

3-D Facial Expression Representation using B-spline Statistical Shape Model

Wei Quan, Bogdan J. Matuszewski, Lik-Kwan Shark, Djamel Ait-Boudaoud
Applied Digital Signal and Image Processing Research Centre
University of Central Lancashire
Preston PR1 2HE, UK
{wquan,bmatuszewski,lshark,dait}@uclan.ac.uk

Abstract

Effective representation and recognition of human faces are essential in a number of applications including human-computer interaction (HCI), biometrics or video conferencing. This paper presents initial results obtained for a novel method of 3-D facial expressions representation based on the shape space vector of the statistical shape model. The statistical shape model is constructed based on the control points of the B-spline surfaces of the training data set. The model fitting for the data is achieved by a modified iterative closest point (ICP) method with the surface deformations restricted to the estimated shape space. The proposed method is fully automated and tested on the synthetic 3-D facial data with various facial expressions. Experimental results show that the proposed 3-D facial expression representation can be potentially used for practical applications.

1 Introduction

Facial expressions provide important information in communication between people, and can be used to enable communication with computers in a more natural way. Recently, some research efforts have been made in facial expression recognition for human-computer interaction (HCI) systems, video conferencing and augmented reality [15][9]. Most of the current facial expression representation methods are for 2-D images [14][1] and are only able to yield a good recognition performance in constrained environments such as those with proper head pose and lighting. By using 3-D faces instead of 2-D images, the performance of facial expression recognition can be significantly improved in terms of robustness and accuracy due to the extra information in the additional dimension [16]. In this paper, a novel approach is proposed for effective 3-D facial expression representation based on the shape space vector of the statistical shape model.

The objective of the data representation for expression recognition is to extract the data characteristics which can be utilised to distinguish different facial expressions. The representation methods can be generally classified into two categories: feature based and model based. The feature based approaches rely on global appearance features (for example, feature points associated with the Facial Action Coding System (FACS) [14][12]) and local appearance features (for example, the Local Binary Patterns (LBP) [18]) to provide most discriminating information of the face. The model based approaches rely on various

models such as deformable templates and statistical models [7][2]. An example for the former is a deformable template combined with the non-rigid registration method based on the iterative closest point (ICP) algorithm to achieve facial expression representation [16], and an example for the latter is the popular active appearance model which uses the geometric information as well as the texture information of the faces [7].

The method proposed in this paper uses the statistical shape model (SSM) in order to represent the facial expressions. Whereas most previously proposed methods use data points directly to build SSM (also known as point distribution model (PDM)), in the proposed method SSM is constructed from the control points of the B-spline surface fitted to the data points. The proposed method is similar in principle to the active shape model (ASM). Whereas ASM method models local texture distribution to search for the best data points to find the match for each model point, the proposed method uses modified ICP method for this purpose. Unlike the active appearance model (AAM) the SSM only relies on the shape of the face. Although AAM utilises more information, the SSM is independent of texture information of the face, and as such can be used when this information is missing (e.g. when a 3-D shape is measured using one of the sensors based on the time-of-flight principle) or the texture information is difficult to interpret due to varying lighting condition and/or head pose.

The remainder of the paper is organised as follows. Sections 2 and 3 explain the model building stages with Section 2 devoted to the facial surface fitting method based on B-splines and Section 3 describing the methodology for building the statistical shape model based on the control points of B-spline surfaces. This is followed by the model fitting stage described in Section 4 and the evaluation of the proposed method in Section 5, with the concluding remarks given in Section 6. Finally, the future work is sketched briefly in Section 7.

2 Spline Facial Surfaces

2.1 B-spline Surfaces

Most of 3-D model based face representation methods use a polygon mesh for representing the skin surface [21][16]. There are limited number of approaches which use a B-spline surface instead of a polygon mesh. Hoch et al [10] combined the control points of the B-spline surface and a FACS system to model the facial variations. Although a polygon mesh is easy to handle and control, it requires a significant storage space and is sensitive to the noise embedded in the data. Parametric B-spline surfaces are attractive because they are capable to efficiently represent 3-D faces and can resist a certain level of noise. The B-spline surface is given by

$$f(u, v) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} Q_{i,j} N_{i,k}(u) M_{j,l}(v) \quad (1)$$

where $Q_{i,j}$ are the control polygon net, with subscripts i and j being the indices of the control points, and $n \times m$ is the number of point in the control polygon net. $N_{i,k}$ and $M_{j,l}$ are the basis functions in the bi-parametric directions u and v . The basis functions are

calculated using following recursive formulae

$$N_{i,1}(u) = \begin{cases} 1 & \text{If } x_i \leq u < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \quad (3)$$

and

$$M_{j,1}(v) = \begin{cases} 1 & \text{If } y_j \leq v < y_{j+1} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$M_{j,l}(v) = \frac{(v - y_j)M_{j,l-1}(v)}{y_{j+l-1} - y_j} + \frac{(y_{j+l} - v)M_{j+1,l-1}(v)}{y_{j+l} - y_{j+1}} \quad (5)$$

where the x_i and y_i are elements of knot vectors.

2.2 Facial Surface Fitting

The purpose of the surface fitting is to calculate the control points of the B-spline surfaces which represent 3-D data usually by approximating the data in the least squares sense. Equation 1 can be expressed compactly in matrix notation as

$$f(u, v) = \beta(u, v)\Psi \quad (6)$$

which is a inner product between a vector of the B-spline basis functions

$$\beta(u, v) = (N_{0,k}(u)M_{0,l}(v), N_{0,k}(u)M_{1,l}(v), \dots, N_{n-1,k}(u)M_{m-1,l}(v)) \quad (7)$$

and a vector of control points

$$\Psi = (Q_{0,0}, Q_{0,1}, \dots, Q_{n-1,m-1})^T \quad (8)$$

In practice, the surface must be sampled, so the parametric variables u and v are replaced by their samples u_s and v_r , and the corresponding control points calculated in the least squares sense are given as

$$\Psi = (B^T B)^{-1} B^T F \quad (9)$$

where B is the matrix containing all the sampled vectors of the B-spline basis functions

$$B = \begin{pmatrix} \beta(u_0, v_0) \\ \beta(u_0, v_1) \\ \vdots \\ \beta(u_{p-1}, v_{q-1}) \end{pmatrix} \quad (10)$$

where F is a vector of data samples

$$F = (f_{0,0}, f_{0,1}, \dots, f_{p-1,q-1})^T \quad (11)$$

where p and q are the number of the data samples in the u and v directions, respectively. Additional optimisation methods, such as Newton or Levenberg-Marquardt, can be incorporated to optimise the values of the sampled parameter u_s and v_r in order to improve

the fitting result [17]. Figure 1 demonstrates the results of the B-spline surface fitting. The size of data samples is 70×90 , and the fitted B-spline surface is bi-cubic and contains 28×36 control points. Figure 1 (c)~(e) show the spatial distribution of data samples based on the value of the corresponding closest distance. The average closest distance between the data points and the fitted B-spline surface is $0.033mm$, which confirms that the B-spline surfaces are capable of representing 3-D facial data efficiently and accurately.

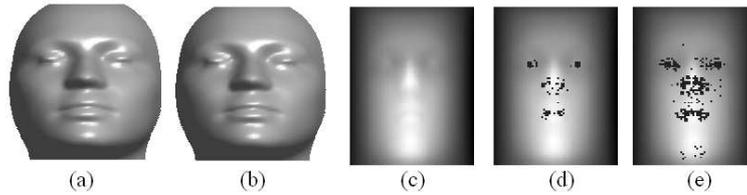


Figure 1: Results for B-spline surface fitting: (a) rendered 3-D data, (b) Fitted B-spline Surface, (c)-(e) display the closest distance maps which respectively are not bigger than 0.5, 0.2, and 0.1 mm (bigger the distance the brighter the point)

3 Statistical Shape Model

The statistical shape model also known as point distribution model was first proposed by Cootes et al [8], and it is one of the most widely used techniques in the model-based data representation. The model describes shape variations based on statistics of the position of the corresponding points in the training data set. In the proposed method, the statistical shape model is built on the corresponding control points of the B-spline surfaces fitted for the training data set by employing the principal component analysis (PCA).

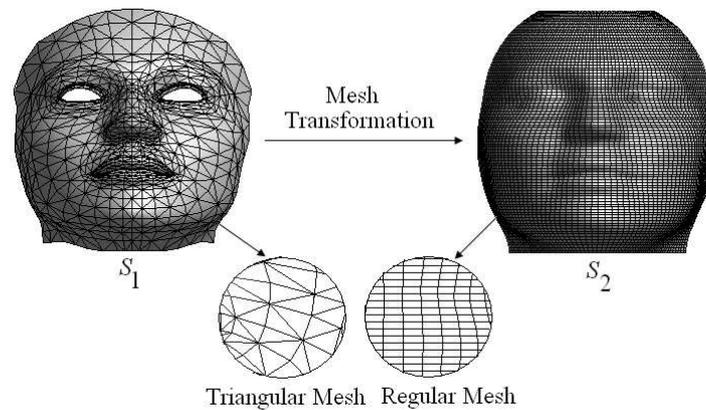


Figure 2: Mesh transformation from face a triangular mesh to a regular mesh

3.1 Establishing Correspondence

The first step of building a statistical shape model is to establish point-to-point correspondence between each face and a reference face in the training data set. The reference face is predefined as the first face in the training data set. The correspondence is essential, since incorrect correspondence can either introduce too much variations or lead to illegal instances of the model [5]. In the proposed method, a synthetic training data set with given correspondence information is generated using the *FaceGenModeller*[®] [19], therefore, the correspondence estimation for the training data set is not needed.

With the known point-to-point correspondence, the correspondence between control points of B-spline surfaces can be easily established by fitting the B-spline surfaces to the corresponding points in the training data set. The B-spline data approximation used in this method requires the data point to be on a regular mesh. Since the pattern of the output data from *FaceGenModeller*[®] is given as an irregular triangular mesh, the mesh transformation is necessary. The whole transformation contains three stages: (1) transformation from the Cartesian coordinate system into the cylindrical coordinate system, (2) regular mesh interpolation; and (3) transformation from the cylindrical coordinate system to the Cartesian coordinate system. These steps are described in more detail below, and the result of the mesh transformation using one of the training data is shown in Figure 2.

1. Transformation from Cartesian coordinates to cylindrical coordinates. The original training data set is given in the Cartesian coordinates. The cylindrical coordinates are used to transform the point pattern into regular mesh. The transformation can be written as

$$r = \sqrt{x^2 + y^2} \quad (12)$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (13)$$

$$z = z \quad (14)$$

where $r \in [0, \infty)$, $\theta \in [0, 2\pi)$ and $z \in (-\infty, \infty)$. The training data set is represented by (r, θ, z) instead of (x, y, z) in Cartesian coordinates as shown in Figure 3.

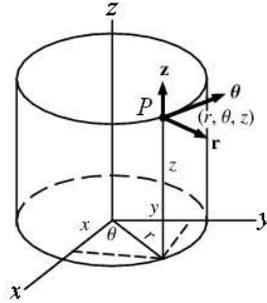


Figure 3: Transformation from Cartesian coordinate system to cylindrical coordinate system

2. Interpolation for vertices on a regular mesh defined on $\theta - z$ plane in the cylindrical coordinate system using vertices originally defined on a triangular mesh in $x - y - z$

coordinate system. Figure 4 shows the projection of the triangular mesh on $\theta - z$ plane for one of the training data. The interpolation can be achieved using one of the common surface interpolation methods such as radial basis function [6] or multi-level B-splines [13].

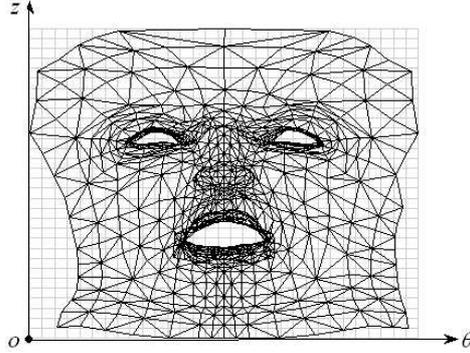


Figure 4: Face projection on the $\theta - z$ plane with original triangular mesh and interpolated regular mesh

3. Vertices transformation from the cylindrical coordinate system to the Cartesian coordinate system. The interpolated regular mesh is then transformed back to the Cartesian coordinate system. The transformation can be stated as

$$x = r \cos \theta \quad (15)$$

$$y = r \sin \theta \quad (16)$$

$$z = z \quad (17)$$

Having the regular mesh pattern, the training data set can be approximated by a B-spline surface. In this work, the mesh transformation is performed using the *InSpeckEM*[®] software [11]. Figure 5 shows a part of the training data set, which is in the regular mesh and contains 125 3-D synthetic faces with various expressions.

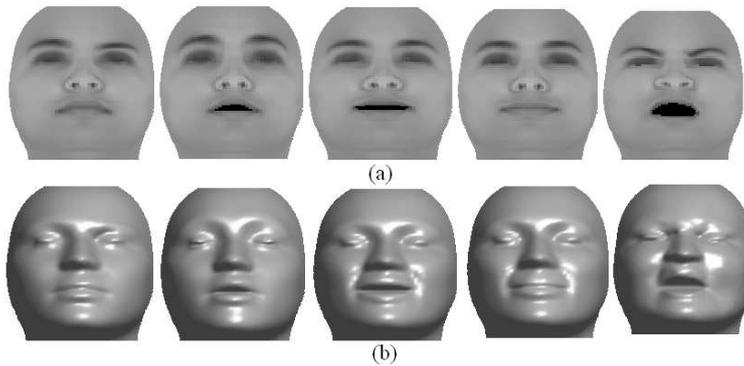


Figure 5: Part of training data set: (a) texture representation, (b) geometric representation

3.2 Principle component analysis

Using a standard principle component analysis (PCA), each face in the training data set can be approximately represented in a low dimensional shape vector space [5] instead of the original high dimensional data vector space. Given a training data set of N faces, $Q_i (i = 0, 1, \dots, N - 1)$, each containing M corresponding control points $Q_i \in R^{3M}$, where Q_i are the control points of the i -th face. Since the data points for each face in the training data set are in correspondence, the control points Q_i are in correspondence as well. The first step of the PCA is to construct the mean vector of the control points \bar{Q} (representing the mean face) for the training data set.

$$\bar{Q} = \frac{1}{N} \sum_{i=1}^N Q_i \quad (18)$$

Let C be defined as the covariance matrix calculated from the training data set.

$$C = \frac{1}{N} \sum_{i=1}^N (Q_i - \bar{Q})(Q_i - \bar{Q})^T \quad (19)$$

By building a matrix X of “centered” shape vectors with $Q_i - \bar{Q}$ as the i -th column of the matrix X , the covariance matrix C is defined as

$$C = X \cdot X^T \quad (20)$$

where matrix C has $3M$ rows and columns. Since the number of faces N in the training data set is smaller than the number of control points, the eigen decomposition of matrix $C' = X^T X$ is performed first. The first N largest eigenvalues $\lambda_i (i = 1, \dots, N)$ and eigenvectors $\mathbf{p}_i (i = 1, \dots, N)$ of the original covariance matrix C are then determined respectively by

$$\lambda_i = \lambda'_i \quad (21)$$

$$\mathbf{p}_i = \frac{X \cdot \mathbf{p}'_i}{\|X \cdot \mathbf{p}'_i\|} \quad (22)$$

where λ'_i and \mathbf{p}'_i are respectively eigenvalues and eigenvectors of the matrix C' . By using these eigenvalues and eigenvectors, the control points of any shape in the training data set can be approximately represented using a linear model of the form

$$\hat{Q} = Pb + \bar{Q} \quad (23)$$

where the so called “Shape Matrix” $P = [\mathbf{p}_1, \dots, \mathbf{p}_K]$ is a $3M \times K$ matrix of K eigenvectors associated with the first K largest eigenvalues “modes of variation” of matrix C . The shape space vector b controls contribution of each mode of variation in the approximation surface [5]. Most of variation can usually be modeled by a small number of modes K . Equation 23 can be used to generate new examples of faces by varying the shape space vector b with suitable limits, so the new faces will be similar to those in the training data set [8]. The suitable limits are typically defined as

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \quad (24)$$

Figure 6 shows the effects of varying the first three elements of the shape space vector.

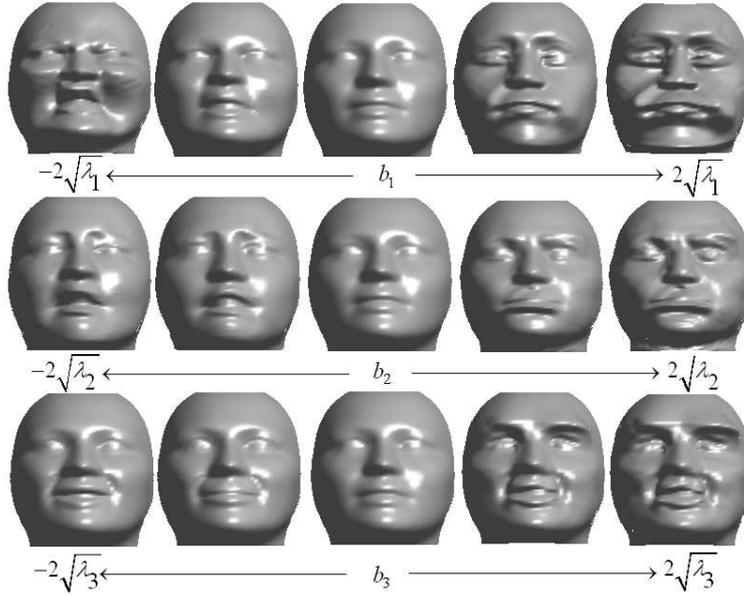


Figure 6: Effects of varying the first three shape space vector b_1 , b_2 and b_3

4 Model Fitting

Provided that the faces in the training data set are a fair sample from the population of faces to be represented by the model the statistical shape model can be used to represent faces not present in the training data set. In the proposed method the model fitting includes estimation of the shape and pose parameters. Whereas the pose parameters consists of translation vector, rotation matrix and a scaling factor the shape parameters are defined by the shape space vector. The algorithm starts with aligning data and the model mean face using similarity transformation. Subsequently model refinement continues by iteratively estimating the shape space vector and the pose parameters.

4.1 Similarity Registration

The iterative closest point (ICP) method can be used to achieve similarity registration between the new faces and a deformed model. The ICP [4] is a widely used point-based surface matching algorithm. The whole procedure iteratively refines the alignment by alternately choosing corresponding points and finding the best similarity transformation that minimizes a cost function based on the distance between the corresponding points. The cost function is defined by

$$E = \sum_{i=1}^{N_p} \|p'_i - (sRp_i + T)\|^2 \quad (25)$$

where p'_i and p_i ($i = 1, \dots, N_p$) are respectively the corresponding vertices from the model and the re-meshed data face represented as 3×1 vectors. R is a 3×3 rotation matrix, T

is a 3×1 translation vector and s is a scaling factor. Following the algorithms in [3][20], R , T and s are calculated as follow:

1. From the point sets, $\{p_i\}$ and $\{p'_i\}$ ($i = 1, \dots, N$), compute the mean vectors, \bar{p} and \bar{p}' , using

$$\bar{p} = \frac{1}{N} \sum_{i=1}^N p_i \quad (26)$$

$$\bar{p}' = \frac{1}{N} \sum_{i=1}^N p'_i \quad (27)$$

2. Calculate q_i and q'_i ($i = 1, \dots, N$) using

$$q_i = p_i - \bar{p} \quad (28)$$

$$q'_i = p'_i - \bar{p}' \quad (29)$$

3. Calculate the matrix H using

$$H = \sum_{i=1}^N q'_i q_i^T \quad (30)$$

4. Find the SVD of H

$$H = U \Sigma V^T \quad (31)$$

5. Calculate

$$X = UV^T \quad (32)$$

6. Compute the rotation matrix using

$$R = UDV^T \quad (33)$$

$$D = \begin{cases} I & \mathbf{If} \det(X) = +1 \\ \text{diag}(1, 1, -1) & \mathbf{If} \det(X) = -1 \end{cases} \quad (34)$$

7. Find the translation vector and scaling factor using

$$s = \frac{\text{tr}(\mathbf{q}\mathbf{q}^T R)}{\text{tr}(\mathbf{q}\mathbf{q}^T)} \quad (35)$$

$$T = \bar{p}' - sR\bar{p} \quad (36)$$

where $\mathbf{q} = [q_1, \dots, q_N]$ and $\mathbf{q}' = [q'_1, \dots, q'_N]$ are $3 \times N$ matrices.

In Equation 33, matrix D is a "safeguard" which makes sure the calculated matrix R is a rotation matrix, and not a reflection in 3-D space. The similarity registration involves two steps: (i) estimation of correspondence, based on the closest point, between points from the new face and the points from the current model and (ii) computation of the pose parameters. The outline of the similarity registration algorithm is given in Algorithm 1. Figure 7 shows the result of similarity registration. The mesh and rendered surface indicate the new face and the model, respectively.

Algorithm 1: Similarity Registration Algorithm

input : vertices \mathcal{P} from the new face and vertices \mathcal{P}' sampled from the current model surface

output: transformed, using estimated similarity transformation, vertices \mathcal{P}

initialization :

set threshold τ ($\tau > 0$) for terminating the iteration, $k = 0$, $d_0 = \text{inf}$, $e = \tau$;

while $e \geq \tau$ **do**

$k = k + 1$;

 Compute correspondence $p_i \leftrightarrow p'_{j(i)}$ with $j(i) = \arg \min_{j \in 1:N_p} \|p_i - p'_j\|$;

 Compute the pose parameters : R , T , and s using equations 26 - 36;

 Transform the points from the set \mathcal{P} using similarity transformation

$p_i = sRp_i + T$ and update the set \mathcal{P} accordingly;

 Measure the misalignment d_k between corresponding points in the new data point set \mathcal{P} and model point set \mathcal{P}' ;

$e = d_{k-1} - d_k$;

end

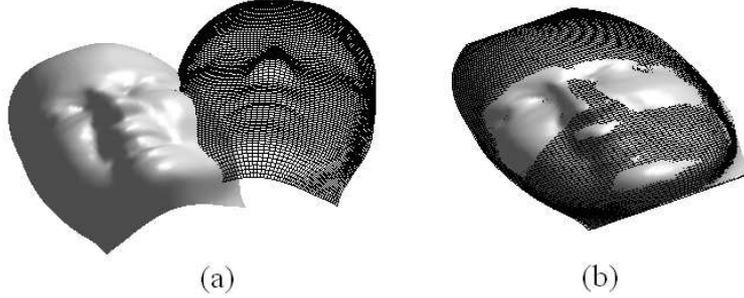


Figure 7: Results of the similarity registration: (a) before the registration, (b) after the registration

4.2 Model Refinement

With the data registered to the current model using similarity transformation, the objective of the model refinement is to deform the model so it is aligned to the transformed data points. To estimate optimal pose and shape parameters the whole process has to iterate. This can be seen as a hybrid of the ICP method and least squares projection onto the shape space. The least squares projection onto the shape space can provide the shape space vector \hat{b} which controls the deformation of the model and can be used as a feature vector for interpretation of facial expression of the new faces. The shape space vector \hat{b} for the new face is calculated as

$$\hat{b} = P^T(Q_d - \bar{Q}) \quad (37)$$

where $Q_d \in R^{3M}$ is a vector of M 3-D control points representing the new face. The mean vector of control points \bar{Q} and shape matrix P are obtained from Equations 18 and 22. The details of the algorithm are explained in Algorithm 2.

Algorithm 2: Model Refinement Algorithm

input : The point set \mathcal{P} of a new face, and the face model: P, \bar{Q}

output: Estimate of the shape space vector \hat{b}

initialization :

set threshold σ ($\sigma > 0$) for terminating the refinement, $\hat{b}_0 = 0, k = 0$;

while $\|\hat{b}_k - \hat{b}_{k-1}\| \geq \sigma$ **do**

 Calculate B-splines control points of the deformed model: $\hat{Q} = P\hat{b}_k + \bar{Q}$;

 Build the point set \mathcal{P}' from the deformed model surface, defined by \hat{Q} , using equation 6 with uniformly sampled parametric variables u and v ;

 Perform similarity registration between points sets \mathcal{P} and \mathcal{P}' using Algorithm 1 ;

 Calculate the control points Q_d for \mathcal{P} on the transformed new face using method described in section 2.2 ;

$k = k + 1$;

 Project the control points Q_d of the transformed new face onto the shape space $\hat{b}_k = P^T(Q_d - \bar{Q})$;

end

Examples of the results of the described algorithm are shown in Figure 8 and 9. It shows that the deformed model is well aligned with the new face in Figure 8. The corresponding distributions of the closest point distance before and after the model fitting are shown in Figure 8, and the maximum closest distance is reduced from above 10mm to around 3mm with average closest distance of 2.33mm.

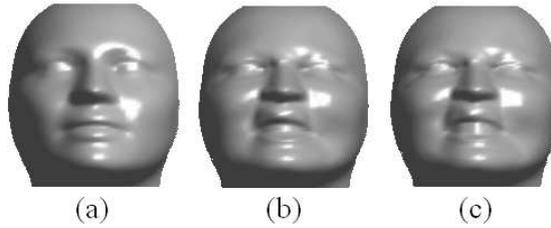


Figure 8: Results of the model fitting: (a) model mean face, (b) new face, (c) deformed model

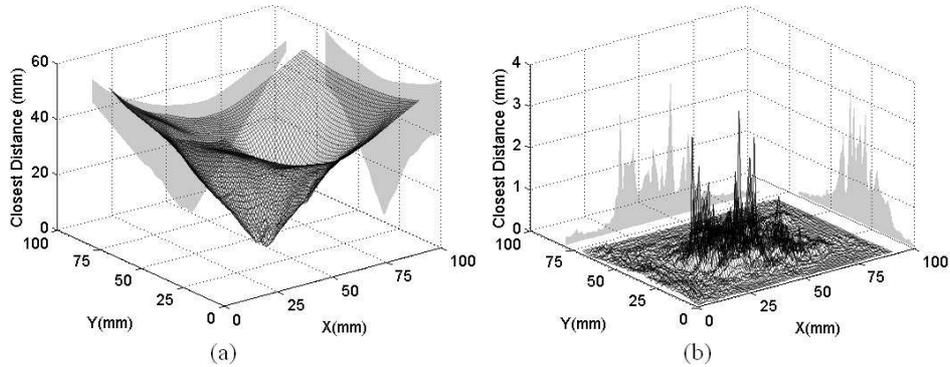


Figure 9: Closest distance as a function of point position: (a) before the model fitting, (b) after the model fitting

5 Experiments and Results

In this section, a selection of results based on different experiment configurations are presented to demonstrate the properties of the described method. In the first two experiments the training data set of 100 faces is used to build statistical shape model. In this data set 60% of faces are of the same individual with 7 different expressions, namely: anger, disgust, fear, surprise, smile closed, smile open, and sadness. The exact makeup of the expressions, including number of faces with the specific expression and level of the expression, is chosen randomly. The remaining 40% of the training data set consists of faces of 4 people of different ethnic origin with randomly selected expressions. For the third and final experiment the training data set has been extended by further 26 faces of two different individuals with randomly selected expressions from the "smile" and "anger" categories. Figure 5 shows a part of this training data set with five different facial expressions.

For the tests three different face categories are used. In the first experiment the same face is used as in the first 60% of the initial training data set, but with the expressions not present in the training data set. In the second experiment four different faces are used showing different level of anger expression. These faces were not included in the training data set. In the third experiment the extend training data set is used, with 126 faces in total, to produce new statistical shape model. In this case the test was carried out using the same faces as used in the additional 26 faces training subset but with face expressions not present in the training data set.

All the faces were generated using *FaceGenModeller*[®], and Figure 10 shows the 3-D mean face and superposition of the corresponding five principal 3-D eigenfaces with the mean face, obtained from the training data set. These five eigenfaces cover around 99% variation of the training data set.

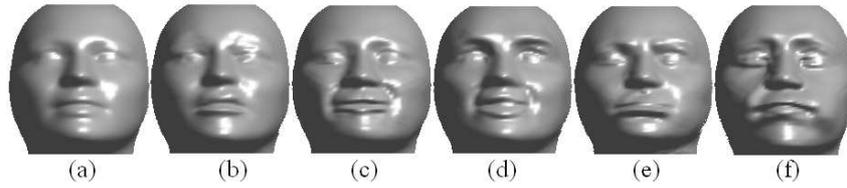


Figure 10: Face model: (a) mean face , (b)-(f) superposition of the mean face with first five eigenfaces

For the results obtained from the test performed on the same face type with different facial expressions figure 11 shows the 16 test faces. These faces are seen to have 4 different facial expressions, namely: anger, disgust, fear and smile, and each facial expression is seen to consist of four different degrees of expressiveness. Figure 12, shows the first three elements of the 5-dimensional shape space vectors $b = (b_1, \dots, b_5)^T$ calculated, using algorithm 2, for each face from figure 11. From Figure 12, it is seen that the shape space vectors corresponding to different facial expressions are well separated. Furthermore, the different degrees of expressiveness are indicated by the positions of shape space vectors along a line which orientation is defined by the expression type.

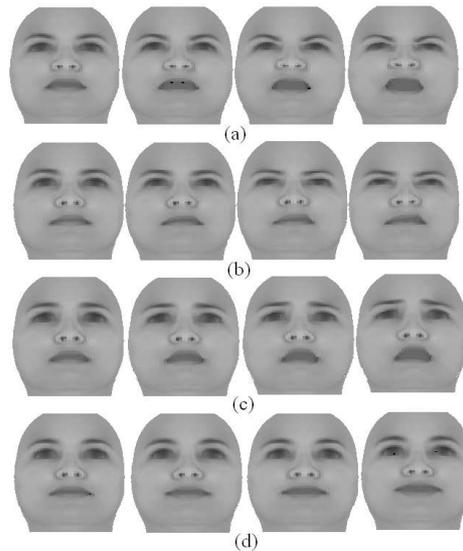


Figure 11: Faces representing different expressions for the same individual: (a) anger, (b) disgust, (c) fear, (d) smile

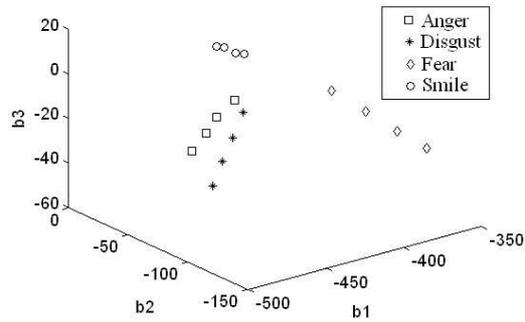


Figure 12: Representation of the first three elements of the shape space vectors for the faces from the Figure 11

For the results obtained from the test performed on different ethnic faces with the same expression of anger, Figure 13 shows the 16 test faces. These faces are seen to belong to 4 different ethnic groups, namely: African, European, Southeast Asian, and Indian, and the four faces in each group are seen to express a different degree of anger. Figure 14 shows the first three elements of the 5-dimensional shape space vectors b calculated for all test faces shown in figure 13. From Figure 14, it is seen that the shape space vectors are again well separated with the shape vectors corresponding to the same face located approximately on the same line. All the lines seem to have, in this case, the same orientation.

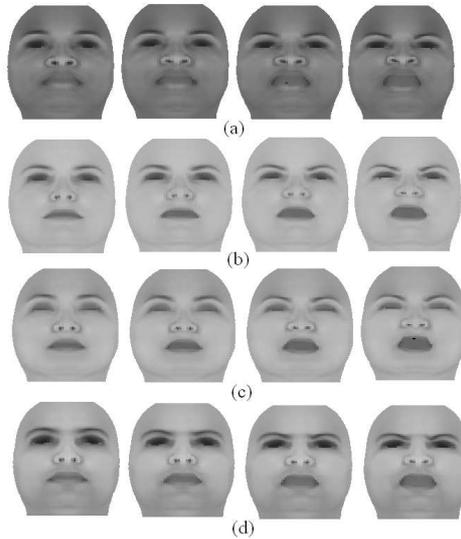


Figure 13: Faces representing the anger expression for individuals from different ethnic origin: (a) African, (b) European, (c) South East Asian, (d) Indian

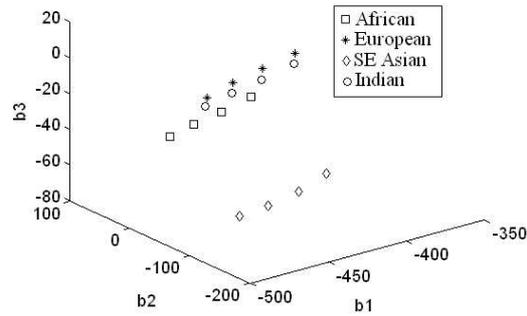


Figure 14: Representation of the first three elements of the shape space vectors for the faces from the Figure 13

For the results obtained from the test performed on different ethnic faces with anger and smile expressions figure 15 shows the 16 test faces. These faces are seen to belong to two different ethnic groups with particular expressions, namely: African-Anger, African-Smile, European-Anger and European-Smile, and each facial expression is seen to consist of four different degrees of expressiveness. Similarly, Figure 16 shows the first three elements of the 5-dimensional shape space vectors b . From Figure 16, it is seen that the shape space vectors associated with the same type of facial expression from African and European group are positioned in similar orientation in the shape space, and different type of facial expression from African and European groups are well separated.

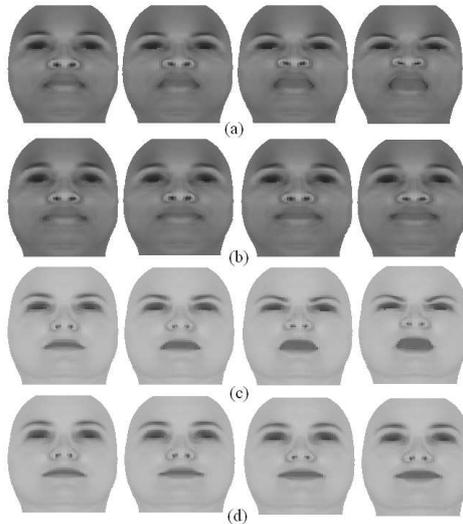


Figure 15: Faces representing the anger and smile expressions for individuals from two ethnic groups: (a) African-Anger, (b) African-Smile, (c) European-Anger, (d) European-Smile

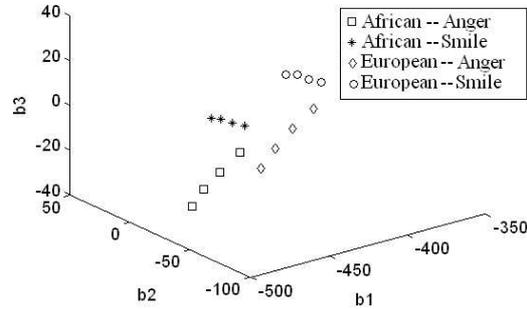


Figure 16: Representation of the first three elements of the shape space vectors for the faces from the Figure 15

6 Conclusions

In this paper, a method for a compact representation of 3-D faces has been presented. The novel aspects of the method include the use of B-splines for surface fitting of whole 3-D faces; the development of the B-spline control points based SSM, and the idea of using low dimensional shape space vectors as an efficient descriptor of facial expressions and facial surface deformations.

The effectiveness of the proposed method has been demonstrated by tests carried out on three synthetic data sets. It is shown that different types of facial expressions produce well separated low dimensional shape space vectors. The faces from different ethnic groups with the same facial expression produce shape space vectors which are also well separated, and that facial expressions belonging to the same category (with different degrees of expressiveness) are indicated by the positions of shape space vectors along the same direction. These results indicate that the proposed method can offer efficient and effective 3-D face representation which can handle large face variations and subtle expression differences.

7 Future work

This paper describes initial results and as such they are not comprehensive or complete. There is more work to be done to evaluate the proposed methodology as well as further develop the algorithm. The most obvious omission in this paper is lack of results obtained for real data. Although some results for real data have been obtained, they are very limited and therefore authors decided not to include them in this paper. Recently authors acquired a 3-D facial expression database (BU-3DFEDB) of real human faces [22]. This will be used to evaluate further the proposed methodology. Authors also plan to investigate to what extent the database of synthetic faces can replace the database constructed from real faces. This would be important for practical applicability of the method as construction of 3-D databases is still an expensive proposition. The authors also plan to extend the method by adopting a multi-resolution registration method, possibly by using one of subdivision schemes, to improve robustness and speed up computation time. Further improvements of the method will probably include building a hierarchical system where firstly the face

type is decided upon and subsequently the facial expression is searched using a statistical shape model build from the facial expression database constructed for a specific face type.

8 Acknowledgements

Bogdan Matuszewski would like to acknowledge the Vision, Video, and Graphics Network for supporting his visit in the Computer Vision and Image Analysis group at Heriot-Watt University which to some extent inspired the work described in this paper. Additionally he would like to express his gratitude to all the staff at Heriot-Watt and in particular to Prof. Emanuele Trucco, Dr Craig Robertson, and Ms Spela Ivekovic for making him feel at home and for many stimulating discussions during his stay there.

References

- [1] Bouchra Abboud, Franck Davoine, and Mo Dang. Facial expression recognition and synthesis based on an appearance model. *Signal Processing: Image Communication*, 19(8):727–740, 2004.
- [2] J. Ahlberg. An active model for facial feature tracking. In *Int. Workshop on Image Analysis for Multimedia Interactive Services.*, 2001.
- [3] K. S. Arun and T. S. Huang. Least-square fitting of two 3-d point sets. *IEEE TVGC.*, 9(5):698–700, 1987.
- [4] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. PAMI.*, 14(2):239–256, 1992.
- [5] Andrew Blake and Michael Isard. *Active Contours*. Springer-Verlag Berlin and Heidelberg, 1998.
- [6] Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, W. Richard Fright, Bruce C. McCallum, and Tim R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 67–76, 2001.
- [7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.
- [8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, 1995.
- [9] P. Eisert and B. Girod. Analyzing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications*, 18(5):70–78, 1998.
- [10] M. Hoch, G. Fleischmann, and B. Girod. Modeling and animation of facial expressions based on b-splines. *The Visual Computer*, 11(2):87–95, 1994.
- [11] InSpeck, www.inspeck.com. *InSpeckEM*, 2003.
- [12] Sumedha Kshirsagar, Stephane Garchery, and Nadia Magnenat-Thalmann. Feature point based mesh deformation applied to mpeg-4 facial animation. In *Proceedings of the IFIP, Workshop on Deformable Avatars*, pages 24–34, 2001.
- [13] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3).
- [14] J. J. Lien, T. Kanade, J. F. Cohn, and C. C. Li. Automated facial expression recognition based on facial action units. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 390, 1998.
- [15] C. L. Lisetti and D. J. Schiano. Automatic facial expression interpretation: Where human-computer interaction, artificial intelligence and cognitive science intersect. *Pragmatics and Cognition (Special Issue on Facial Information Processing: A Multidisciplinary Perspective)*, 8(1):185–235, 2000.
- [16] Xiaoguang Lu, Anil K. Jain, and Dirk Colbry. Matching 2.5d face scans to 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):31–43, 2006.
- [17] Biplab Sarkar and Chia-Hsiang Menq. Parameter optimization in approximating curves and surfaces to measurement data. *Comput. Aided Geom. Des.*, 8(4):267–290, 1991.

- [18] Caifeng Shan, Shaogang Gong, and P.W. McOwan. Robust facial expression recognition using local binary patterns. In *IEEE International Conference on Image Processing 2005*, pages 11–14, 2005.
- [19] Singular Inversions, www.facegen.com. *FaceGen Modeller*, 2003.
- [20] S. Umeyama. Least-square estimation of transformation parameters between two point patterns. *IEEE TVGC.*, 13(4):376–380, 1991.
- [21] Yang Wang, Xiaolei Huang, Chan-Su Lee, Song Zhang, and Zhiguo Li. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *Int. Workshop on EUROGRAPHICS 2004*, 2004.
- [22] Lijun Yin, Xiaozhou Wei, Jun Wang, and Matthew J. Rosato. A 3d facial expression database for facial behaviour research. In *FGR06: IEEE 7th International Conference on Automatic Face and Gesture Recognition*, pages 211–216, 2006.