Bilinear Models for 3D Face and Facial Expression Recognition

Iordanis Mpiperis, Sotiris Malassiotis and Michael G. Strintzis, *Fellow, IEEE*

***Abstract*— In this paper, we explore bilinear models for jointly addressing 3D face and facial expression recognition. An elastically deformable model algorithm that establishes correspondence among a set of faces is proposed first and then bilinear models that decouple the identity and facial expression factors are constructed. Fitting these models to unknown faces enables us to perform face recognition invariant to facial expressions and facial expression recognition with unknown identity. A quantitative evaluation of the proposed technique is conducted on the publicly available BU-3DFE face database in comparison with Wang *et al.*'s work [1] on facial expression recognition and our previous work [2] on face recognition. Experimental results demonstrate an overall 90.5% facial expression recognition rate and an 86% rank-1 face recognition rate.**

***Index Terms*— 3D facial expression recognition, 3D face recognition, elastically deformable model, bilinear model.**

I. Introduction

THE 3D geometry of the face conveys valuable information on the identity as was recently demonstrated by 3D face recognition research [3], [4]. But in addition, deformation of the facial surface caused by facial expressions may also provide cues useful to automatic facial expression classification.

Traditionally, researchers have examined these problems separately, e.g. using neutral images for face recognition and known identity for facial expressions. In practice however such decoupling does not exist. Thus an ideal 3D face analysis system should be able to recognize simultaneously (a) faces across any possible expression and (b) facial expressions and their intensity independently of identity, gender and ethnicity. Also, in order to be applicable under real life conditions, this face analyzer should meet additional requirements such as full automation in all stages of processing from 3D data acquisition to face and facial expression classification, near-real time response and robustness to head pose, aging effects and partial or self occlusion.

In this paper we present a technique for joint 3D identity-invariant facial expression recognition and expression-invariant face recognition (Fig. 1). First, we present a novel model-based framework for establishing correspondence among 3D point clouds of different faces. This allows us to create subsequently bilinear models that decouple expression and identity factors. A bootstrap set of faces is used to tune an asymmetric bilinear model which is incorporated in a probabilistic framework for the recognition of the six prototypic expressions proposed by Ekman *et al.* [5]. Face recognition on the other hand is performed by "modulating" the probe surface to resemble the expression depicted in the associated gallery image. This allows expression-invariant face recognition. Modulation is done using the expression and identity bilinear model. Finally, we present results evaluating our algorithms using the BU-3DFE[1] face database [6], and demonstrate comparisons with our previous work [2] on 3D face recognition and Wang *et al.*'s work [1] on 3D facial expression recognition.

I. Mpiperis and M. G. Strintzis are with the Centre for Research and Technology Hellas/Informatics and Telematics Institute, Thessaloniki, Greece; and the Information Processing Laboratory of Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Greece (e-mail: iordanis@iti.gr; strintzi@iti.gr).

S. Malassiotis is with the Centre for Research and Technology Hellas/Informatics and Telematics Institute (e-mail: malasiot@iti.gr).

A. Related work

Over the past three decades, face and facial expression recognition have received growing interest within the computer vision community. Most works in the literature assume neutral expression for 3D face recognition, e.g. [7]–[9], and use 2D imagery for facial expression recognition. Extensive surveys on 3D face recognition and facial expression recognition from 2D images and video may be found in [3] and [10] respectively. Here, we focus on research performed recently towards expression-invariant 3D face recognition and facial expression recognition from 3D surfaces.

To cope with facial expressions, the majority of techniques in the literature detect regions of the face that are affected by facial expressions and then try to minimize their contribution to similarity computation. Other techniques use expression invariant features or representations.

In [11] and [12] the authors use an Annotated Face Model which is deformed elastically to fit each face thus allowing the annotation of its different anatomical areas like nose, eyes, mouth, etc. To account for expressions, the authors then classify faces using wavelet coefficients representing face areas that remain unaffected by facial expressions, such as eyes and nose. However, the best recognition rate is achieved when the whole face is used, which implies that rejection of deformable parts of the face leads to loss of valuable discriminative information. Similarly, Chang *et al.* [13] follow a multi-region technique in which multiple overlapping regions around the nose are matched using the Iterative Closest Point algorithm (ICP). This approach too suffers from the disadvantage that deformable parts of the face that still encompass discriminative information are rejected during matching.

[1]The database is available at `http://www.cs.binghamton.edu/~lijun/Research/3DFE/3DFE_Analysis.html`

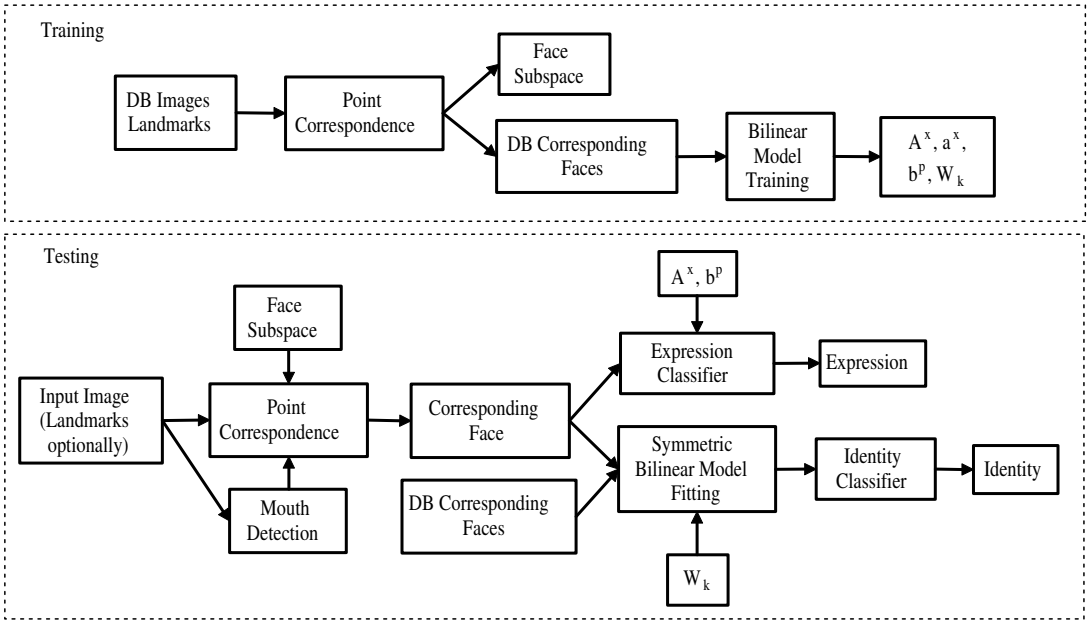

Fig. 1. Flow chart of the proposed algorithm for joint 3D face and facial expression recognition.

On the other hand, Li *et al.* [14] classify faces using expression-invariant descriptors that are based on surface curvature, geodesic distance and attributes of a 3D mesh fitted to the face. The descriptors are weighted appropriately to form a final face signature that is used to identify faces. Bronstein *et al.* [15] use multi-dimensional scaling on pair-wise geodesic distances to embed the surface to a 4D sphere where classification is performed on the basis of normalized moments. In the same spirit, we used a geodesic polar parameterization of the facial surface to construct expression-invariant attribute images [2] which can be classified following a variety of 2D classification techniques. The approaches [2], [14], [15] depend on the assumption that facial skin deforms isometrically, which is not valid in case of extreme expressions. In addition, the computation of expression-invariant features using curvature and geodesic distance is problematic because of their sensitivity to noise, which is also magnified by the approximate character of the isometric modelling of facial deformations.

Although great effort has been directed towards 3D face recognition, not many works have examined 3D facial expression recognition. Instead, most studies on automatic facial expression recognition are based on still images and image sequences [16]–[18]. The only work using purely 3D information we are aware of is that of Wang *et al.* [1]. In that work, expression classification is based on the distribution of geometric descriptors defined on different regions of the facial surface which are delimited according to the neuro-anatomic knowledge of the configuration of facial muscles and their dynamics. In each region, surface points are labeled according to their principal curvatures and then histograms of the labels are used to classify facial expressions. However, this technique is also limited by the need of computation of curvature features which may be problematic as we have already described.

All of the above techniques treat 3D face recognition and expression recognition as two separate problems. To the best of our knowledge, there are no reported studies about joint expression-invariant facial identity recognition and identity-invariant facial expression recognition based on 3D information. A few researchers have addressed the problem using 2D images by trying to encode identity and expression variability of facial appearance in independent control parameters which are then used for recognition. Vasilescu *et al.* [19] use the N-mode SVD tensor decomposition on face images to separate the influence of identity, pose, illumination and expression, while Wang *et al.* [20] use Higher-Order SVD (Singular Value Decomposition) to recognize and synthesize facial expressions. Bilinear models proposed by Tenenbaum *et al.* [21], [22] offer an efficient way for modeling bi-factor interactions, since they combine simplicity in training and implementation with capability of capturing subtle interactions between factors. As such, bilinear models are used in this work to model the 3D facial surface as the interaction of expression and identity component. After separating the parameters which control expression and those which control identity, joint expression-invariant face recognition and identity-invariant expression recognition is efficiently achieved.

Apart from presenting a novel unified framework for 3D face and facial expression recognition, this work introduces several contributions:

- We propose using bilinear models for handling jointly identity and expression contribution to facial appearance and we provide a generic solution for the minimization involved in bilinear model training. The proposed technique is applicable even with incomplete data sets in contrast to the conventional SVD approach which is intended for evenly distributed training data.
- We present a novel technique for establishing point corre-

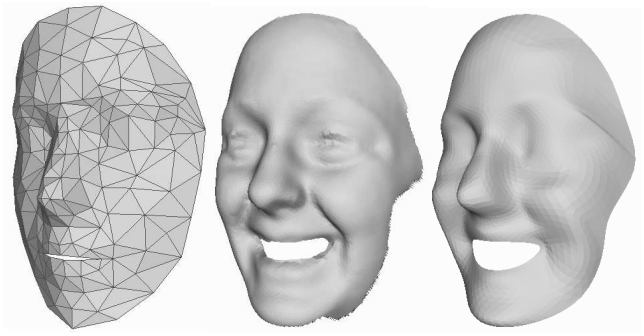

Fig. 2. Fitting base-mesh to a surface. a: base-mesh, b: original surface, c: base-mesh fitted to the surface.

spondence among faces which is based on an elastically deformable 3D model and is achieved by solving a simple linear system of equations. Unlike other relevant techniques, we use both surface-to-model and model-to-surface distances during model deformation which leads to more plausible point correspondence.

- Correspondence is established automatically by building a low-dimensional face eigen-space, but at the expense of possible poor correspondence in the mouth region in case of expressions with widely open mouth. To handle this problem caused by the rough approximation of the face manifold, we detect landmarks that define the mouth boundary instead of using complex non-linear models as usual.

The rest of the paper is organized as follows: In Section II we present how elastically deformable models can be employed to resolve the problem of point correspondence and how the use of landmarks can improve model fitting. Bilinear models training and fitting are described in Section III while their application on facial expression recognition and face recognition is demonstrated respectively in Section IV and Section V. The proposed algorithm is evaluated in Section VI where experimental results are reported and conclusions are drawn.

II. Establishing point correspondence

A. Elastically deformable model

Our goal is to establish a point-to-point correspondence among 3D facial surfaces guaranteeing alignment among anatomical facial features. Since faces are represented as clouds of 3D points acquired by a 3D sensor, it is more convenient that this correspondence is established instead between surface models fitted to the point clouds. We achieve this by deforming a prototypic facial surface model (neutral expression, average identity) so that it resembles the expression/identity depicted by the point clouds. Once this deformation has been estimated, finding correspondences is straightforward.

In this work, face is modelled as a subdivision surface, similarly to [23]. A triangular 3D mesh M_0 with N vertices $\mathbf{v} = [\mathbf{v}_1^T \dots \mathbf{v}_N^T]^T$ is used, where $\mathbf{v}_i^T = [x_i \ y_i \ z_i]$ are the coordinates of each vertex. At each subdivision step each edge of the mesh is split with the introduction of new vertices using linear subdivision rules[2]. Thus each triangle is subdivided into 4 sub-triangles and the mesh becomes more dense. Each vertex of the resulting 3D mesh may be written as a linear combination of the vertices of the previous level mesh and eventually of the initial mesh M_0. That is

$$\mathbf{v}^{(n)} = \mathbf{H}^{(n)}\mathbf{v} \tag{1}$$

where $\mathbf{v}^{(n)}$ is the set of vertex coordinates at level n of subdivision and $\mathbf{H}^{(n)}$ is a matrix which may be easily computed given the subdivision rule. After an infinite number of subdivisions the mesh converges to a continuous smooth surface which is a function of the initial mesh and the subdivision rule. In practice however, we do not have to make infinite subdivisions, since after a few levels (e.g. 3 in our experiments) the mesh becomes dense enough to approximate the subdivision surface. For notation simplicity, let the mesh $\widetilde{M}$ at the last level of subdivision that best approximates the subdivision surface be called subdivision-mesh and let $\widetilde{\mathbf{v}} = [\widetilde{\mathbf{v}}_1^T \dots \widetilde{\mathbf{v}}_S^T]^T$ denote its vertices. If also $\widetilde{\mathbf{H}}$ denotes the corresponding matrix with entries h_{ij}, then altogether we have

$$\widetilde{\mathbf{v}}_i = \sum_{j=1}^{N} h_{ij}\mathbf{v}_j \tag{2a}$$

$$\widetilde{\mathbf{v}} = \widetilde{\mathbf{H}}\mathbf{v} \tag{2b}$$

As is evident from the above, the geometry of the subdivision surface may be uniquely determined by defining the base-mesh M_0 vertices and topology (e.g. vertex connectivity).

Now let us return to the problem of fitting the subdivision surface to a 3D cloud of points. To make the fit anatomically valid, a set of landmarks corresponding to anatomically salient points of the face has to be defined both on the surface and the cloud of points. Let $\mathbf{p}_i$, $i = 1 \dots K$ denote the points of the cloud and $\mathbf{y}_i$, $i = 1 \dots L$ the associated landmarks, e.g. point $\mathbf{y}_1$ corresponds to the left eye leftmost point, $\mathbf{y}_2$ to the left eye rightmost point and so on. Similarly we select a subset $\mathcal{M}_0$ of vertices on the base-mesh M_0 that anatomically correspond to the L landmarks. According to the subdivision rule, the coordinates of vertices in M_0 remain untouched on the subdivision-mesh but are indexed differently. Thus, we can easily define a table $c(i)$ that maps each landmark index i to the corresponding vertex index of $\mathcal{M}_0$.

Fitting the subdivision surface to the cloud of 3D points is formulated as an energy minimization problem. We define a deformation energy which consists of terms giving rise to opposed forces between the subdivision surface and the cloud of points. The interaction of these forces deforms the subdivision surface or equivalently displaces the vertices of the base-mesh which control the surface, until equilibrium is established. The terms comprising the deformation energy are defined so that certain intuitively reasonable criteria are met.

The deformation should obey the a priori known correspondences between landmarks and associated mesh vertices. Therefore, the first term of the deformation energy minimizes

[2]We have used the Loop subdivision scheme [23].

the distance of each landmark from its corresponding vertex in $\mathcal{M}_0$. That is

$$E_c = \sum_{i=1}^{L} \left(\mathbf{y}_i - \mathbf{v}_{c(i)}\right)^2 \tag{3}$$

The contribution of this term is important in the first stages of optimization because it drives the minimization close to the global minimum avoiding getting trapped in local minima of the objective function.

The final form of the subdivision surface should also be as close as possible to the original surface represented by the cloud of points. Therefore the distance between the surface and the cloud should be minimized. We formulate two terms for this minimization, one for the distance directed from the subdivision surface to the point cloud, that is between each vertex of the subdivision-mesh and the nearest point of the cloud, and one for the distance with reverse direction. The use of two terms leads to two force fields between the surface and the point cloud, which are approximately similar in flat regions of the face but quite different in regions of high curvature (see Fig. 3). The resultant force field is smoother than both of them separately and establishes anatomically more plausible correspondence.

If we define function $sc(i)$ which returns the index j of the point $\mathbf{p}_j$ nearest to the $\widetilde{M}$ vertex i and function $cs(i)$ that returns inversely the index of the $\widetilde{M}$ vertex nearest to point $\mathbf{p}_i$, we can write the energy terms described above as

$$E_{sc} = \sum_{i=1}^{S} \left(\widetilde{\mathbf{v}}_i - \mathbf{p}_{sc(i)}\right)^2 = \sum_{i=1}^{S} \left(\sum_{j=1}^{N} h_{ij}\mathbf{v}_j - \mathbf{p}_{sc(i)} \right)^2 \tag{4}$$

$$E_{cs} = \sum_{i=1}^{K} \left(\mathbf{p}_i - \widetilde{\mathbf{v}}_{cs(i)}\right)^2 = \sum_{i=1}^{K} \left(\mathbf{p}_i - \sum_{j=1}^{N} h_{cs(i)j}\mathbf{v}_j \right)^2 \tag{5}$$

If the subdivision-mesh deformation is based solely on the correspondence between landmarks and the proximity to the cloud of points, the result will be a rough mesh with folded triangles. This is because the forces that act on the mesh may attract its vertices to disparate positions and thus fold the triangles. This problem can be overcome by posing a smoothness constraint to make the underconstrained optimization tractable. The smoothness is defined as a measure of the elastic energy of the base-mesh which penalizes non-parallel displacements of the edges and it is given by the equation

$$E_e = \sum_{i=1}^{N} \frac{1}{N_i} \sum_{j \in \mathcal{N}_i} \left(\mathbf{v}_i - \mathbf{v}_j - \mathbf{v}_i^0 + \mathbf{v}_j^0\right)^2 \tag{6}$$

where $\mathcal{N}_i$ is the set of $\mathbf{v}_i$'s neighbors, N_i is its cardinality and $\mathbf{v}_i^0$, $\mathbf{v}_j^0$ are the initial positions of the vertices.

The deformation energy whose minimization is sought, is defined as the weighted sum of the above energy terms

$$E_{def} = \lambda_1 E_c + \lambda_2 E_{sc} + \lambda_3 E_{cs} + \lambda_4 E_e \tag{7}$$

The coordinates of mesh M_0 vertices $\mathbf{v}_i$ that minimize E_{def} can be found by differentiating Eq. 7 with respect to $\mathbf{v}_i$ and setting the partial derivatives equal to zero. Differentiation leads to a linear system which can be solved easily. However, we may dispense with differentiation by writing E_{def} in matrix notation (see Appendix) and show that E_{def} is minimized by the solution of the overdetermined linear system Eq. 8 solved using singular value decomposition.

$$\begin{bmatrix} \mathbf{\Phi_t} \\ \mathbf{H} \\ \mathbf{H_{cs}} \\ \mathbf{\Psi_t} \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{\Phi_t t} \\ \mathbf{p_{sc}} \\ \mathbf{p} \\ \mathbf{\Psi_t v^0} \end{bmatrix} \tag{8}$$

Matrices $\mathbf{\Phi_t}$ and $\mathbf{\Psi_t}$ show whether a base-mesh vertex corresponds to a landmark and which is the neighborhood of each vertex respectively. $\mathbf{t}$ is a vector formed by landmarks coordinates while $\mathbf{v}^0$ is the vector of base-mesh vertex initial positions. $\mathbf{H}$, $\mathbf{H_{cs}}$, $\mathbf{p_{sc}}$ and $\mathbf{p}$ are used to define which vertices ($\mathbf{H}$, $\mathbf{H_{cs}}$) correspond to which points of the cloud ($\mathbf{p_{sc}}$, $\mathbf{p}$). (More details are given in the Appendix.)

As we have already described, vertex displacement is governed by two opposed force fields: (a) forces stemming from E_c, E_{sc} and E_{cs}, which attract the subdivision-mesh towards the cloud of points, and (b) forces stemming from E_e, which try to keep vertices to their initial positions. However, the forces due to E_{sc} and E_{cs} attract vertices towards the nearest points of the cloud instead of the anatomically corresponding points. This is not a problem if the mesh is relatively close to the cloud, since nearest points and anatomically corresponding points almost coincide. But if the mesh is relatively far from the cloud, vertices may be displaced so that anatomically erroneous correspondence is established. To overcome this problem, we iterate the minimization process several times letting vertices move progressively until they converge to a final position. Thereby, at the k-th iteration, vertices are updated according to

$$\mathbf{v}[k] = (1-\eta)\mathbf{v}[k-1] + \eta\hat{\mathbf{v}}[k] \tag{9}$$

where η is the step size usually chosen in $[0.2, 0.8]$ and $\hat{\mathbf{v}}[k]$ is the solution of the system Eq. 8 at the k-th iteration. We note that $\mathbf{H_{cs}}$, $\mathbf{p_{sc}}$ and $\mathbf{v^0} = \mathbf{v}[k-1]$ in Eq. 8 vary at each iteration. Finally, since the most time consuming part of the minimization is searching for nearest points according to Eq. 4 and Eq. 5, a space partitioning technique, kD-trees [24], is applied to accelerate the optimization. An illustration of mesh fitting can be seen in Fig. 2.

The formulation of the point-to-point correspondence problem as an energy minimization problem described by equations Eq. 7 and Eq. 8 has several advantages over other approaches reported in literature. In [25], the authors solve the problem of 3D correspondence using the optical flow between the associated color images. Apart from the need of texture, the color images should also be parameterized in the same cylindrical coordinate frame which is another problem of its own. Therefore this technique cannot be applied when the 3D data are textureless and in the form of a point cloud as in our case. A possible alternative for our method is the use of elastically adaptive deformable models, proposed by Metaxas [26] and used by Kakadiaris *et al.* [12] and Passalis *et al.* [11] for face recognition. However, this method involves the integration of a system of second order differential equations,

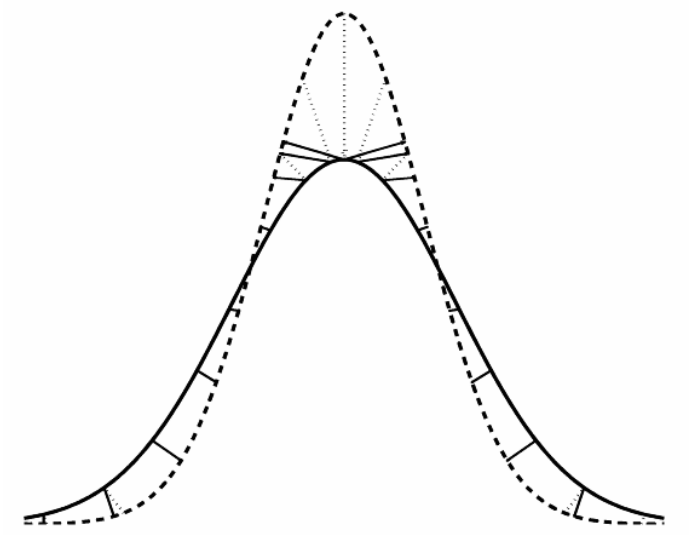

Fig. 3. Correspondence based on proximity. The deformable model is depicted with the continuous curve while the cloud of points with the dashed one. Arrows show the point which is nearest to their start. Dotted arrows depict the cloud-to-model correspondences, while continuous arrows depict the model-to-cloud correspondence. The usefulness of using both sets of directed distances is apparent in the high curvature region, where there is a discontinuity in the model-to-cloud set of correspondences (continuous arrows).

which is much more demanding than the solution of a linear system as we propose.

Allen *et al.* [27] present the method most similar to ours, even though it is intended for establishment of point correspondence between whole human body scans. They also use landmarks and minimize the distance between the model and the surface in hand using a smoothness term for regularization. The main difference however is that they use only distances directed from the model to the cloud of points, while we use also the distances with the inverse direction. This is important in the early stages of the optimization when the model is still far from the cloud (see Fig. 3). In this case, the model-to-cloud distances (continuous arrows in Fig. 3) may have discontinuities in regions of high curvature and thus give rise to anatomically inaccurate correspondences. Using also the other set of directed distances (dotted arrows in Fig. 3) the resultant vector field becomes smoother and helps the optimization avoid getting trapped in local minima.

B. Subspace-guided elastically deformable model

In most cases of practical interest, the number of facial landmarks which can be detected automatically and relatively accurately is limited. Furthermore, facial surface captured by depth acquisition devices usually includes extraneous parts of the body, such as the neck and the upper head. In such a case, mesh fitting cannot be based on forces stemming from landmarks correspondence and surface-to-mesh distance. Recovering the surface deformation in this case is an ill-posed problem which however becomes more tractable by assuming that the deformations lie on a subspace of low dimensionality. This subspace may be estimated from training data by means of Principal Component Analysis (PCA). Once the faces of the training set have been set in correspondence following the procedure described in the previous section, base-mesh vertices of any novel face may be written as a linear combination of eigen-meshes

$$\mathbf{v} = \mathbf{U}\mathbf{q} + \overline{\mathbf{v}} \tag{10}$$

where $\mathbf{U}$ is the truncated matrix which describes the principal modes of deformation, $\mathbf{q}$ is the control vector and $\overline{\mathbf{v}}$ is the mean vector of aligned vertex coordinates (i.e. mean face).

Now, by replacing Eq. 10 in Eq. 8 (the terms E_c and E_{cs} are excluded), we may rewrite the deformation energy as a function of $\mathbf{q}$. Similarly to the analysis in the previous section, the control vector $\mathbf{q}$ which minimizes E_{def} is the solution of the linear system

$$\begin{bmatrix}\mathbf{H}\\ \mathbf{\Psi_t}\end{bmatrix}\mathbf{U}\mathbf{q} = \begin{bmatrix}\mathbf{p_{sc}}\\ \mathbf{\Psi_t v^0}\end{bmatrix} - \begin{bmatrix}\mathbf{H}\\ \mathbf{\Psi_t}\end{bmatrix}\overline{\mathbf{v}} \tag{11}$$

Our experiments showed that this approach converges to the global minimum of the deformation energy most of the time. It was also observed that failure usually occurs if the person displays an expression with widely open mouth like when displaying surprise. Mesh fitting in this case may result in a deformed closed mouth (Fig. 4), which implies that the low dimensional approximation of the face manifold is so rough in the region of mouth that optimization is trapped in a local minimum. To overcome this problem, one can use a more complex model to capture the structure of the face manifold. For instance, in [28] the authors use active appearance models [29] combined with a multi-layer perceptron to establish correspondence among 2D face images. This however makes the problem non-linear. Thus in this paper we have experimented with a simpler technique that relies on the detection of the mouth boundary.

Mouth boundary is detected using depth and curvature information following an approach similar to that of [30]. A vertical profile curve, as illustrated in Fig. 4, is used to detect the upper and lower points of the mouth contour exploiting the fact that these points are local extrema of the profile curve. Then, a bounding box of the mouth is defined where the mean curvature $H(x,y)$ of the surface $S(x,y)$ is calculated. Mean curvature is used because it gets opposite signs on the lip ridges and the mouth hollow assuming that the z axis is parallel to the gaze direction. A measure of cornerness $C(x,y)$ is also computed to identify the horizontal outermost points of the mouth contour, that is the corners of the lips. Cornerness is defined by the equation

$$C(x,y) = \frac{\frac{\partial^2 S}{\partial x^2}\frac{\partial^2 S}{\partial y^2} - \left(\frac{\partial^2 S}{\partial x \partial y}\right)^2}{\frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2}} \tag{12}$$

which is derived by the Harris corner detector [30]. Mean curvature and cornerness are then fused with a mini-max rule to a single index that indicates the extent to which points of the surface belong to the boundary. The boundary is finally formed by fitting a spline curve through points with indices above a certain threshold set empirically. Once the boundary has been detected, the associated landmarks are included in the

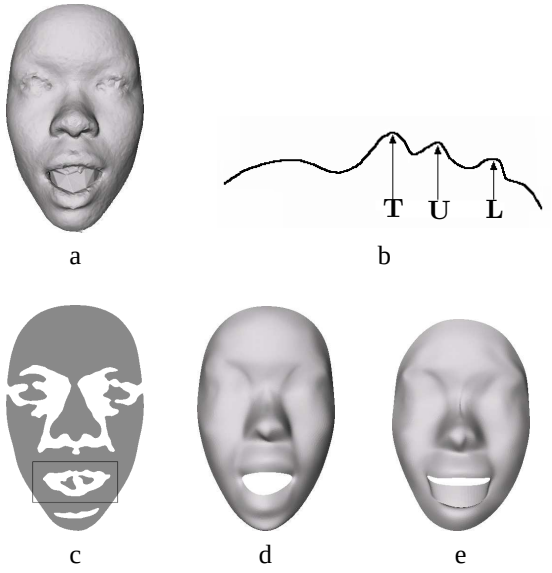

Fig. 4. Mouth boundary detection. a) The original face surface. b) A vertical profile used to define a mouth bounding box. The tip of the nose, the upper and lower points of the mouth are denoted by **T**, **U**, and **L** respectively. c) The sign of mean curvature H of the surface. The sign of H along with a measure of cornerness inside the mouth bounding box are used to detect the mouth boundary and define the associated landmarks. d) The base-mesh fitted to the original surface using mouth associated landmarks. e) The base-mesh fitted to the original surface without mouth associated landmarks.

formulation of the deformation energy and the linear system in Eq. 11 is modified to

$$\begin{bmatrix} \mathbf{\Phi_t'} \\ \mathbf{H} \\ \mathbf{\Psi_t} \end{bmatrix} \mathbf{Uq} = \begin{bmatrix} \mathbf{\Phi_t' t'} \\ \mathbf{p_{sc}} \\ \mathbf{\Psi_t v^0} \end{bmatrix} - \begin{bmatrix} \mathbf{\Phi_t'} \\ \mathbf{H} \\ \mathbf{\Psi_t} \end{bmatrix} \overline{\mathbf{v}} \qquad (13)$$

Boundary detection relies on the quality of 3D data since it involves estimation of second order surface derivatives which is sensitive to noise. As expected, higher depth quality results in better detection. However, our experiments showed that even an approximate detection of the mouth boundary is enough for the minimization of the deformation energy to avoid local minima. Therefore, the proposed system is insensitive to small boundary mis-localizations.

In summary, the overall procedure for establishing correspondence among faces is as follows: During the training phase, correspondence among a set of annotated training 3D images is established. These faces are rigidly registered with each other using the ICP algorithm and then each facial point cloud is modeled by a subdivision surface. Modeling consists in finding the base-mesh vertices that minimize the deformation energy defined by Eq. 7 with minimization achieved by iterating Eq. 8 upon convergence. Once correspondence is established, principal component analysis is applied to learn the principal modes of base-mesh deformation.

In the test phase, for a novel unseen face, its mouth boundary has to be detected first. Then Eq. 13 is solved to obtain the control parameters $\mathbf{q}$ from which the base-mesh vertices $\mathbf{v}$ may be recovered (Eq. 10). Having fitted a subdivision surface to the new face, correspondence with all faces in the gallery is established, since all fitted surfaces originate from the deformation of an average face surface.

III. Modeling facial expression and identity variation

Once point correspondence is established, each facial surface can be represented by the vector $\mathbf{v}$ of the base-mesh vertices since it defines unambiguously the subdivision surface that approximates the cloud of facial points. Then, one may classify faces or facial expressions using typical pattern classification techniques that rely on distance metrics of vector spaces. However, this approach will result in degraded recognition performance, because such a representation cannot distinguish whether a certain shape of the face is attributed to the identity of the person or the expression he or she displays. For example, we cannot distinguish whether a puffy cheek comes from a fat person or it is due to a smiling expression. It becomes clear therefore, that we have to encode identity and expression in independent control parameters in order to be able to perform joint expression-invariant facial identity recognition and identity-invariant facial expression recognition.

In this work, we use bilinear models to capture the identity-expression structure of face appearance. Bilinear models are linear in either factor when the other is held constant and as such they share almost all the advantages of linear models: they are simple in structure, computation and implementation, they can be trained with well known algorithms and their complexity can be easily adjusted by their dimensionality as a compromise between exact reproduction of training data and generalization during testing [22]. Simultaneously and despite their simplicity, they can model subtle interactions by allowing factors to modulate each other's contribution multiplicatively.

We use two types of bilinear models, the symmetric and the asymmetric bilinear model, suitable for identity and expression recognition respectively. In case of face recognition, we neutralize the effect of expressions by deforming the matched faces so that they display a common expression (e.g. a neutral expression or the expression displayed by the gallery face). This is accomplished by modifying their expression control parameters which are extracted by fitting faces to a symmetric bilinear model. The asymmetric model on the contrary is used to perform identity-invariant expression recognition. After training the model with several subjects depicting different facial expressions, it is incorporated in a maximum likelihood classification framework which allows facial expression discrimination across identity.

In the following sections we describe in detail the symmetric and asymmetric bilinear model and we present a novel general solution of the minimization involved in their training.

A. Symmetric model

Let $\mathbf{v}^{xp}$ be the stacked column vector of the N base-mesh vertices of the facial surface of person p with expression x. The dimension of $\mathbf{v}^{xp}$ which is $3N$, is denoted by K for simplicity. Then each component v_k^{xp} is given by the general bilinear form [21], [22]

$$\mathrm{v}_k^{xp} = \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ijk} a_i^x b_j^p \qquad (14)$$

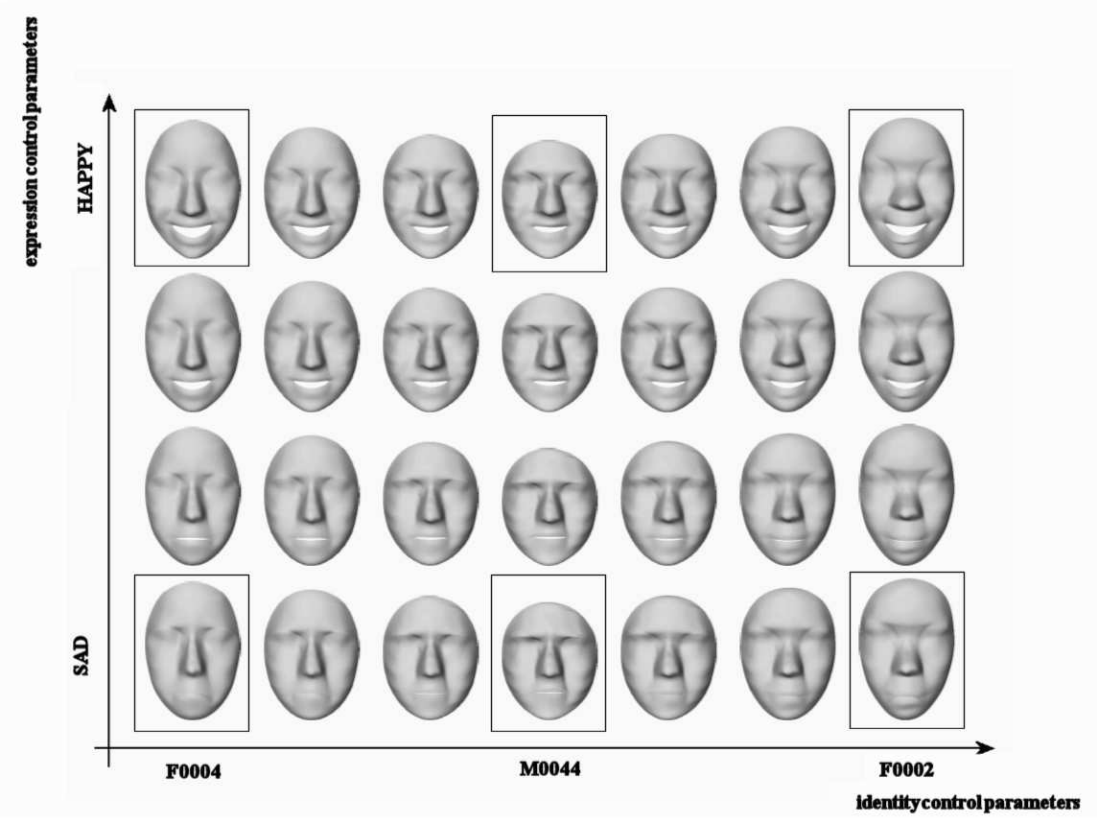

Fig. 5. Deformation of facial surface across expression and identity control parameters. Inside boxes are surfaces which are stored in the BU-3DFE database. Specifically, it is subjects F0004, M0044 and F0002 displaying the *sad* and *happy* expressions. The rest surfaces have been generated by linear interpolation of expression and identity control parameters of the former surfaces.

Here, a_i^x and b_j^p are the control parameters which control expression and identity respectively, while w_{ijk} are the coefficients which model the interaction of the factors and we try to estimate from the training set. The above equation can be written equivalently as

$$\mathbf{v}^{xp} = \sum_{i=1}^{I}\sum_{j=1}^{J} \mathbf{w}_{ij} a_i^x b_j^p \tag{15}$$

where $\mathbf{w}_{ij} = [w_{ij1} \dots w_{ijk} \dots w_{ijK}]^T$ are stacked vectors of dimension $K = 3N$. Now, it is clear that $\mathbf{v}^{xp}$ is the bilinear combination of basis vectors $\mathbf{w}_{ij}$ which are mixed by control coefficients a_i^x and b_j^p. The above equation also shows that control parameters weigh symmetrically the basis vectors and therefore this model is referred to as symmetric bilinear model.

Let us assume that there exist T faces in our database belonging to T_p individuals and each one depicting one of T_x possible expressions. Our goal is to find the interaction coefficients w_{ijk} and the control parameters a_i^x and b_j^p for each individual-expression couple. Using matrix notation Eq. 14 is simplified to

$$\mathrm{v}_k^{xp} = \mathbf{a}^{xT} \mathbf{W}_k \mathbf{b}^p \tag{16}$$

where $\mathbf{a}^x = [a_1^x \dots a_I^x]^T$, $\mathbf{b}^p = [b_1^p \dots b_J^p]^T$ and $\mathbf{W}_k(i,j) = w_{ijk}$. Since in practice a facial expression from the T_x possible ones may not be available for some persons, we also define the zero-one function $h_{xp}[t]$ which is one if the t-th face $\mathbf{v}[t]$ belongs to individual p with expression x. Unknown coefficients arise from the minimization of the total squared error [21]

$$E_s = \sum_{t=1}^{T}\sum_{x=1}^{T_x}\sum_{p=1}^{T_p}\sum_{k=1}^{K} h_{xp}[t](\mathrm{v}_k[t] - \mathbf{a}^{xT}\mathbf{W}_k\mathbf{b}^p)^2 \tag{17}$$

By differentiating and setting the partial derivatives equal to zero we end up with the system of equations

$$\mathbf{a}^x = \left(\sum_{p=1}^{T_p}\sum_{k=1}^{K} n_{xp}\mathbf{W}_k\mathbf{b}^p\mathbf{b}^{pT}\mathbf{W}_k^T\right)^{-1}\left(\sum_{p=1}^{T_p}\sum_{k=1}^{K} m_k^{xp}\mathbf{W}_k\mathbf{b}^p\right) \tag{18}$$

$$\mathbf{b}^p = \left(\sum_{x=1}^{T_x}\sum_{k=1}^{K} n_{xp}\mathbf{W}_k^T\mathbf{a}^x\mathbf{a}^{xT}\mathbf{W}_k\right)^{-1}\left(\sum_{x=1}^{T_x}\sum_{k=1}^{K} m_k^{xp}\mathbf{W}_k^T\mathbf{a}^x\right) \tag{19}$$

$$\sum_{x=1}^{T_x}\sum_{p=1}^{T_p} n_{xp}\mathbf{a}^x\mathbf{a}^{xT}\mathbf{W}_k\mathbf{b}^p\mathbf{b}^{pT} = \sum_{x=1}^{T_x}\sum_{p=1}^{T_p} m_k^{xp}\mathbf{a}^x\mathbf{b}^{pT} \tag{20}$$

where $n_{xp} = \sum_{t=1}^{T} h_{xp}[t]$ is the number of training faces which belong to subject p displaying expression x and $\mathbf{m}^{xp}$ is their sum, $\mathbf{m}^{xp} = [m_1^{xp} \dots m_k^{xp} \dots m_K^{xp}]^T = \sum_{t=1}^{T} h_{xp}[t]\mathbf{v}[t]$. Eq. 20 is a general Sylvester equation [31] which can be

rewritten as

$$\mathrm{vec}(\mathbf{W}_k) = \left(\sum_{x=1}^{T_x} \sum_{p=1}^{T_p} n_{xp} \mathbf{b}^p \mathbf{b}^{pT} \otimes \mathbf{a}^x \mathbf{a}^{xT} \right)^{-1} \mathrm{vec} \left(\sum_{x=1}^{T_x} \sum_{p=1}^{T_p} m_k^{xp} \mathbf{a}^x \mathbf{b}^{pT} \right) \tag{21}$$

where $\otimes$ is the Kronecker product operator and $\mathrm{vec}(\cdot)$ is the matrix vectorization operator which stacks the columns of the matrix.

Interaction matrices $\mathbf{W}_k$ and control vectors $\mathbf{a}^x$ and $\mathbf{b}^p$ may now be found by iterating equations Eq. 21, Eq. 18 and Eq. 19 respectively. To ensure the stability of the solution, updating is performed progressively according to the rule

$$\mathbf{X}[n] = (1-\eta)\mathbf{X}[n-1] + \eta \hat{\mathbf{X}}[n] \tag{22}$$

where η is the step size which is usually chosen in $[0.2, 0.8]$, $\mathbf{X}[n]$ stands for the final value of $\mathbf{a}^x$, $\mathbf{b}^p$ or $\mathbf{W}_k$ in the n-th iteration, and $\hat{\mathbf{X}}[n]$ stands for the value resulted from Eq. 18, Eq. 19 or Eq. 21 respectively.

It should be noted that convergence depends on the dimensionalities I of $\mathbf{a}^x$, and J of $\mathbf{b}^p$, which control the exactness of the training data reproduction. Convergence is guaranteed if I and J are less than or equal to T_x and T_p, the number of expressions and the number of individuals respectively. If I is equal to T_x and J is equal to T_p, training data are reproduced exactly, while more coarse but also more compact representations result if these dimensionalities are decreased.

The minimization of the total squared error through equations Eq. 18, Eq. 19 and Eq. 21 presented here differs from the optimization scheme proposed by Tenenbaum *et al.* in [22] and their previous work. There, minimization is achieved by means of Singular Value Decomposition (SVD) applied to the $(T_x K) \times T_p$ mean observation block matrix $\bar{\mathbf{V}}$,

$$\bar{\mathbf{V}} = \begin{bmatrix} \bar{\mathbf{v}}_{11} & \cdots & \bar{\mathbf{v}}_{1T_p} \\ \vdots & \ddots & \\ \bar{\mathbf{v}}_{T_x 1} & & \bar{\mathbf{v}}_{T_x T_p} \end{bmatrix} \tag{23}$$

where $\bar{\mathbf{v}}_{xp}$ is the mean vertex vector of the facial surfaces of subject p with expression x. This technique relies on evenly distributed data across expression and identity. In practice however, data may not be evenly distributed across expression and identity or even worse, there may not be available data for a particular expression-identity combination. Then $\bar{\mathbf{V}}$ will have some indeterminate entries and SVD will not be applicable. One remedy is filling the missing entries with the mean values of the appropriate expression and identity, but this substitution does not guarantee the global minimization of the total squared error and thus the best fit of the bilinear model to the training data[3]. In contrast, the proposed method may be applied directly without any assumptions for the data distribution, but at the expense of computational complexity.

An illustration of facial surface modeling using the symmetric bilinear model is shown in Fig. 5, where the surfaces have been generated by interpolating linearly the coefficients corresponding to two expressions and three identities.

[3]Generalizations of bilinear models using tensors, such as [19] and [32], face the same problem, since estimation of model parameters is done by applying singular value decomposition to matrices resulting from tensor mode-n flattening.

B. Asymmetric model

As already mentioned in the previous section, the symmetric model weighs symmetrically the coefficients which control identity and expression. This symmetry implies that the model can generalize on both directions, identity and expression, which is a desirable property for both face and expression recognition. However, there is a difference between these recognition tasks. In case of face recognition, it is usually supposed that unseen faces are free to display an unrestricted number of possible expressions. In case of expression recognition instead, the number of possible expressions is usually considered finite (most studies classify expressions to the six prototypic universal expressions proposed by Ekman [5]). In the latter case the symmetric model can be modified so that a better fit to expressions can be achieved thus improving recognition performance. This is accomplished by letting the interaction coefficients w_{ijk} vary with the expression control parameters a_i [22], that is

$$a_{kj}^x = \sum_{i=1}^{I} w_{ijk} a_i^x \tag{24}$$

Using the above definition and Eq. 14, the vector representation of the face is now given by

$$\mathrm{v}_k^{xp} = \sum_{j=1}^{J} a_{kj}^x b_j^p \tag{25a}$$

$$\mathbf{v}^{xp} = \mathbf{A}^x \mathbf{b}^p \tag{25b}$$

where now matrix $\mathbf{A}^x$ controls expression. The identity of the face is still controlled by vector $\mathbf{b}^p$.

Fitting the asymmetric model to the training data consists in finding the expression matrices $\mathbf{A}^x$ and identity vectors $\mathbf{b}^p$ that minimize the total squared error [21]

$$E_a = \sum_{t=1}^{T} \sum_{x=1}^{T_x} \sum_{p=1}^{T_p} h_{xp}(t) \left(\mathbf{v}(t) - \mathbf{A}^x \mathbf{b}^p \right)^2 \tag{26}$$

$\mathbf{A}^x$ and $\mathbf{b}^p$ are obtained by differentiating the error function and setting the partial derivatives equal to zero. Following this procedure, $\mathbf{A}^x$ and $\mathbf{b}^p$ should satisfy the system of equations [21]

$$\mathbf{A}^x = \left(\sum_{p=1}^{T_p} \mathbf{m}_{xp} \mathbf{b}^{pT} \right) \left(\sum_{p=1}^{T_p} n_{xp} \mathbf{b}^p \mathbf{b}^{pT} \right)^{-1} \tag{27}$$

$$\mathbf{b}^p = \left(\sum_{x=1}^{T_x} n_{xp} \mathbf{A}^{xT} \mathbf{A}^x \right)^{-1} \left(\sum_{x=1}^{T_x} \mathbf{A}^{xT} \mathbf{m}_{xp} \right) \tag{28}$$

which is finally solved by iterating the equations according to rule Eq. 22 until the values of $\mathbf{A}^x$ and $\mathbf{b}^p$ converge.

The exactness of training data reproduction is determined by the number of columns of matrix $\mathbf{A}^x$ or equivalently the dimensionality of vector $\mathbf{b}^p$. More exact reconstruction is achieved if more columns are used. Nevertheless, the number of columns should be restricted in order to avoid overfitting [22] and it must be less than the number of subjects in the training set so that the solution of equations Eq. 27 and Eq. 28 be feasible.

IV. Facial expression recognition

In this section we show how a facial expression classifier may be built on the basis of an asymmetric bilinear model fitted to a training set of faces.

A. Training

The input is a set of training faces annotated with salient facial points. This set should contain images of several persons depicting different facial expressions. First, anatomical correspondence among raw data is established by means of the elastically deformable base-mesh M_0 as explained in Section II-A. Then, the resulting model parameters are used to build the 3D face eigen-space so that newly seen faces may be processed without the need of facial landmarks.

An asymmetric bilinear model is then fitted to the deformable model parameters $\mathbf{v}_i$ of the training faces. That is, we estimate the expression control matrices $\mathbf{A}^x$ and identity control vectors $\mathbf{b}^p$ that minimize the total squared reconstruction error given by Eq. 26. This is done by iterating equations Eq. 27 and Eq. 28 until the relative change in the Frobenius norm of both $\mathbf{A}^x$ and $\mathbf{b}^p$ becomes less than a predefined constant threshold.

Estimated $\mathbf{A}^x$ and $\mathbf{b}^p$ are then used to build a Maximum Likelihood classifier. The goal is to estimate the likelihood of the surface model parameters $\mathbf{v}$ for each expression, that is the conditional probability density function (p.d.f.) $f(\mathbf{v}|x)$, where x is one of the prototypic expressions. To this end, we assume that the vertex vector $\mathbf{v}$ defining the facial surface of person p with expression x is a random vector with spherical gaussian p.d.f. centered at the prediction of the asymmetric bilinear model. That is

$$f(\mathbf{v}|p,x) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^K} e^{-\frac{1}{2\sigma^2}||\mathbf{v}-\mathbf{A}^x\mathbf{b}^p||^2} \tag{29}$$

where σ is the error variance. Using the Total Probability Theorem, the conditional p.d.f. of $\mathbf{v}$ assuming expression x may now be written as

$$f(\mathbf{v}|x) = \sum_{p=1}^{T_p} P(p) f(\mathbf{v}|p,x) \tag{30}$$

where $P(p)$ stands for the a priori probability of person p which may be considered constant for all subjects and equal to $\frac{1}{T_p}$.

B. Classification

The facial expression of a novel test face is classified simply by comparing the conditional p.d.f.'s of $\mathbf{v}$ (Eq. 30) for all expressions. First, the test surface is set into anatomical correspondence with all meshes in the training set. If landmarks are available for the test face then the landmark-guided deformation of the base-mesh explained in Section II-A is followed, otherwise the subspace-guided deformation described in Section II-B is applied. Once the surface model parameters $\mathbf{v}$ are found, the expression of the face is classified to the prototypic expression with the greatest likelihood, that is the expression x_i for which

$$f(\mathbf{v}|x_i) > f(\mathbf{v}|x_j) \;\; \forall\, x_j \neq x_i \tag{31}$$

V. Face recognition

Similarly to facial expression recognition, face recognition also involves a bootstrapping phase before classification. However, classification follows a different approach which consists in altering the expression of the probe face. This is accomplished by modulating the parameters of a symmetric bilinear model fitted to the probe face in order to force it to display the expression of every gallery face before matching. In the following we present in detail this procedure.

A. Bootstrapping

Since subjects to be classified are free to display various expressions, a bootstrap set of faces is used to train a symmetric bilinear model so that we may be able to generalize on any novel unseen identity or expression. The bootstrap set is also used for learning the mesh deformation eigen-space so that newly seen faces may be processed without the need of facial landmarks.

Bootstrap faces are first set into anatomical correspondence as described in Section II-A and the mesh eigen-space is then learnt by means of PCA. The training of the symmetric model starts by training an asymmetric bilinear model such as the one used for facial expression recognition. This is done to obtain initial values for the identity control vectors $\mathbf{b}^p$. After training the asymmetric model, vectors $\mathbf{b}^p$ are used to train an initial symmetric model following the SVD approach presented in [22]. Analytically, this is done by applying SVD to the mean observation block matrix defined in Eq. 23. By defining the vector transpose operator $[\,\cdot\,]^{VT}$, which transposes a matrix block-wise as shown in Fig. 6, the mean observation matrix may be written as

$$\bar{\mathbf{V}} = \left[\mathbf{W}^{VT}\mathbf{A}\right]^{VT}\mathbf{B} \tag{32}$$

$$\left[\bar{\mathbf{V}}\right]^{VT} = \left[\mathbf{W}\mathbf{B}\right]^{VT}\mathbf{A} \tag{33}$$

where matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{W}$ are comprised of $\mathbf{a}^x$, $\mathbf{b}^p$ and $\mathbf{w}_{ij}$ (Eq. 15) respectively,

$$\mathbf{A} = \left[\mathbf{a}^{x_1} \dots \mathbf{a}^{x_{T_x}}\right], \quad \mathbf{B} = \left[\mathbf{b}^{p_1} \dots \mathbf{b}^{p_{T_p}}\right], \quad \mathbf{W} = \begin{bmatrix} \mathbf{w}_{11} & \dots & \mathbf{w}_{1J} \\ \vdots & \ddots & \\ \mathbf{w}_{I1} & & \bar{\mathbf{v}}_{IJ} \end{bmatrix} \tag{34}$$

Matrix $\mathbf{B}$ is initialized by the identity control vectors estimated by the asymmetric model. Then, the SVD of $[\bar{\mathbf{V}}\mathbf{B}^T]^{VT} = \mathbf{R}\mathbf{S}\mathbf{Q}^T$ is computed and matrix $\mathbf{A}$ is updated by the first I rows of $\mathbf{Q}^T$. Similarly, the SVD of $[\bar{\mathbf{V}}^{VT}\mathbf{B}^T]^{VT} = \mathbf{R}'\mathbf{S}'\mathbf{Q}'^T$ is computed and matrix $\mathbf{B}$ is updated by the first J rows of $\mathbf{Q}'^T$. This constitutes one iteration of the error minimization algorithm whose convergence is guaranteed. Upon convergence, $\mathbf{W}$ results as $\mathbf{W} = \left[[\bar{\mathbf{V}}\mathbf{B}^T]^{VT}\mathbf{A}^T\right]^{VT}$. (For more details the reader is referred to [22].) By rearranging $\mathbf{W}$ we may compute interaction matrices $\mathbf{W}_k(i,j) = \mathbf{w}_{ij}(k)$. Then, final optimal parameters are found by iterating equations Eq. 18, Eq. 19 and Eq. 21 until convergence is achieved.

$$\begin{bmatrix} \begin{bmatrix}1\\2\end{bmatrix} & \begin{bmatrix}7\\8\end{bmatrix} \\ \begin{bmatrix}3\\4\end{bmatrix} & \begin{bmatrix}9\\10\end{bmatrix} \\ \begin{bmatrix}5\\6\end{bmatrix} & \begin{bmatrix}11\\12\end{bmatrix} \end{bmatrix}^{VT} = \begin{bmatrix} \begin{bmatrix}1\\2\end{bmatrix} & \begin{bmatrix}3\\4\end{bmatrix} & \begin{bmatrix}5\\6\end{bmatrix} \\ \begin{bmatrix}7\\8\end{bmatrix} & \begin{bmatrix}9\\10\end{bmatrix} & \begin{bmatrix}11\\12\end{bmatrix} \end{bmatrix}$$

Fig. 6. The vector transpose operator ($[\cdot]^{VT}$) acting on a matrix which is considered to consist of 2×1 blocks. The operator transposes the matrix block-wise.

B. Gallery image processing

Using the optimal mixing matrices $\mathbf{W}_k$ found from the bootstrap set, we extract for each gallery face its expression and identity control vectors. This is accomplished by minimizing the reconstruction squared error

$$E = \sum_{k=1}^{K} \left(\mathrm{v}_k - \mathbf{a}^{x^T} \mathbf{W}_k \mathbf{b}^p \right)^2 \tag{35}$$

Minimization is achieved similarly to the minimization of the total squared error in Eq. 17 during the training of the symmetric bilinear model. By differentiating Eq. 35 and setting partial derivatives equal to zero, control vectors are found by iterating the system of equations

$$\mathbf{a}^x = \left(\sum_{k=1}^{K} \mathbf{W}_k \mathbf{b}^p \mathbf{b}^{p^T} \mathbf{W}_k^T \right)^{-1} \left(\sum_{k=1}^{K} \mathrm{v}_k \mathbf{W}_k \mathbf{b}^p \right) \tag{36}$$

$$\mathbf{b}^p = \left(\sum_{k=1}^{K} \mathbf{W}_k{}^T \mathbf{a}^x \mathbf{a}^{x^T} \mathbf{W}_k \right)^{-1} \left(\sum_{k=1}^{K} \mathrm{v}_k \mathbf{W}_k^T \mathbf{a}^x \right) \tag{37}$$

Equations are iterated until the change in the norm of $\mathbf{a}^x$ and $\mathbf{b}^p$ becomes less than a threshold. The estimated expression control vectors of the gallery faces are subsequently used during classification to modify the expression displayed by the probe face.

C. Classification

The classification of a novel probe face starts by setting it into correspondence with the gallery faces according to Section II. If landmarks are available, the landmark-guided approach (Section II-A) is followed, otherwise the subspace-guided technique (Section II-B) is applied using the mesh eigen-space learnt from the bootstrap set. We note that we have made no assumptions about the expression of the gallery faces meaning that they are allowed to display various expressions possibly different from that of the probe faces. Then, to handle the influence of expression on surface matching, we force the probe face to display the expression of the gallery face before matching. Its expression control vector has to be extracted by fitting the symmetric bilinear model as in the case of the gallery faces. Then by substituting the expression control vector by the corresponding vector of the gallery face, we may reconstruct a new probe facial surface which depicts expression similar to that of the gallery face. If the new probe face is represented by the vertex vector $\mathbf{v}_p$ while the gallery face by the vertex vector $\mathbf{v}_g$, then expression free comparison between them may be established on the inverse of their vertex vectors' squared Euclidean distance,

$$d = ||\mathbf{v}_g - \mathbf{v}_p||^{-1} \tag{38}$$

VI. Performance evaluation

In this section, we evaluate the performance of bilinear models on the Binghamton University 3D Facial Expression (BU-3DFE) database [6] and we provide comparisons with Wang *et al.*'s work [1] on facial expression recognition and our previous work [2] on face recognition. BU-3DFE is preferred over other popular databases such as FRGC [33], because expressions are well-defined and distributed better.

BU-3DFE contains 100 subjects, 56 female and 44 male with a variety of ethnic/racial ancestries including white, black, east-asian, middle-east-asian, hispanic latino and others. The subjects display the six universal expressions of *anger*, *fear*, *disgust*, *happiness*, *sadness* and *surprise* in 4 levels of intensity, *low*, *middle*, *high* and *highest*. For each subject, there is also a 3D face scan with neutral expression thus resulting in a total number of 2,500 face scans in the database. 3D facial data are in the form of VRML models which contain about 13,000-21,000 polygons and are associated with a set of feature points located on the eyes, the eyebrows, the nose, the mouth and the boundary of the face. These fiducial points have been detected manually and are used as landmarks during the establishment of point correspondence in training stages. It is emphasized that in the following experiments landmarks are used only in training stages and not in testing stages.

A. Facial expression recognition

In this series of experiments we demonstrate the performance of the asymmetric model in the recognition of the six prototypic expressions following the 10-fold cross-validation approach.

In each experiment, BU-3DFE subjects are divided randomly in two sets, a training set consisting of 90 subjects and a test set consisting of the rest 10 subjects thus assuring the independence on subject identity. First, the base-mesh M_0 is built by selecting $N = 169$ vertices lying on an average facial surface and then training faces are set into point correspondence as described in Section II-A. The training set is also used to learn the mesh deformation subspace (250 principal modes are kept) and then to train the asymmetric bilinear model. That is, we estimate the 6 matrices $\mathbf{A}^x$ corresponding to the 6 possible expressions and the 90 identity control vectors $\mathbf{b}^p$ corresponding to the subjects of the training set. The number of columns of $\mathbf{A}^x$ and the dimension of $\mathbf{b}^p$ is set to 80 while the number of rows K is equal to the triple of vertices number, $K = 507$. Entries of $\mathbf{A}^x$ and $\mathbf{b}^p$ are initialized randomly and then they are computed by iterating Eq. 27 and 28 until the relative change in their Frobenius norm gets below a threshold (0.01) or a maximum number of iterations (150-180) is reached. Estimated matrices $\mathbf{A}^x$ and vectors $\mathbf{b}^p$ are then used to build the Maximum Likelihood classifier letting the error variance be $\sigma^2 = 10^5$.

TABLE I

EXPRESSION RECOGNITION BASED ON ASYMMETRIC BILINEAR MODEL.

True\Classified	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	**83.6** (5.2)[a]	0.0	0.0	0.0	16.4 (5.2)	0.0
Disgust	0.0	**100.0**	0.0	0.0	0.0	0.0
Fear	0.5 (1.1)	0.0	**97.9** (3.6)	0.0	1.6 (3.2)	0.0
Happiness	0.0	0.8 (0.6)	0.0	**99.2** (0.7)	0.0	0.0
Sadness	34.5 (11.5)	0.0	3.1 (3.3)	0.0	**62.4** (11.4)	0.0
Surprise	0.0	0.0	0.0	0.0	0.0	**100.0**

[a]Values in parentheses are standard deviations.

TABLE II

EXPRESSION RECOGNITION BASED ON PRIMITIVE SURFACE FEATURE DISTRIBUTION (WANG *et al.* [1]).

True\Classified	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	**80.0**	1.7	6.3	0.0	11.3	0.8
Disgust	4.6	**80.4**	4.2	3.8	6.7	0.4
Fear	0.0	2.5	**75.0**	12.5	7.9	2.1
Happiness	0.0	0.8	3.8	**95.0**	0.4	0.0
Sadness	8.3	2.5	2.9	0.0	**80.4**	5.8
Surprise	1.7	0.8	1.2	0.0	5.4	**90.8**

During testing, each test face is set into correspondence following the subspace-guided approach described in Section II-B (we assume that landmarks are not available for test faces). Once the vertex vector $\mathbf{v}$ is computed, the expression of the face is classified to the class with the highest likelihood (see Section IV).

This procedure (training-testing) is repeated 10 times assuring that every subject is included in one test set. Results are averaged and presented as a confusion matrix in Table I. Expressions have been labelled by the subjects who have performed them and this labelling is used as ground truth in our experiments. In order to provide a measure of comparison, we report that the average recognition rate achieved by human experts [6] varies from 94.1% for low intensities up to 98.1% for highest intensities. We also provide Table II which shows the corresponding confusion matrix obtained according to [1]. From Table I it can be seen that highest misclassification occurs between the expressions of *anger* and *sadness*. The decrease in these recognition rates is attributed to their similarity especially in low intensities. We note that in our experiments we used for testing facial models of all intensities for every expression, instead of the highest two as in [1]. The main difference between the *angry* and *sad* expression lies mostly on the configuration of the eyebrows, which cannot be captured effectively using depth (at least with our point correspondence technique) especially in low intensities, where the difference is so subtle even for a human eye. Nevertheless, the total average recognition rate of 90.5% achieved proves the overall superior performance of the proposed algorithm.

B. Face recognition

To evaluate 3D face recognition performance we adopt the following procedure which simulates a realistic application scenario. We split the database into two parts based on subject identity. The first part serves as a bootstrap set and is used for training the elastically deformable model (subspace learning described in Section II-B) and computing the bilinear model coefficients. The rest of the data, the test set, is split into the gallery set that contains a single 3D image per subject (neutral or non-neutral), and the probe set that contains the images to be classified (various facial expressions).

The bootstrap set is comprised of 50 subjects chosen randomly from the database while the gallery and probe set are comprised of the rest 50 subjects. Faces in the bootstrap set are first set into correspondence following the landmark-guided approach of Section II-A. Then, they are used to learn the mesh deformation subspace (250 principal modes are kept again) and to train a symmetric bilinear model. The training of the model starts by training an asymmetric bilinear model such as the one used for facial expression recognition above. This model is used to initialize the training of another initial symmetric model following the SVD approach described in Section V. The dimension of vectors $\mathbf{b}^p$ is set to 45 while the dimension of vectors $\mathbf{a}^x$ is set to 5. The final symmetric model results from the optimization of the estimated parameters $\mathbf{a}^x$, $\mathbf{b}^p$ and $\mathbf{W}_k$ which is achieved by iterating equations Eq. 18, Eq. 19 and Eq. 21 until the relative change in their Frobenius norm gets below a threshold (0.01) or a maximum number of iterations (150-180) is reached.

Then we proceed with gallery image processing. First the deformable model has to be fitted to each gallery image. We assume that we do not have any landmarks and therefore the subspace-guided approach described in Section II-B is used. Then, the bilinear model is fitted to each gallery image to acquire its expression and identity control parameters.

This procedure (deformable model and bilinear model fitting) is also repeated for each probe image during testing. Having obtained its expression and identity control parameters,

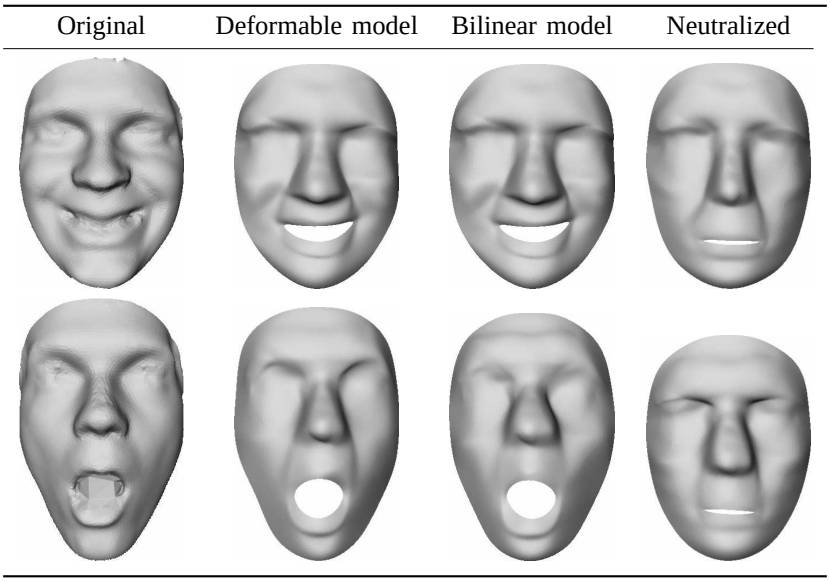

Fig. 7. Expression manipulation. First column shows the original 3D face scans of the same subject displaying a smiling (first row) and a surprising (bottom row) expression. Second column shows the elastically deformable model fitted to the original surfaces, while third column shows reconstructions of the surfaces using bilinear model coefficients. Neutralization of expressions shown in fourth column is achieved by modulating the expression control parameters.

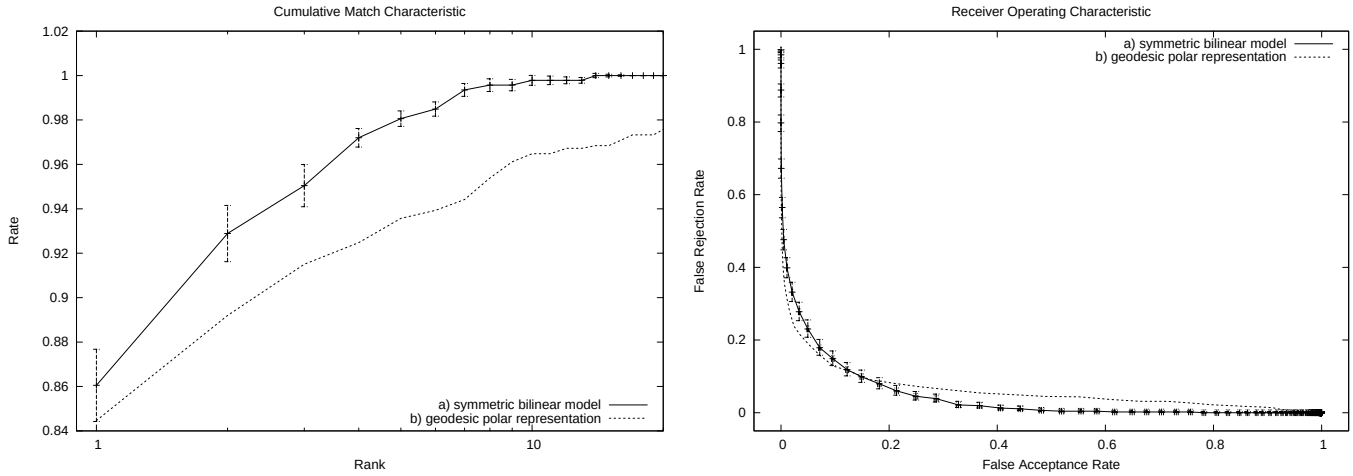

Fig. 8. Cumulative Match Characteristic (CMC) and Receiver Operating Characteristic (ROC) for face recognition based on: a) the symmetric bilinear model, b) the geodesic polar representation presented in [2]. Error bars indicate standard deviation in CMC and 95% confidence interval in ROC.

the probe image is then compared with every image in the gallery to obtain the similarity score. (See Section V).

We have repeated the above experiments on several randomly chosen subdivisions of bootstrap and test sets, under the constraint that all subjects are included at least once in the test set. The recognition results are averaged and presented in Fig. 8, which shows the cumulative match characteristic and the receiver operating characteristic of the proposed system compared with the results obtained by our previous work in [2]. There, we used an expression-invariant face representation, based on geodesic polar coordinates and an isometric model of the facial surface deformation, which showed to be better than Bronstein *et al.*'s [15] canonical images and a PCA-based algorithm. We also note that some images depicting extreme expressions that violated the isometry assumption and had to be excluded from experiments in [2] are now included in these experiments. The increase in the rank-1 recognition rate shows that the proposed algorithm may deal well even with extreme expressions which are one of the main limitations of current 3D face recognition algorithms.

Deformable model and bilinear model training requires a few hours in a typical Pentium V, 3GHz, 1GB RAM workstation running non-speed-optimized code. On the other hand, depth acquisition is performed in less than 2 milliseconds [6], while processing of a novel image takes less than 3 seconds (2 seconds for point correspondence, 1 second for bilinear model fitting), which means that the proposed algorithm may be used in near-real-time systems.

C. Limitations

In the previous sections, we showed that bilinear models may capture effectively the bi-factor nature of the facial surface geometry and thus lead to high facial identity and expression recognition rates. Nevertheless, there are still some

issues that limit recognition performance, especially face recognition, and provide room for further investigation.

The main limitation is the need of a large bootstrap set which should also be annotated with respect to facial expressions. The more different expressions are present in the bootstrap set, the better is the estimation of the interaction matrices $\mathbf{W}_k$ and the better is the fit to novel faces. Training with a few expressions leads to unbalanced generalization ability in favour of identity which in turn leads to better surface approximation but poorer expression control. However, building and annotating in practice a bootstrap set may be difficult considering that a great number of possibly ambiguous expressions have to be classified into a finite number of expression classes.

Another factor that affects performance is the accuracy of point correspondence between faces. In our experiments we observed that poor correspondence affects substantially bilinear model training and eventually recognition performance. The problem is twofold: During training, the bilinear model cannot learn the true identity-expression manifold implying errors in bilinear parameters estimation. During testing, expression manipulation is actually applied on a slightly (or quite) different face. This error is further amplified by inaccurate bilinear parameters leading to a distorted facial surface.

VII. Summary

In this paper we proposed a technique for joint 3D face and facial expression recognition. We first presented a novel model-based approach for establishing point correspondence among faces which involves the solution of a simple linear system. We also proposed using directed distances both from the mesh to the cloud of facial points and inversely which leads to a smoother force field and thus more plausible anatomical correspondence. Another advantage of this approach is that correspondence may be performed fully automatically after training the system with a number of facial surfaces annotated with anatomical salient points. We also provided a solution for the problem of open mouth in this case whose configuration might cause problems during the establishment of correspondence. Then, we proposed bilinear models for joint face and expression recognition and we provided the general solution of the error minimization during the training of the symmetric bilinear model. Our algorithm was finally evaluated on the BU-3DFE database and proved its superiority over expression-invariant face representations for face recognition [2] and primitive surface features for expression recognition [1].

VIII. Acknowledgment

This work was supported by a) the European Commission under the FP6 IST Project: "PASION - Psychologically Augmented Social Interaction over Networks" (contract No FP6-027654) and FP6 IST Network of Excellence: "3DTV-Integrated Three-Dimensional Television - Capture, Transmission, and Display" (contract FP6-511568), b) the Greek Ministry of Development-General Secretariat of Research and Technology under Project: PENED 2003 Ontomedia (03EΔ475).

Appendix

Let $\mathbf{v}$ be the $3N$-dimensional vector of the base-mesh vertices. Let also $\mathbf{t} = [\mathbf{t}_1^T \dots \mathbf{t}_N^T]^T$ be a $3N$-dimensional vector containing landmark coordinates $\mathbf{y}_k$ and ϕ_i, $i = 1 \dots N$, 3-dimensional vectors so that,

$$\mathbf{t}_i = \begin{cases} \mathbf{y}_k & \text{if vertex } i \text{ of } M_0 \text{ corresponds to a} \\ & \text{landmark } k = c^{-1}(i) \\ \mathbf{0}_3{}^4 & \text{otherwise} \end{cases} \tag{39}$$

$$\phi_i = \begin{cases} \mathbf{1}_3 & \text{if vertex } i \text{ is a landmark vertex} \\ \mathbf{0}_3 & \text{otherwise} \end{cases} \tag{40}$$

Then Eq. 3 yields

$$\lambda_1 E_c = (\mathbf{v} - \mathbf{t})^T \, \mathbf{\Phi_t}^T \mathbf{\Phi_t} \, (\mathbf{v} - \mathbf{t}) \tag{41}$$

where $\mathbf{\Phi_t}^T \mathbf{\Phi_t} = \lambda_1 \mathbf{\Phi} = \lambda_1 \text{diag}\{[\phi_1^T \dots \phi_N^T]\}$ using incomplete Cholesky decomposition.

By defining the $3S$ and $3K$ vectors $\mathbf{p_{sc}}$ and $\mathbf{p}$ respectively, and the $3S \times 3N$ and $3K \times 3N$ block matrices $\mathbf{H}$ and $\mathbf{H_{cs}}$ made by 3×3 blocks

$$\mathbf{p_{sc}} = \sqrt{\lambda_2} \left[\mathbf{p}_{sc(1)}^T \, \mathbf{p}_{sc(2)}^T \, \cdots \, \mathbf{p}_{sc(S)} \right]^T \tag{42}$$

$$\mathbf{p} = \sqrt{\lambda_3} \left[\mathbf{p}_1^T \, \mathbf{p}_2^T \, \cdots \, \mathbf{p}_K \right]^T \tag{43}$$

$$[\mathbf{H}]_{ij} = \sqrt{\lambda_2} h_{ij} \mathbf{I}_3 \tag{44}$$

$$[\mathbf{H_{cs}}]_{ij} = \sqrt{\lambda_3} h_{cs(i)j} \mathbf{I}_3 \tag{45}$$

Eq. 4 and 5 yield

$$\lambda_2 E_{sc} = (\mathbf{H}\mathbf{v} - \mathbf{p_{sc}})^T (\mathbf{H}\mathbf{v} - \mathbf{p_{sc}}) \tag{46}$$

$$\lambda_3 E_{cs} = (\mathbf{H_{cs}}\mathbf{v} - \mathbf{p})^T (\mathbf{H_{cs}}\mathbf{v} - \mathbf{p}) \tag{47}$$

The elastic energy E_e in Eq. 6 can be written in matrix notation using the $3 \times 3N$ block matric $\mathbf{\Psi}_i = [\, \mathbf{0}_3 \, \cdots \, \underbrace{\mathbf{I}_3}_{i-\text{th block}} \, \cdots \, \mathbf{0}_3 \,]$ as follows

$$\begin{aligned} \lambda_4 E_e &= \sum_{i=1}^{N} \frac{\lambda_4}{N_i} \sum_{j \in \mathcal{N}_i} \left(\mathbf{v}_i - \mathbf{v}_j - \mathbf{v}_i^0 + \mathbf{v}_j^0 \right)^2 \\ &= \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{\lambda_4}{N_i} \left((\underbrace{\mathbf{\Psi}_i - \mathbf{\Psi}_j}_{\mathbf{\Psi}_{ij}})(\mathbf{v} - \mathbf{v^0}) \right)^2 \\ &= \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{\lambda_4}{N_i} \left(\mathbf{\Psi}_{ij}(\mathbf{v} - \mathbf{v^0}) \right)^T \left(\mathbf{\Psi}_{ij}(\mathbf{v} - \mathbf{v^0}) \right) \\ &= (\mathbf{v} - \mathbf{v^0})^T \left(\underbrace{\sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{\lambda_4}{N_i} \mathbf{\Psi}_{ij}^T \mathbf{\Psi}_{ij}}_{\mathbf{\Psi}} \right) (\mathbf{v} - \mathbf{v^0}) \\ &= (\mathbf{v} - \mathbf{v^0})^T \mathbf{\Psi} (\mathbf{v} - \mathbf{v^0}) \\ &= (\mathbf{v} - \mathbf{v^0})^T \mathbf{\Psi_t}^T \mathbf{\Psi_t} (\mathbf{v} - \mathbf{v^0}) \end{aligned} \tag{48}$$

[4] $\mathbf{0}_3$ is the vector $[0\ 0\ 0]^T$, $\mathbf{1}_3$ is the vector $[1\ 1\ 1]^T$ and $\mathbf{I}_3$ is the 3×3 identity matrix.

where $\mathbf{\Psi_t}$ is a truncated triangular matrix resulting from the incomplete Cholesky decomposition of $\mathbf{\Psi}$.

Replacing the above equations to Eq. 7 it can be easily shown that E_{def} is minimized by the solution of the overdetermined linear system Eq. 49 which is solved using singular value decomposition.

$$\begin{bmatrix} \mathbf{\Phi_t} \\ \mathbf{H} \\ \mathbf{H_{cs}} \\ \mathbf{\Psi_t} \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{\Phi_t t} \\ \mathbf{p_{sc}} \\ \mathbf{p} \\ \mathbf{\Psi_t v^0} \end{bmatrix} \tag{49}$$

REFERENCES

[1] J. Wang, L. Yin, X. Wei, and Y. Sun, "3D facial expression recognition based on primitive surface feature distribution," in *Proc. Conf. Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1399–1406.

[2] I. Mpiperis, S. Malassiotis, and M. G. Strintzis, "3D face recognition with the geodesic polar representation," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 3, pp. 537–547, September 2007.

[3] K. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition," *Comp. Vision and Image Understanding*, vol. 101, pp. 1–15, 2006.

[4] K. Chang, K. Bowyer, and P. Flynn, "An evaluation of multi-modal 2D + 3D face biometrics," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 27, no. 4, pp. 619–624, April 2005.

[5] P. Ekman and W. Friesen, *Facial Action Coding System (FACS): Manual*. Palo Alto: CA: Consulting Psychologists Press, 1978.

[6] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, April 2007, pp. 211–216.

[7] X. Lu and A. Jain, "Deformation analysis for 3D face matching," in *7th IEEE Workshop on Applications of Computer Vision*, Colorado, 2005.

[8] C. Samir, A. Srivastava, and M. Daoudi, "Three-dimensional face recognition using shapes of facial curves," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 28, pp. 1858–1863, 2006.

[9] C.-S. Chua, F. Han, and Y.-K. Ho, "3D human face recognition using point signature," in *Proc. 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000, pp. 233–238.

[10] M. Pantic and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 22, no. 12, pp. 1424–1445, December 2000.

[11] G. Passalis, I. Kakadiaris, T. Theoharis, G. Toderici, and N. Murtuza, "Evaluation of 3D face recognition in the presence of facial expressions: An annotated deformable model approach," in *Proc. IEEE Workshop Face Recognition Grand Challenge Experiments*, June 2005.

[12] I. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 29, no. 4, pp. 640–649, April 2007.

[13] K. Chang, K. Bowyer, and P. Flynn, "Multiple nose region matching for 3D face recognition under varying facial expression," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 28, no. 10, pp. 1695–1700, October 2006.

[14] X. Li and H. Zhang, "Adapting geometric attributes for expression-invariant 3D face recognition," in *IEEE Int. Conf. on Shape Modeling and Applications*, 2007, pp. 21–32.

[15] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Expression-invariant representations of faces," *IEEE Trans. Image Processing*, vol. 16, no. 1, January 2007.

[16] P. Aleksic and A. Katsaggelos, "Automatic facial expression recognition using facial animation parameters and multistream HMMs," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 1, pp. 3–11, March 2006.

[17] S. Ioannou, G. Caridakis, K. Karpouzis, and S. Kollias, "Robust feature detection for facial expression recognition," *EURASIP J Image Video Processing. Special issue on Facial Image Processing*, 2006.

[18] C. S. Lee and A. Elgammal, "Nonlinear shape and appearance models for facial expression analysis and synthesis," in *18th Int. Conf. on Pattern Recognition (ICPR'06)*, 2006, pp. 497–502.

[19] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2003, pp. 93–99.

[20] H. Wang and N. Ahuja, "Facial expression decomposition," in *Proc. Ninth IEEE Int. Conf. on Computer Vision*, vol. 2, October 2003, pp. 958–965.

[21] J. Tenenbaum and W. Freeman, "Separating style and content," in *Advances in Neural Information Processing Systems*, vol. 9, 1997.

[22] ——, "Separating style and content with bilinear models," *Neural Computation*, vol. 12, pp. 1247–1283, 2000.

[23] C. Mandal, H. Qin, and B. C. Vemuri, "Novel FEM-based dynamic framework for subdivision surfaces," *Computer-Aided Design*, vol. 32, no. 8, pp. 479–497, 2000.

[24] J. L. Bentley, "K-d trees for semidynamic point sets," in *Proc. Sixth Annual Symposium on Computational Geometry*, 1990, pp. 187–197.

[25] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 25, no. 9, pp. 1063–1074, September 2003.

[26] D. Metaxas and I. Kakadiaris, "Elastically adaptive deformable models," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 24, no. 10, pp. 1310–1321, October 2002.

[27] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans," in *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*. New York, NY, USA: ACM Press, 2003, pp. 587–594.

[28] G. Edwards, A. Lanitis, C. Taylor, and T. Cootes, "Statistical models of face images: Improving specificity," in *British Machine Vision Conference*, vol. 2, 1996, pp. 765–774.

[29] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[30] X. Lu and A. K. Jain, "Multimodal facial feature extraction for automatic 3D face recognition," Department of Computer Science, Michigan State University, East Lansing, Michigan, Tech. Rep. MSU-CSE-05-22, August 2005.

[31] C. V. Loan, "The ubiquitous kronecker product," *Journal of Computational and Applied Mathematics*, vol. 123, p. 85100, 2000.

[32] D. Lin, Y. Xu, X. Tang, and S. Yan, "Tensor-based factor decomposition for relighting," in *Proc. IEEE Int. Conf. on Image Processing (ICIP2005)*, vol. 2, September 2005, pp. 386–389.

[33] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Proc. IEEE Computer Vision and Pattern Recognition*, June 2005, pp. 947–954.