mapping from that point in Ω to a point in the physical space. To satisfy these requirements the surface $\underline{X}(u, v)$ is regarded as a solution of a PDE based on the bi-harmonic equation $\nabla^4 = 0$ namely,

$$\left(\frac{\partial^2}{\partial u^2} + a^2\frac{\partial^2}{\partial v^2}\right)^2 \underline{X}(u, v) = 0. \tag{1}$$

Here the boundary conditions on the function $\underline{X}(u, v)$ and its normal derivatives $\frac{\partial \underline{X}}{\partial n}$ are imposed at the edges of the surface patch. The parameter a (where $a \neq 0$) is a special design parameter which controls the relative smoothing of the surface in the u and v directions.

Note that the Eq. (1) is known as a form of the well known Biharmonic equation. The Biharmonic equation can model phenomena related to solid and fluid mechanics as well as stress/strain analysis problems in engineering analysis. Further, there are great many solution techniques to solve PDEs of the nature described in Eq. (1). These range from analytical to numerical methods [24–26].

2.1 Solution of the PDE

As stated above, there exist many methods to determine the solution of Eq. (1) ranging from analytic solution techniques to sophisticated numerical methods. For the work described here restricting to periodic boundary conditions a closed form analytic solution of Eq. (1) is utilised.

Choosing the parametric region to be $0 \leq u \leq 1$ and $0 \leq v \leq 2\pi$, the periodic boundary conditions can be expressed as, $\underline{X}(0, v) = \underline{P}_1(v)$, $\underline{X}(1, v) = \underline{P}_2(v)$, $\underline{X}_u(0, v) = \underline{d}_1(v)$ and $\underline{X}_u(1, v) = \underline{d}_2(v)$.

Note that the boundary conditions $\underline{P}_0(v)$ and $\underline{P}_1(v)$ define the edges of the surface patch at $u = 0$ and $u = 1$ respectively. Using the method of separation of variables, the analytic solution of Eq. (1) can be written as,

$$\underline{X}(u, v) = \underline{A}_0(u) + \sum_{n=1}^{\infty} [\underline{A}_n(u) \cos(nv) + \underline{B}_n(u) \sin(nv)], \tag{2}$$

where

$$\underline{A}_0 = \underline{a}_{00} + \underline{a}_{01} u + \underline{a}_{02} u^2 + \underline{a}_{03} u^3, \tag{3}$$

$$\underline{A}_n = \underline{a}_{n1} e^{anu} + \underline{a}_{n2} u e^{anu} + \underline{a}_{n3} e^{-anu} + \underline{a}_{n4} u e^{-anu}, \tag{4}$$

$$\underline{B}_n = \underline{b}_{n1} e^{anu} + \underline{b}_{n2} u e^{anu} + \underline{b}_{n3} e^{-anu} + \underline{b}_{n4} u e^{-anu}, \tag{5}$$

where $\underline{a}_{00}, \underline{a}_{01}, \underline{a}_{02}, \underline{a}_{03}$ $\underline{a}_{n1}$, $\underline{a}_{n2}$, $\underline{a}_{n3}$, $\underline{a}_{n4}$, $\underline{b}_{n1}$ $\underline{b}_{n2}$, $\underline{b}_{n3}$ and $\underline{b}_{n4}$ are vector constants, whose values are determined by the imposed boundary conditions at $u = 0$ and $u = 1$.

For a general set of boundary conditions, in order to define the various constants in the solution, it is necessary to Fourier analyse the boundary conditions and identify the various Fourier coefficients. If the boundary conditions can be expressed exactly in terms of a finite Fourier series, the solution given in Eq. (2) will also be finite. However, this is often not possible, in which case the solution will be the infinite series given Eq. (2).

The technique for finding an approximation to $\underline{X}(u, v)$ is based on the sum of the first few Fourier modes and a 'remainder term', i.e.,

$$\underline{X}(u,v) \simeq \underline{A}_0(u) + \sum_{n=1}^{N}[\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)] + \underline{R}(u,v), \tag{6}$$

where N is usually small (e.g. $N < 10$) and $\underline{R}(u,v)$ is a remainder function defined as,

$$\underline{R}(u,v) = \underline{r}_1(v)e^{wu} + \underline{r}_2(v)e^{wu} + \underline{r}_3(v)e^{-wu} + \underline{r}_4(v)e^{-wu}, \tag{7}$$

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

$$\underline{F}(u,v) = \underline{A}_0(u) + \sum_{n=1}^{N}[\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)]. \tag{8}$$

An important point to note here is that although the solution is approximate this new solution technique guarantees that the chosen boundary conditions are exactly satisfied since the remainder function $\underline{R}(u,v)$ is calculated by means of the difference between the original boundary conditions and the boundary conditions satisfied by the function $\underline{F}(u,v)$ [27].

Note, with the above solution form one could define the position and derivative boundary conditions at the edges of $u = 0$ and $u = 1$ to generate a typical surface patch. Such a surface patch can be controlled by the position vectors (defined by the position boundary conditions) and derivative vectors (defined by the derivative boundary conditions).

However, for practical reasons, in order for a given surface patch to pass through all the four boundary conditions one could make a slight adjustment to the way the boundary conditions are defined as described below.

Choosing the parametric region to be $0 \leq u \leq 1$ and $0 \leq v \leq 2\pi$, the periodic boundary conditions can be expressed as in terms of just positional conditions as, $\underline{X}(0,v) = \underline{P}_1(v)$, $\underline{X}(1,v) = \underline{P}_2(v)$, $\underline{X}_a(a,v) = \underline{P}_a(v)$, and $\underline{X}_b(b,v) = \underline{P}_b(v)$.

The above does not affect the core of the PDE formulation and without loss of generality, PDE surface representation with these curves can be obtained [28]. For example, in order to create the vase with a closed base, we can use seven boundary curves. These seven boundary curves will generate two different patches; one patch for the vase body and the other patch for the base of the vase. Using four consecutive boundary curves accordingly (e.g. 1, 2, 3, 4 and 4, 5, 6, 7) means the adjacent PDE patch shares a common boundary curve (i.e. in this case curve 4). This ensures positional continuity across PDE patches. We show this by way of an example in Fig. 1. Figure 1a shows the boundary curves (1, 2, 3, 4, 5, 6 and 7) that are used to generate the vase shape in Fig. 1b.

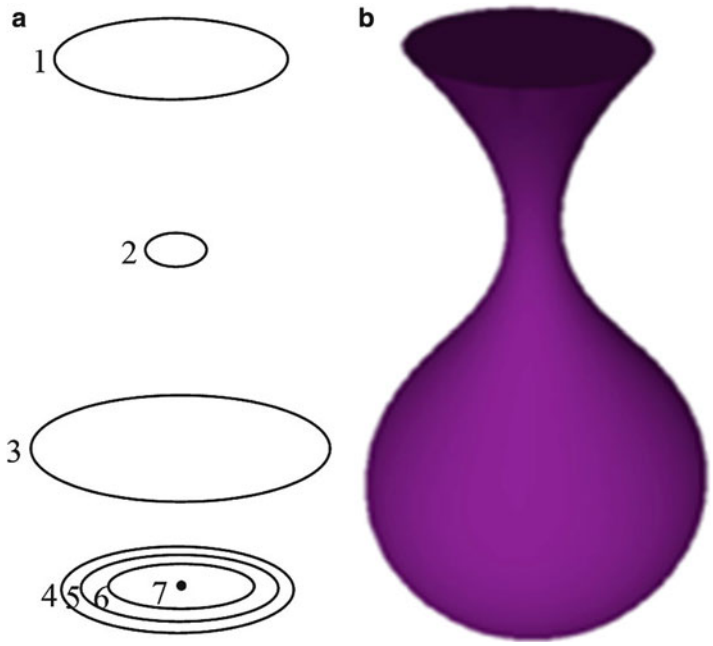

Fig. 1 Example of a PDE surface. (**a**) Boundary curves, (**b**) PDE surface of a vase shape

3 Method of Facial Modelling

The basic idea we have used for representing facial geometry is to identify and utilise a number of curves across the face which define facial geometry accurately. For this purpose we one can obtain such curve data from real face measurements such as the use of a 3D scanning device.

For the purpose of this work, we have utilized a database of dynamic 3D FACS developed at the Department of Computer Science at the University of Bath [29]. This database contains ten subjects, consisting of four males and six females. All subjects performed between 19 and 97 different AUs both individually and in combination with a total of 519 AU sequences. The peak expression frame of each sequence has been manually FACS coded by certified FACS experts.

The 3D data in the database are available in .OBJ file format for the 3D spatial data along with .BMP file format for UV color texture map data, for each frame of the dynamic data. The OBJ data consists of the two pod half-face meshes joined together. A database of this nature has been highly useful for us to develop and verify our work described in this paper. In particular, we have utilised this database to extract boundary curves in order to realistically describe the face as well as develop the PDE based AUs.

Within the 3D FACS database, there are ten subjects performing single action units and a combination of action units. From the sequences of 3D mesh data available, the neutral face for each person was chosen. The chosen neutral configuration

acts as a template of PDE face for each person. Then, the template is used for extracting boundary curves for a given action unit.

The obtained 3D mesh data from the database would require pre-processing prior to that 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.

To start with 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 of the nose is located the mesh is translated so the nose tip is located at origin. Next, the nose bridge and the center of the nose bottom are located manually. Based on the nose tip, nose bridge and nose bottom, the facial mesh can be rotated by aligning the mesh parallel to *yz* Cartesian coordinate plane. This task is performed automatically within Maya using a MEL script specifically written for this purpose.

3.1 Boundary Curve Extraction

The most important task when shape modelling using the PDE formulation is to make available a set of suitable boundary curves that represent the object in question. For the purpose of human face representation we can ensure curves are extracted so as important features of the face are accurately represented. These include facial feature such as nose, mouth and the eye areas. Previous PDE face models (e.g. [22, 28, 30]) 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 surface patches is generated by four consecutive boundary curves. Note that all nine PDE surface patches share common boundary curves to guarantee positional continuity between all the generated patches.

Referring to the previous work on PDE face representation we have re-formulated the way curves are extracted. This is necessary due to the fact that our prime concern here is not just to represent the face but also the underlying muscle structure of the face. Thus, in this work, the boundary curves are based on the facial anatomy representing the geometric aspects of facial muscle distribution and the movement of muscles of the face. In order to position points to form boundary curves, we determine a set of characteristic points that correspond to feature points corresponding to the muscles as defined by FACS.

For this purpose, vertices on the original mesh are located corresponding to the related muscle and muscle distribution. Once the points were selected, the MEL script was used to export them to an OBJ file. The boundary curves extracted for a typical face are shown in Fig. 2b. We found that a total of 31 boundary curves is sufficient to accurately represent a face, as shown in Fig. 2a.

It is important to mention that all 31 extracted boundary curves must be in closed form to ensure that the correct PDE face is produced. The generating boundary

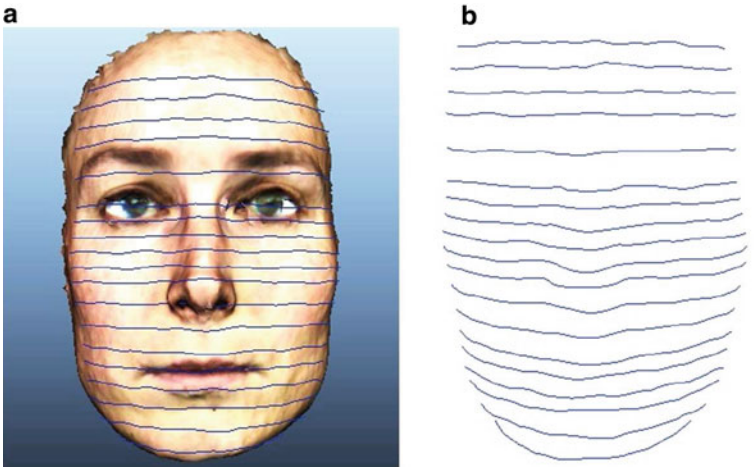

Fig. 2 A neutral face and position of extracted curves, (**a**) a face from the FACS database, (**b**) generating boundary curves

curves shown in Fig. 2b are not in closed form as the mesh image only contains the image of front face. Thus the MEL script is used to close the curves by reading the OBJ file and adding fictitious points by reflecting the coordinate to opposite side.

Following this, we then represent the points of the curves in terms of their corresponding Fourier series. If we approximate the boundary curves using a finite Fourier series such as,

$$f_i(v) = \underline{a}_{0u} + \sum_{n=1}^{5} [\underline{A}_{ni} \cos(nv) + \underline{B}_{ni} \sin(nv)], \tag{9}$$

where $i = 1, 2, 3, 4$.

This procedure enables us to store the facial information in the form of Fourier coefficients corresponding to the curves representing the face. i.e. taking Eq. (9), we can represent facial information in form,

$$M_f = \begin{bmatrix} a_{01}\ a_{11}\ \ldots a_{51}\ b_{11}\ \ldots b_{51} \\ a_{02}\ a_{12}\ \ldots a_{52}\ b_{12}\ \ldots b_{52} \\ a_{03}\ a_{13}\ \ldots a_{53}\ b_{13}\ \ldots b_{53} \\ a_{04}\ a_{14}\ \ldots a_{54}\ b_{14}\ \ldots b_{54} \end{bmatrix}. \tag{10}$$

With this formulation a full PDE face is generated by four sets of consecutive boundary curves resulting in ten continuous facial surface patches as shown in Fig. 3. Figure 3a shows the neutral PDE face with ten different patches. The corresponding PDE face with a texture applied on it is shown in Fig. 3b.

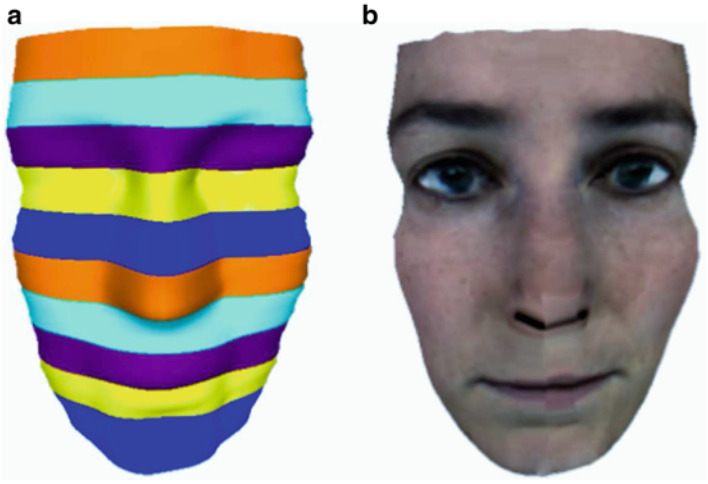

Fig. 3 PDE generated neutral fac, (**a**) original PDE face with patches shown, (**b**) Textured PDE neutral face

4 Modeling Action Units Using PDE Formulation

Previously in [23] it has been shown how simple mathematical transformations can be utilised to create generic facial expression, by identifying the boundary curves and points in these curves that are affected by each expression.

In this work we utilise the facial expression data from the 3D FACS database in order to produce more accurate facial expressions. For this purpose we use the FACS formulation. Here 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.

The process to create a given action unit from neutral PDE face representation with a given action unit is carried out by aligning the neutral mesh and the action unit mesh in the same position. Key features of the two meshes must be positioned so that they nearly overlap for facilitating the correct correspondence between two surfaces. Once the mesh for a given action unit is properly aligned, the Mel script reads the points of neutral configuration of boundary curves from external file and file the closest point for all set of points from action unit mesh image. Then, the script will automatically close the boundary curves and write the OBJ file based on the closest point found by the script. The process was repeated for all 31 boundary curves on the mesh with the given action unit.

If we refer M_f as the matrix of Fourier coefficients representing the natural pose of the face and M_g corresponding to the matrix of Fourier coefficients for a given action unit then for the given action we store the action unit data as,

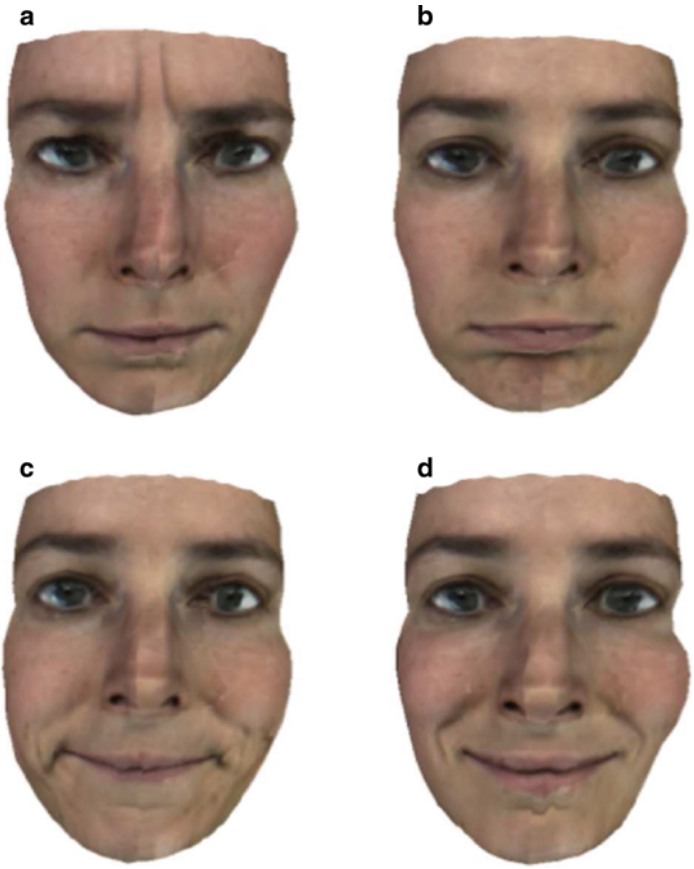

Fig. 4 PDE generated facial action units, (**a**) AU4, (**b**) AU17, (**c**) AU14 and (**d**) AU12

$$M_{AU} = M_f - M_g \tag{11}$$

Once we have computed all the necessary M_{AU}, (which we have done by utilising the data from the 3D FACS database) then a given FACS can be created using by adding M_{AU} to M_f, the neutral pose of the face.

Thus, in order to generate a PDE face with a given action unit, the coefficients M_{AU} are added to neutral Fourier coefficients, M_f, and Eq. (1) is solved to generated the PDE face with the given action unit. Figure 4 shows some example of generic PDE faces with action units. Figure 4a–d respectively represent AU4 (brow lowerer), AU17 (chin raiser), AU14 (dimpler) and AU12 (lip corner puller).

5 Results and Analysis

The PDE based FACS approach proposed here has been implemented using C# and Mel scripting. Once the relevant FACS are created for the PDE geometry we have then tested our FACS generation capacity for the other facial data in the database.

Apart from the efficiency through which we can generate facial feature via the PDE formulation it has an added advantage in that we can store facial expressions very efficiently. For instance, facial images found in the 3D FACS database range in mesh density between 15K and 16K vertices. The file size of the mesh in the database varies from 3 to 5 MB. In contrast to this, by only storing small amounts of parametric data in terms of Fourier coefficients, the PDE formulation enables us to store the facial data very efficiently. In fact, a typical facial expression can be stored using only 50 kB data, which include storage of the both the matrices M_{AU} and M_f. It is important to highlight that given M_{AU} and M_f the given facial expression corresponding to the given set of action units can be created to any given resolution using the pseudo-analytic solution given in Eq. (6).

A natural question one would ask is the accuracy of the facial data we are able to represent using the PDE formulation when compared to raw scanned data, for example. To demonstrate the accuracy of our data representation with respect to the raw data we overlaid the generic PDE mesh and the mesh from the database as illustrated in Fig. 5. Here, the green surface represents the PDE generated face which is placed on the corresponding facial data from the database.

Further, we created three cut-planes through the facial data as shown in Fig. 5. Note that, cut-plane 1 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

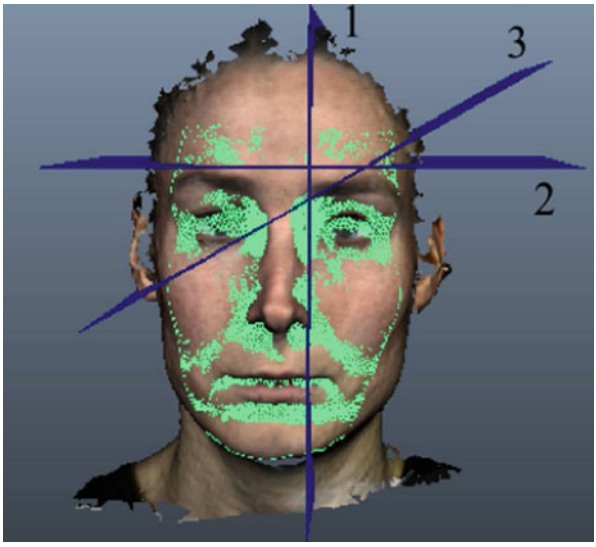

Fig. 5 Original face mesh and the corresponding PDE face overlaid along with three different cut-planes across the faces

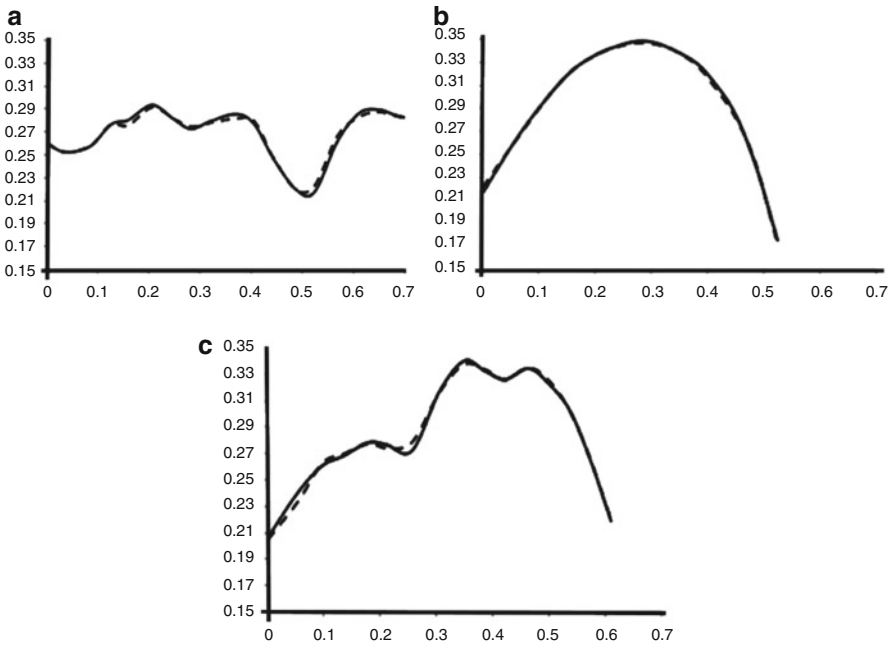

Fig. 6 Accuracy comparison between original and PDE generated facial mesh. Graphs showing three different cut-planes for a neutral face. (**a**) Vertical, (**b**) horizontal and (**c**) diagonal cut-planes

not effect the action unit movement. Cut-plane 2 and cut-plane 3 are somewhat randomly placed where the aim is to measure the accuracy of data the PDE method produces. To do this, we identify the corresponding points on both the faces (the PDE face and the corresponding face from the database) on each of the cut-planes and compare their differences. The results for neutral face are shown in Fig. 6.

The solid line and the dotted lines, respectively, in the graphs shown in Fig. 6 represent the original mesh taken from database and generic PDE generated face. From Fig. 6, it is clear that PDE face is very close to the original mesh. The mean error for vertical, horizontal and diagonal planes are within the 10^{-4} range.

Finally we look at the errors in the PDE generated face for action units in comparison with the corresponding meshes from the database. In particular, we look at the errors for AU4 and AU14. This is illustrated in Figs. 7 and 8. Again from the graphs shown in Figs. 7 and 8 we note that the PDE generated facial expressions are in close agreement with the real expressions found in the database.

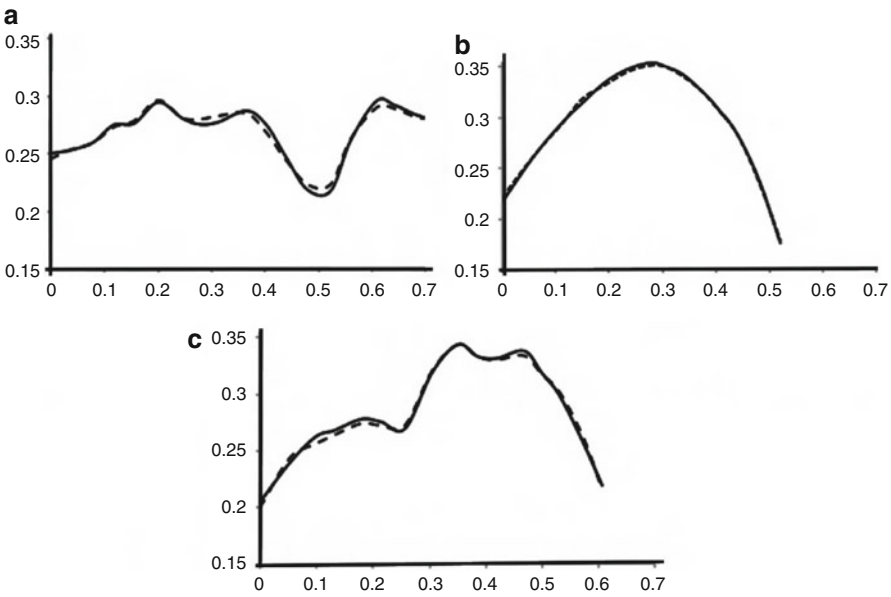

Fig. 7 Accuracy comparison between original and PDE generated facial mesh. Graphs showing three different cut-planes for a face with AU4. (**a**) Vertical, (**b**) horizontal and (**c**) diagonal cut-planes

6 Conclusions

In this paper we have described an approach to model action units on human faces based on Partial Differential Equations. In particular, we show how a boundary-value approach to solving a Biharmonic type PDE can be utilised to efficiently model facial action units and store the facial expression data in a mathematically compressed form. This work has also demonstrated that the PDE based approach is not only flexible but also accurately represent human facial expression data.

Work is currently under way to utilise this formulation so as it can be used to define a wide variety of realistic facial expressions. Such expression can then be used for realistic facial animations whereby subtle changes in expressions among individual faces can also be taken into account.

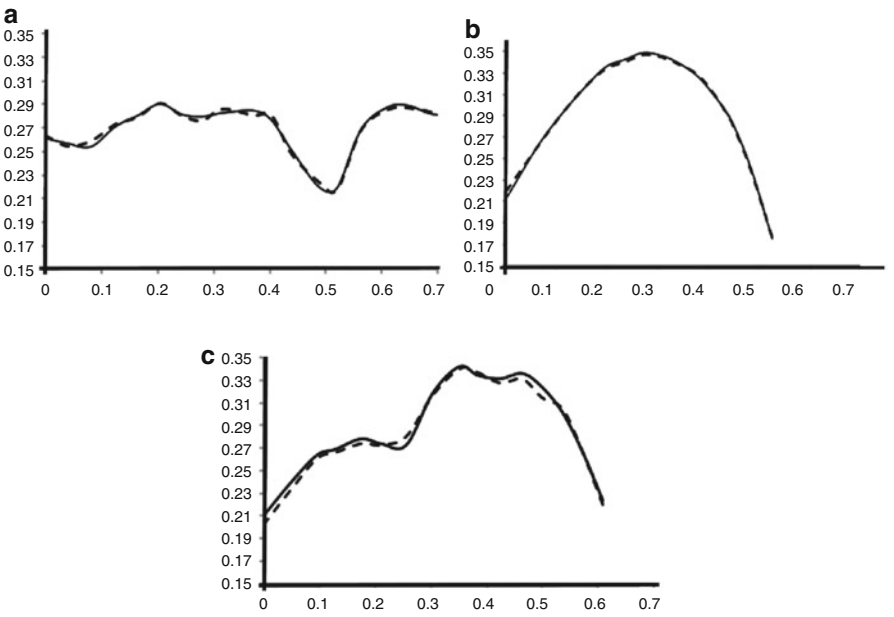

Fig. 8 Accuracy comparison between original and PDE generated facial mesh. Graphs showing three different cut-planes for a face with AU14. (**a**) Vertical, (**b**) horizontal and (**c**) diagonal cut-planes

References

1. I.A. Essa, A.P. Pentland, Coding, analysis, interpretation, and recognition of facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 757–763 (1997)
2. H. Ugail, *Partial Differential Equation for Geometric Design* (Springer, London, 2011)
3. Y. Sheng, P. Willis, G. Castro, H. Ugail, Facial geometry parameterisation based on partial differential equations. Math. Comput. Model. **54**(5–6), 1536–1548 (2011)
4. J. Noh, U. Neumann, A survey of facial modeling and animation techniques. USC Technical Report (1998), pp. 99–705
5. F.I. Parke, Computer generated animation of faces, in *ACM 72, Proceedings of the ACM Annual Conference*, vol. 1 (1972), pp. 451–457
6. D. Terzopoulus, K. Waters, Physically-based facial modeling. J. Vis. Comput. Animat. **1**(4), 73–80 (1990)
7. F.I. Parke, Parameterized models for facial animation. IEEE Comput. Graph. Appl. **2**(9), 61–68 (1982)
8. T. Beier, S. Neely, Feature-based image metamorphosis. ACM SIGGRAPH Comput. Graph. **26**, 35–42 (1992)
9. M. Oka, K. Tsutsui, A. Ohba, Y. Karauchi, T. Tago, Real-time manipulation of texture-mapped surfaces. ACM Comput. Graph. **21**(4), 181–188 (1987)
10. Y. He, Y. Zhao, D. Jiang, H. Sahli, Speech driven photo-realistic face animation with mouth and jaw dynamics, in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. IEEE, Kaohsiung (ACM, 2013), pp. 1–4

11. V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Tech., SIGGRAPH 99* (1999), pp. 187–194
12. L. Zhang, S. Member, D. Samaras, Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 351–363 (2006)
13. B. Moghaddam, J. Lee, H. Pfister, R. Machiraju, Model-based 3D face capture with shape-from-silhouettes, in *IEEE Work Analysis and Modelling of Faces and Gesture Recognition* (2003), pp. 20–27
14. S. Romdhani, T. Vetter, Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior, in *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2 (2005), pp. 20–27
15. H. Yu, O.G.B. Garrod, P.G. Schyns, Perception-driven facial expression synthesis. Comput. Graph. **36**(3), 152–162 (2012)
16. S. Villagrasa, A.S. Sánchez, Face! 3D facial animation system based on facs, in *IV Iberoamerican Symposium in Computer Graphics - SIACG 2009* (2009), pp. 202–207
17. A. Wojdel, L.J.M. Rothkrantz, Parametric generation of facial expressions based on FACS. Comput. Graph. Forum **24**(4), 743–757 (2005)
18. P. Havaldar, Performance driven facial animation, in *ACM SIGGRAPH 2006* (2006), pp. 1–20
19. M. Hoch, G. Fleischmann, B. Girod, Modeling and animation of facial expressions based on B-splines. Vis. Comput. **11**(2), 87–95 (1994)
20. D. Huang, H. Yan, Modeling and animation of human expressions using NURBS curves based on facial anatomy. Signal Process. Image Commun. **17**(6), 457–465 (2002)
21. D. Huang, H. Yan, NURBS curve controlled modelling for facial animation. Comput. Graph. **27**(3), 373–385 (2003)
22. E. Elyan, H. Ugail, Reconstruction of 3D human facial images using partial differential equations. J. Comput. **2**(8), 1–8 (2007)
23. G.G. Castro, H. Ugail, P. Willis, 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
24. G.G. Castro, H. Ugail, P. Willis, I. Palmer, A survey of partial differential equations in geometric design. Vis. Comput. **24**(3), 213–225 (2008)
25. E. Elyan, H. Ugail, Interactive surface design and manipulation using PDE-method through autodesk maya plug-in, in *International Conference on Cyberworlds 2009* (2009), pp. 119–125
26. H. Ugail, Method of trimming PDE surfaces. Comput. Graph. **30**(2), 225–232 (2006)
27. M.I.G. Bloor, M.J. Wilson, Spectral approximations to PDE surfaces. Comput. Aided Des. **28**(2), 145–152 (1996)
28. Y. Sheng, P. Willis, G. Castro, H. Ugail, PDE face: a novel 3D face model, in *Proc. of the 8th IASTED International Conference Visualization, Imaging, and Image Processing (VIIP 2008)* (2008), pp. 408–415
29. D. Cosker, E. Krumhuber, A. Hilton, A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling, in *IEEE International Conference on Computer Vision (ICCV) 2011*. IEEE, Barcelona (Elsevier, 2011), pp. 2296–2303
30. Y. Sheng, P. Willis, G. Castro, H. Ugail, PDE-based facial animation: making the complex simple, in *Advances in Visual Computing*. Lecture Notes in Computer Science, vol. 5359 (Springer, Heidelberg, Germany, 2008), pp. 723–732