Ugail H and Elyan E (2007): "Efficient 3D Data Representation for Biometric Applications", ***Scientific Support for the Decision Making in the Security Sector, NATO Science for Peace and Security Series - D: Information and Communication Security. Volume 12. O. Kounchev, R. Willems,,*** **V. Shalmanov and Tsv. Tsachev (eds.), pp. 215 - 229.**

Efficient 3D Data Representation for Biometric Applications

HASSAN UGAIL *&* EYAD ELYAN

School of Informatics
University of Bradford
Bradford, BD7 1DP, UK
(h.ugail, e.elyan)@bradford.ac.uk

ABSTRACT. An important issue in many of today's biometric applications is the development of efficient and accurate techniques for representing related 3D data. Such data is often available through the process of digitization of complex geometric objects which are of importance to biometric applications. For example, in the area of 3D face recognition a digital point cloud of data corresponding to a given face is usually provided by a 3D digital scanner. For efficient data storage and for identification/authentication in a timely fashion such data requires to be represented using a few parameters or variables which are meaningful. Here we show how mathematical techniques based on Partial Differential Equations (PDEs) can be utilized to represent complex 3D data where the data can be parameterized in an efficient way. For example, in the case of a 3D face we show how it can be represented using PDEs whereby a handful of key facial parameters can be identified for efficient storage and verification.

KEYWORDS. Partial differential equations, 3D facial modelling, data storage

Introduction

Representation of 3D data involves the construction of smooth mathematical functions over distribution of data samples which often have a non-uniform distribution over the domain of interest. Thus, the aim of data representation is to reconstruct an underlying mathematical function (e.g. a surface) which can then smoothly propagate the information associated with the scattered data onto all positions in the domain. This is an important issue since scattered data sets that often arise from various data sources play a significant role in many areas of biometric applications. In biometric applications, ideally one would like to represent the data exactly, however the real world usually has more complexity than we are able to represent in geometric modeling, and therefore the goal is to approximate. As a rule, the more complex a model is, the more it is approximated. A good surface representation scheme returns an approximation of the geometry model minimizing the error term.

Today there exist many commercial Computer Aided Design systems which employ the conventional polynomial surface modeling schemes that allow us to represent the geometry of complex objects. Of these methods, one of the widely used methods is the spline based schemes which is now dominated by Non Uniform Rational B-Splines (NURBS) [1, 2].

One example of surface representation using spline based methods is that developed by Piegl [3, 4]. In his work Piegl describes how the geometry of a brush handle can be re-created whereby the aim was to produce a smooth surface given a series of cross sectional curves representing the original brush handle. For this problem, the points per cross section (defined through a space curve) were between 10 and 48 and the number of control points after merging all the patches was nearly 4000. In addition to the control points, the associated weights of the spline also needed to be taken into account. In a similar fashion, Pottmann [5] describes the approximation of a ruled cylindrical surface using NURBS. Here a surface of bi-degree with 7×25 control points was utilized.

One of the main drawbacks while using spline based methods is the need for extra storage to define complex geometry corresponding to real world objects. In some applications of NURBS the combinations of weights can be significantly small which often results in zero denominators and hence becomes a very difficult condition to impose during curve and surface fitting. Furthermore, handling possible errors can complicate any NURBS-based geometry program.

Triangular meshes or subdivision schemes [6,7,8] for surface representation has recently been popular as an alternative to spline based techniques. For example, Hubeli and Gross [9] describe a geometric surface in which Doo-Sabin subdivision scheme is applied to represent the geometry of a two manifold surface. For this example, in order to reconstruct the geometry of the shape they utilized around fifty smoothing steps using the subdivision method. Similarly, Catmull [10] reconstructs a teapot using a subdivision algorithm. Although being much more flexible than spline based techniques, mesh based schemes also posses restrictions and disadvantages. For example, when applying an extreme deformation to a triangular mesh surface, certain triangles exhibit strong stretching which leads to numerically and visually undesirable triangles that have to be overcome using other surface processing techniques.

Surfaces based on PDEs have recently emerged as a powerful tool for geometric shape modeling [11,12,13,14, 15, 16]. Using this methodology, a surface is generated as the solution to an elliptic Partial Differential Equation (PDE) using a set of boundary conditions. Reconstruction methods based on PDEs are efficient in the sense that it can represent complex three-dimensional geometries in terms of a relatively small set of design variables.

In this paper we discuss how the geometry representation of biometric data can be performed mathematically in order to define the shape as close as possible to the real surface in question. For this purpose we utilize techniques based on PDEs whereby we take an elliptic PDE based on the standard Biharmonic equation similar to which widely known as the 'PDE method' in the Computer Aided Geometric Design literature due to the pioneering work of Bloor and

Wilson [17]. Thus, we show how we can represent an existing geometry of an object as accurately as possible with minimal shape data information. In particular, we show how the 3D scan data corresponding human faces can be reconstructed. Furthermore, we show how 3D facial data can be processed in order to significantly reduce the data size for efficient storage and retrieval. As part of this work we also show how one could process the raw 3D facial data, generated from a 3D scanning device, in order to extract geometrically meaningful parameters from the data.

The paper is organized as follows. First in Section 2 we present the mathematical description of PDE surfaces and provide generic examples of how the method can be used for 3D data representation. Then in Section 3 we discuss how 3D faces can be reconstructed using our methodology. For this purpose we provide examples of 3D raw data and reconstructed data showing how data can be efficiently stored. In section 4 we discuss a general framework for processing 3D facial data in order to extract geometrically meaningful parameters from raw data for the purpose of storage and detection. Finally we provide some concluding remarks.

PDE Surfaces

In geometric design, it is common practice to define curves and surfaces using a parametric representation. Thus, surfaces are defined in terms of two parameters u and v so that any point on the surface $\underline{X}$ is given by an expression of the form, $\underline{X} = \underline{X}(u,v)$. This can be viewed as a mapping from a domain Ω in the (u,v) parameter space to Euclidean 3-space.

With the above formulation, 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 $\partial\Omega$. To satisfy these requirements the surface $\underline{X}(u,v)$ is regarded as a solution of a PDE of the form,

$$\frac{\partial^4 \mathbf{X}}{\partial u^4} + 2\frac{\partial^4 \mathbf{X}}{\partial u^2 \partial v^2} + \frac{\partial^4 \mathbf{X}}{\partial v^4} = 0 . \qquad (1)$$

The PDE given in Equation (1) is the fourth order elliptic PDE based on the Laplace equation. In order to solve Equation (1) four boundary conditions are required. For this work we take these boundary conditions in the form of space curves. These curves are taken to be the positions containing the data from the original object whose geometry we wish to represent.

There are various techniques for solving Equation (1) which range from analytic solution schemes to sophisticated numerical techniques. For this work where we require geometry of facial data to be created and re-created in real time we utilize an analytic solutions scheme. We assume periodicity in the solution such that Ω is taken to be the finite domain defined as $\{\Omega : 0 \le u \le 1, 0 \le v \le 2\pi\}$ such that,

$$X(0,v) = P_0(v),\quad X(s,v) = P_s(v),\quad X(t,v) = P_t(v),\quad X(1,v) = P_1(v) \tag{2}$$

where $P_o(v)$ and $P_1(v)$ define the edges of the surface at $u=0$ and $u=1$ respectively. The conditions $P_s(v)$ and $P_t(v)$ are then defined to be some intermediate positions within the data that require to be approximated. With this formulation the analytic solution of Equation (1) can be written down as,

$$\underline{X}(u,v) = \underline{X}(u)\cos(nv) + \underline{X}(u)\sin(nv) + \underline{R}(u,v), \tag{3}$$

where n is an integer. The form of $\underline{X}(u)$ subject to the general boundary conditions is given as,

$$\underline{X}(u) = c_1 e^{nu} + c_2 u e^{nu} + c_3 e^{-nu} + c_4 u e^{-nu}, \tag{4}$$

where c_1, c_2, c_3 and c_4 are given by,

$$c_1 = [-P_0\left(2n^2e^{2n} + 2ne^{2n} + e^{2n} - 1\right) + P_1\left(ne^{3n} + e^{3n} + ne^n - e^n\right) - 2P_s ne^{2n} - P_t\left(e^{3n} - e^n\right)]/d, \tag{5}$$

$$c_2 = [P_0\left(2n^2e^{2n} + ne^{2n} - n\right) - P_1\left(ne^{3n} + 2n^2e^n - ne^n\right) - P_s\left(2ne^{2n} - e^{2n} + 1\right) + P_t\left(e^{3n} - 2ne^n - e^n\right)]/d, \tag{6}$$

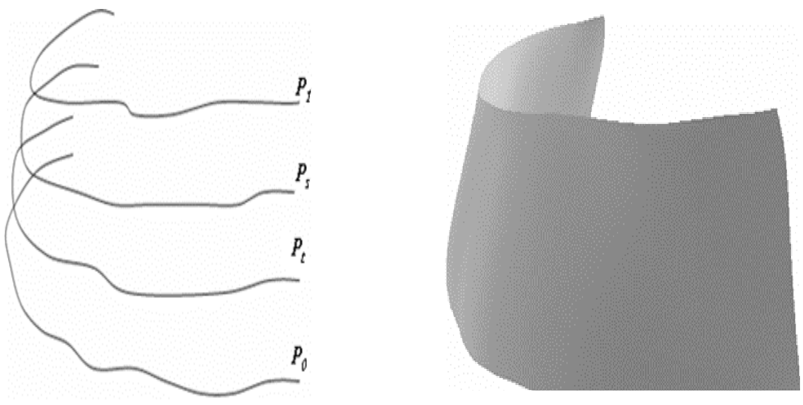
$$c_3 = [P_0\left(e^{4n} - 2n^2e^{2n} + 2ne^{2n} - e^{2n}\right) - P_1\left(ne^{3n} + e^{3n} + ne^n - e^n\right) + 2P_s ne^{2n} + P_t\left(e^{3n} - e^n\right)]/d, \tag{7}$$

$$c_4 = [P_0\left(ne^{4n} + 2n^2e^{2n} - ne^{2n}\right) - P_1\left(2n^2e^{3n} + ne^{3n} - ne^n\right) + P_s\left(e^{4n} - 2ne^{2n} - e^{2n}\right) + P_t\left(2ne^{3n} - e^{3n} + e^n\right)]/d, \tag{8}$$

where $d = e^{4n} - 4n^2e^{2n} - 2e^{2n} + 1$.

Given the above explicit solution scheme the unknowns c_1, c_2, c_3 and c_4 can be determined by the imposed conditions at u_i where $0 \le u_i \le 1$. The function

$\underline{R}(u, v)$ is computed using the spectral approximation methods based on the difference between a finite Fourier series representing a given boundary condition and the original data corresponding to that boundary condition. Details of this approximation method can be found in [13].



(a) **(b)**

Figure 1. A typical PDE surface path: (a) the boundary curves and (b) the resulting PDE surface

As an example consider the curves shown in Figure 1(a) which define the necessary conditions for solving the PDE given in Equation (1). One should note that the analytic solution scheme defined in Equation (3) is applicable to periodic conditions and therefore in order to solve Equation (1) we restrict the domain of the solution to $0 \le u \le 1$ and $0 \le v \le \pi$ which adequately describe the boundary conditions shown in Figure 1(a). Figure 1(b) shows the shape of the surface patch generated by solving Equation (1) subject to the conditions shown in Figure 1(a). It should be noted that the resulting surface shown in Figure 1(b) contains all the curves shown in Figure 1(a) whereby the PDE enables a smooth interpolation of the boundary curves. It is also worth highlighting that the resulting surface patch is solely controlled by the four boundary curves.

1. Three-dimensional Facial Data Representation

The most critical issue in representing 3D facial information using PDE based surface representation is to allocate a set of boundary curves that adequately describe the geometry of a human face. Clearly a human face is far more complicated than the surface patch shown in Figure 1. Our approach was based on treating the human face as a set of surface patches (segments) each defined in terms of 4 curves in the 3-space. These curves then serve us as the boundary conditions for solving Equation (1).

Essentially, there could be a number of ways to divide a human face into a set of surface patches. For this purpose we carried out some initial experimentations in order to decide how best to segment the facial data which will provide us with suitable curves serving as boundary conditions for the Equation (1). These experiments were carried out manually on empirical basis. Our main objective was to allocate a minimum number of parameters that best describe the geometry of a face whereby we focussed on the main characteristic features of the facial data, such as eyes, nose, and the central region. Having undertaken experimentations with several segmentation approaches, we have eventually adopted an arrangement of boundary curves that adequately represent the face. Our approach was based on representing the face as a set of cross-sectional curves.

Figure 2. illustrates our approach in defining the boundary conditions for solving Equation (1) to generate a 3D human facial surface. The simplest surface patch is clearly the forehead area, which is represented by four curves in 3-space. For a human face, the area belonging to the eyes is more complicated, thus needed more than four boundary curves in order to ensure the geometry is properly preserved. Similar to the area belonging to the eyes the area belonging to the nose and mouth are equally complicated and therefore require appropriate number of curves to ensure proper geometry is preserved. Thus, our experimentations showed that around 28 cross-section curves across the face are sufficient to generate the surface of a face through the PDE given in Equation (2).

Once the appropriate number of curves is identified for every two adjacent surface patches (for example, between forehead and eyes) the series of curves are taken together which form the necessary boundary conditions for generating a smooth surface corresponding to the face. In order to ensure that a smooth surface corresponding to the identified curves is generated we create a single surface patch using multiple calls to the solution of the PDE with the appropriate boundary curves at a given time. For this purpose we define the domain of the parametric region to be such that $0 \le u \le 9$ and $0 \le v \le \pi$. Thus, for the particular case we set u to range from 0 up to 9 since we define the complete face using 9 smoothly connected surface patches.

Figure 3 shows the resulting surface patch generated using the method described above, and using the set of profiles shown in Figure 2 as the boundary conditions for the chosen PDE.

It is important to note that the above arrangement of boundary profiles (as shown in Figure 2) is a generic representation of a 3D facial data so that it could be reconstructed using the solution of the chosen PDE. In addition, altering the boundary curves of this generic representation would produce varying geometries of faces other than the one shown in Figure 3.

Figure 2. Curves defining the boundary conditions of the PDE for generating a smooth surface.

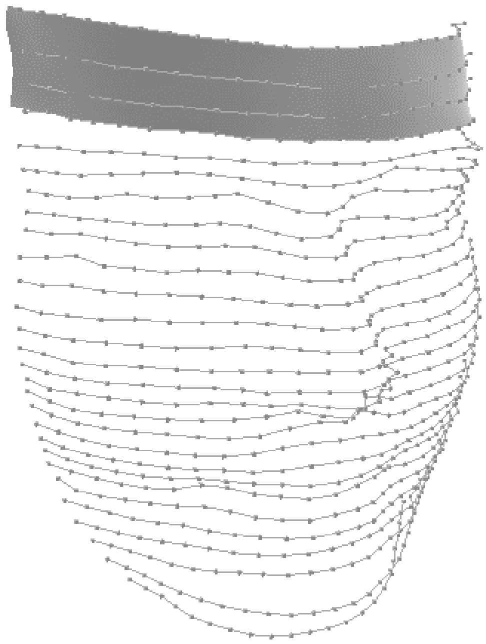


Figure 3 Smooth surface generated using the curves defined in Figure 2 as the solution of the PDE.

2. Techniques for Processing Raw Data Corresponding to 3D Faces

In this section we discuss some of the specific techniques we have developed in order to process raw data which usually comes for a 3D scanning device. The aim here is to process the raw facial data so that important features of the face can be automatically identified and then the face can be characterized in terms of the curves defined in the previous section.

Consider the facial data shown in Figure 4. Such data are usually the output from a 3D scan device after a facial scan and depending the type of the scanning device the data can be either in the form of a point cloud or a triangular mesh. For the purposes of reading the data in a convenient format we store the data (either point clouds or meshes) in a standard geometry description format know as OBJ format which contains information on the location of the data points in the physical space and the connectivity information defined between these points. It is important to highlight that captured facial data can be of any orientation or pose and therefore when developing our processing techniques we have made no assumption on the face orientation or the pose of the face. Moreover, as far as characterization of a given face is concerned the data from a scanner usually contains unwanted geometry (such as neck, shoulders and hair) that need to be identified and discarded at a pre-processing stage.

The aim here is to develop a robust methodology to process the raw data from a scanning device such that we can represent the facial information by a set of boundary curves (profiles) which can be used to solve PDE in order to reconstruct the geometry of the face automatically in real time. Once the appropriate facial profiles are extracted they can then be used to alter the generic template of the boundary curves (shown in Figure 2) in order to generate a particular face. The algorithm we have developed for processing facial data is as follows,

- Allocate the nose tip. Nose tip is chosen as our first step in the algorithm since it is considered as one the main facial features and is generally easiest to allocate.
- Define the symmetry plane of the face.
- Extract the symmetry profile based on the previously defined symmetry plane. This profile will essentially pass through the nose tip.
- Define a set of feature points on the symmetry profile.
- Extract the cross-sectional profile that passes through the corners of eyes. This profile will nearly be perpendicular to the symmetry profile.
- Extract a set of profiles from the facial data. These profiles will be arranged to match the arrangement of our generic template discussed above.
- Reconstruct the model through the solution of the PDE.

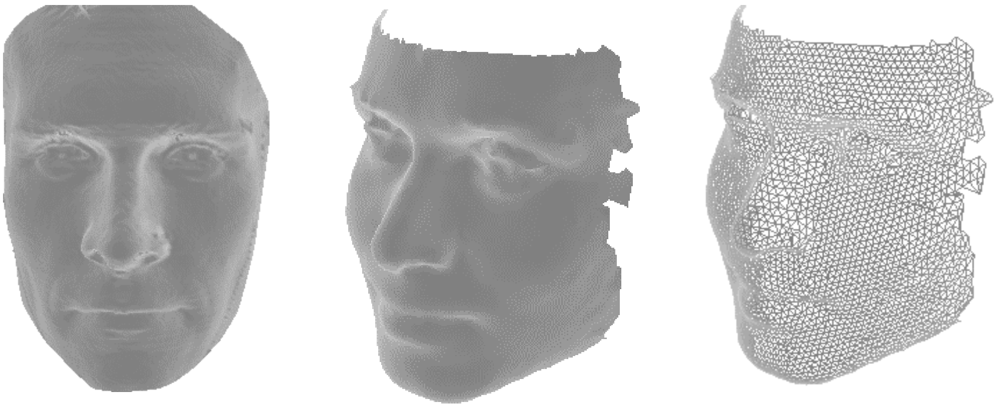


Figure 4 3D facial data obtained through a 3D scanning device, shaded models left, and wire-frame model right.

In the following section we will briefly discuss these steps, and show the resulting PDE surfaces based on the extracted boundary curves based on this procedure.

2.1. Nose Tip and Symmetry Profile Allocation

The tip of the nose is considered as one of the main feature points of a human face. In addition, it is generally the easiest to identify within the facial data. Theoretically the nose tip is the highest point at the human face. Therefore, applying some depth testing would lead to its identification. For this purpose, we construct a plane that provides depth information regarding the facial data. The depth plane is constructed using three points that are based on their relative positions from the centroid of the whole mesh.

The second step involves the projection data onto the depth plane in order to compute the Euclidean distance between them. Through this mechanism the tip of the nose can be extracted. Since facial mesh may contain irregular boundaries or extraneous parts such as hair, or part of the neck, we only consider part of the facial mesh that lies in the central region of the face. This restriction was made, by working out the Euclidean distance between each facial point, and the centroid of the mesh, and then discarding those extremes, which allows us to identify the central region which are often referred as the facial mask in the relevant literature [18]. This procedure enables us to neutralize the facial data with the tip of the nose residing at the origin of a right hand coordinate system.

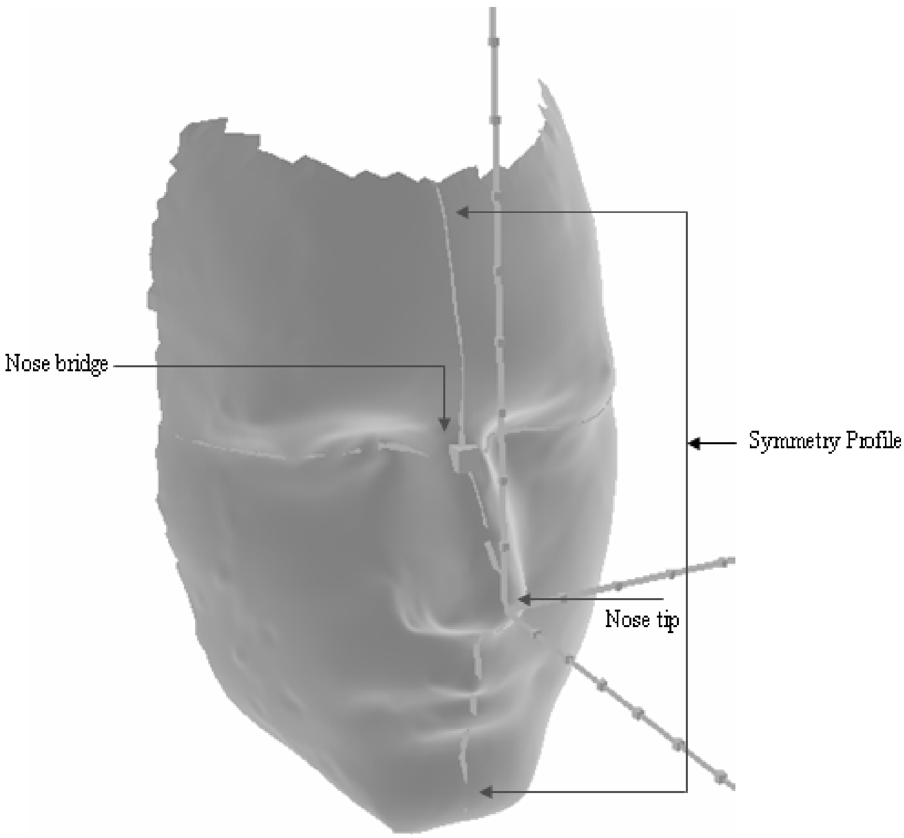


Figure 5 Allocation of nose tip , facial mask and symmetry profile of a 3D mesh

Once the tip of the nose is identified it becomes relatively straightforward to extract the symmetry profile since the tip of the nose is a point on the symmetry profile. Having identified the tip of the nose we then identify two points across the face which are extremes where the criteria being that the two points when projected on the depth plane makes an angle π with the projected nose tip on the same plane. This enables us to construct another plane which is perpendicular to the depth plane which intersects the face. Having constructed such a plane we can extract the set of facial points that intersect it. Defining the symmetry plane is then undertaken by rotating the perpendicular plane by 2π around the z-axis. Each rotation of the plane that intersects the face is done by an angle θ which is dependent on the density of the data. As shown in Figure 5 the profile that has the minimum geodesic distance would essentially correspond to the symmetry plane.

2.2. Extract Profiles from Facial Data and Construction of Facial Geometry

Once the symmetry profile is extracted using the above procedure, we then extract a curve that passes through the bridge of the nose. With these information, and since the face now is aligned within the Cartesian coordinate system, we construct a series of planes along the width of the face, and identify the set of facial points that intersect each of them. Figure 6 shows the profiles extracted through this technique. Figure 6 also shows the symmetry profile and the cross-sectional curve that passes through the nose bridge.

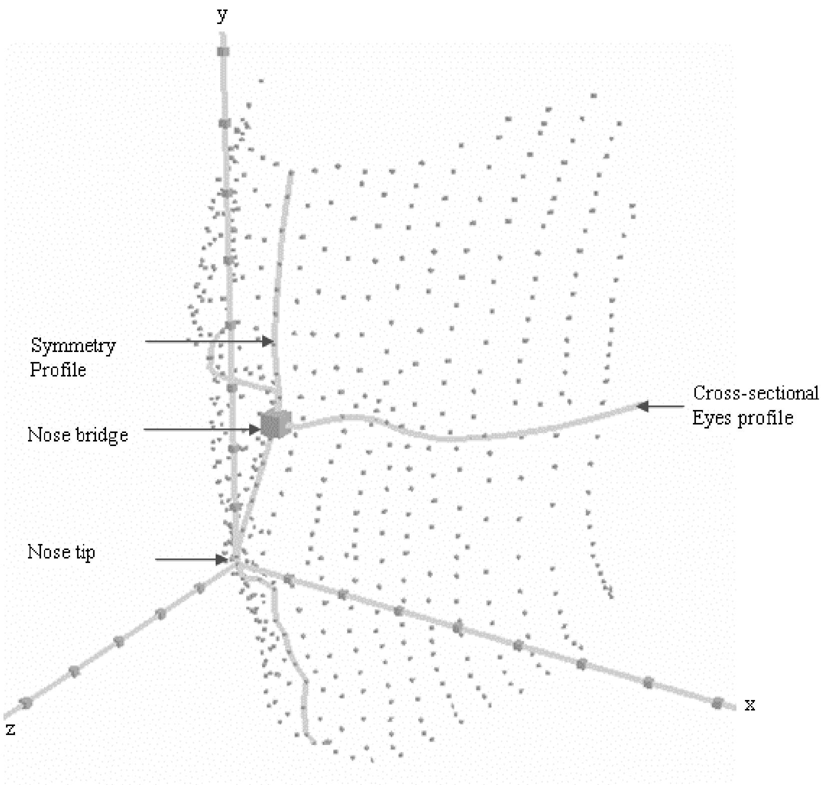


Figure 6 Automatically extracted set of facial profiles, with symmetry profile, and cross-sectional one that passes through nose bridge highlighted.

Once the curves are extracted, fore each curve we then fit a spline of the form $P_i = \sum C_i B_i$ where B_i is a cubic polynomial and C_i are the corresponding control points. This process of curve fitting to the extracted discrete curve data enables us to have a smooth curve passing through the discrete data as well as have an equal number of curve points for each profile curve.

Upon completion of the above process, the arrangement of the generic template discussed in section 3, is adjusted according to the new values of the extracted profiles. As one can see that reconstructing facial models based on allocating a set of boundary curves as discussed above through the solution of the PDE is an efficient technique for real time generation of facial data. Figure 7 shows sample results where both the original scan data and reconstructed facial data are shown.

Conclusions

The aim of this paper is to describe a methodology for efficient description to 3D data for biometric applications. In particular, in this paper we have proposed a methodology for constructing 3D geometry of human faces using solutions to an appropriately chosen PDE. Thus, we have utilised a boundary value approach whereby the solution of the standard Biharmonic equation is solved for appropriately defined curves (which are taken as boundary conditions) corresponding to the original facial data. Hence, in this work we have shown that the solution of the PDE we proposed produces a smooth single patch of the facial surface through a series of profile curves. This process also enables us to store the data corresponding to the face in terms of the profile curves thus saving substantial storage space. We have also discussed how raw facial data arising from 3D scanning devices can be processed in order to assign the profile curves so that facial can be represented through the PDE geometry.

Facial representation using the proposed PDE techniques has several advantages. This includes efficient rendering and visualization of 3D facial data, intuitive manipulation of 3D facial data using relatively few parameters.

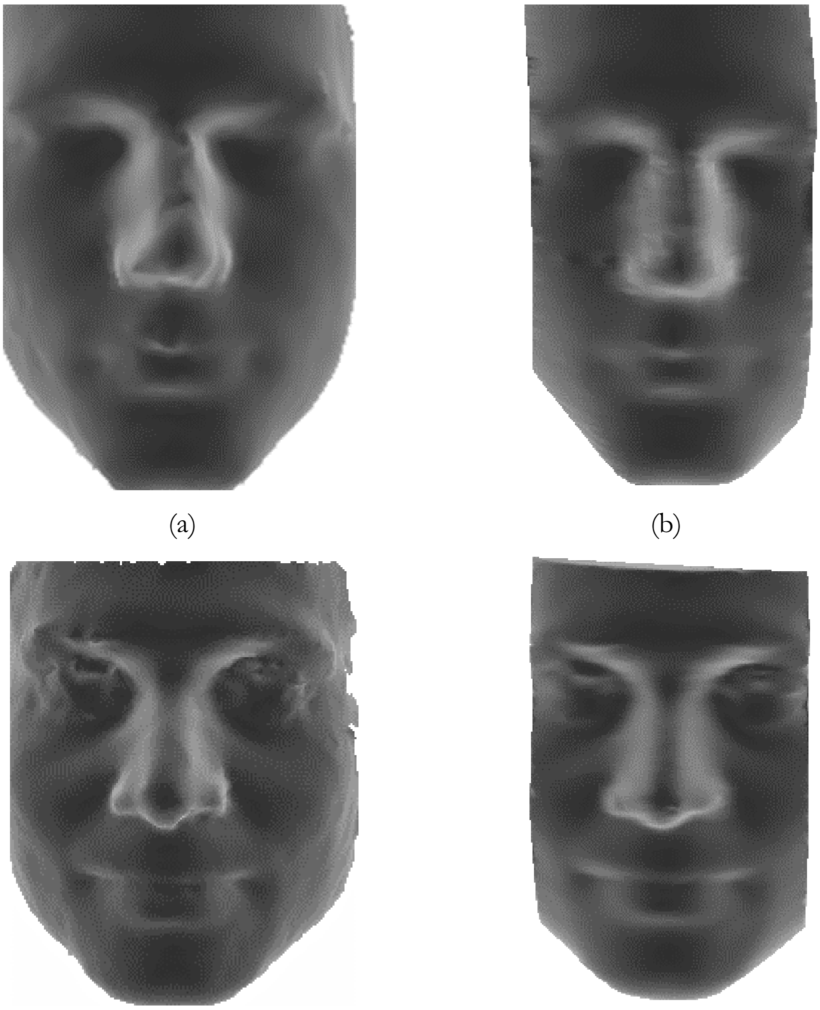
References

[1] I. D. Faux and M. J. Pratt, *Computational Geometry For Design And Manufacture*. Halsted Press, Wiley, 1979.

[2] G. Farin, From conics to NURBS, A Tutorial and Survey, *IEEE Computer Graphics and Applications* **12**(5) (1992), 78-86.

[3] L. Piegl, Recursive Algorithms for the Representation of Parametric Curves and Surfaces, *Computer Aided Design* **17**(5) (1985), 225-229.

[4] L. Piegl, *The NURBS Book*. Springer-Verlag, New York, 1997.



(a) (b)

(c) (d)

Figure 7 Construction facial geometry through PDE solution. (a) and (c) Original facial data, and (b) and (d) Reconstructed faces through PDE geometry.

[6] E. Catmull and J. Clark, Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes, *Computer Aided Design*, **10**(6) (1978), 350-355.

[7] C. T. Loop. *Smooth Subdivision Surfaces based on Triangles*. Master's Thesis, Department of Mathematics, University of Utah, 1987.

[8] N. Dyn, D. Levine and J. A. Gregory, A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, *ACM Transactions on Graphics* **9**(2), (1990), 160-169.

[9] A. Hubeli, and M. Gross, *A Survey of Surface Representations for Geometric Modeling.* Computer Science Department, ETH Zurich, Switzerland, CS Technical report No. 335, (2000).

[10] E. Catmull, *Subdivision Algorithm for the Display of Curved Surfaces.* PhD thesis, University Of Utah, 1974.

[11] M.I.G. Bloor and M.J. Wilson, Generating Blend Surfaces using Partial Differential Equations, *Computer Aided Design* **21**(3), (1989), 165-171.

[12] H. Du and H. Qin, A Shape Design System using Volumetric Implicit PDEs, *Computer Aided Design* **36**(11), (2004), 1101-1116.

[13] H. Ugail, M.I.G. Bloor, and M.J. Wilson, Techniques for Interactive Design Using the PDE Method, *ACM Transactions on Graphics* **18**(2), (1999), 195-212.

[14] L. You, P. Comninos, J.J. Zhang, PDE Blending Surfaces with C2 Continuity, *Computers and Graphics* **28**(6), (2004), 895-906.

[15] J. Monterde and H. Ugail, A General 4th-Order PDE Method to Generate Bézier Surfaces from the Boundary, *Computer Aided Geometric Design* **23**(2), (2006), 208-225.

[16] O. Kounchev, M. J. Wilson, Applications of PDE Methods to Visualization of Heart Data, In: *Mathematics of Surfaces 2003*, M.J. Wilson and R. R. Martin (eds.), Springer Lecture Notes in Computer Science, (2003) 377-391.

[17] M.I.G. Bloor and M.J. Wilson, The Efficient Parameterization of Generic Aircraft Geometry, *Journal of Aircraft* **32**(6), (1995), 1269-1275.

[18] K.W. Bowyer, K. Chang, P. Flynn, A Survey of Approaches to Three-Dimensional Face Recognition, *Pattern Recognition* 2004, ICPR 2004, 358- 361.