Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Fully Automatic Expression-Invariant Face Correspondence

Augusto Salazar · Stefanie Wuhrer · Chang Shu · Flavio Prieto

Received: date / Accepted: date

Abstract We consider the problem of computing accurate point-to-point correspondences among a set of human face scans with varying expressions. Our fully automatic approach does not require any manually placed markers on the scan. Instead, the approach learns the locations of a set of landmarks present in a database and uses this knowledge to automatically predict the locations of these landmarks on a newly available scan. The predicted landmarks are then used to compute point-to-point correspondences between a template model and the newly available scan. To accurately fit the expression of the template to the expression of the scan, we use as template a blendshape model. Our algorithm was tested on a database of human faces of different ethnic groups with strongly varying expressions. Experimental results show that the obtained point-to-point correspondence is both highly accurate and consistent for most of the tested 3D face models.

Augusto Salazar
Perception and Intelligent Control research group, National University of Colombia
Km 9 via al aeropuerto la Nubia, Manizales, Caldas, Colombia
Tel.: +57-68-789400
E-mail: aesalazarj@unal.edu.co

Stefanie Wuhrer
Cluster of Excellence, Multimodal Computing and Interaction, Saarland University

Chang Shu
National Research Council of Canada

Flavio Prieto
GAUNAL research group, National University of Colombia

Keywords Non-rigid 3D registration · Automatic landmark prediction · Facial Expression-invariant · Blendshape model · Energy minimization

1 Introduction

We consider the problem of computing point-to-point correspondences among a set of human face scans with varying expressions in a fully automatic way. This problem arises from building a statistical model that encodes face shape and expression simultaneously using a database of human face scans. In order to build a statistical model, we rely on the correct computation of dense point-to-point correspondences among the subjects of a database. That is, the raw scans have to be parameterized in such a way that likewise anatomical parts correspond across the models [1]. Facial expression affects the geometry of the human face and therefore is important for facial shape analysis. A statistical model of face shapes and expressions can be used in applications such as face recognition, expression recognition, or reconstructing accurate 3D models of faces from input images [2–6].

Computing accurate point-to-point correspondences for a set of face shapes in varying expressions is a challenging task because the face shape varies across the database and each subject has its own way to perform facial expressions. The problem is further complicated by incomplete and noisy data in the scans.

While many approaches have been proposed to compute point-to-point correspondences [7], only few of them have been applied to statistical model building and shape analysis of human face shapes. Blanz and Vetter [2] built a statistical model called

morphable model for a set of 3D face scans with varying expressions. The correspondence algorithm is based on using optical flow on the texture information of the faces. This assumes that the faces are approximately spatially aligned. Xi and Shu [8] built a statistical model based on principal component analysis for a set of 3D face scans with neutral expressions. The correspondence algorithm is based on fitting a template model to the scans using a non-rigid iterative closest point algorithm. To start this algorithm, the faces need to be approximately aligned using a set of manually placed marker positions. Both of these registration approaches fail for misaligned models.

In this work, we develop a novel technique to compute correspondences between a set of facial scans with varying expressions that does not require the scans to be spatially aligned. Our correspondence computation procedure uses a template model P as prior knowledge on the geometry of the face shapes. Unlike Xi and Shu [8], we aim to find correspondences for faces with varying expressions. Hence, it is not enough to have a template model that captures the face shape of a generic model, but we also need to capture the expressions of a generic model. To achieve this, we model P as a blendshape model as in Li et al. [9]. In a blendshape model, expressions are modeled as a linear combination of a set of basic expressions. Hence, blendshape models are both simple and effective to model facial expressions.

Our approach proceeds as follows. We first use a database of human face scans with manually placed landmark positions to learn local properties and spatial relationships between the landmarks using a Markov network. Given an input scan F without manually placed landmarks, we first predict the landmark positions on F by carrying out statistical inference over the trained Markov network. Sections 3.1 and 3.2 discuss this step. In order to perform statistical inference, we need to restrict the search region for each landmark. This is detailed in Sections 3.3 to 3.6. The predicted landmarks are used to align P to F. In order to fit the expression of P to the expression of F, the template is aligned to the scan as outlined in Section 4.1 and the weights of the generic blendshape model are optimized as discussed in Section 4.2. Finally, the shape of P is changed to fit the shape of F as outlined in Section 4.3. Fig. 1 shows an overview of the method.

2 Related Work

This section reviews literature in face shape analysis related to finding landmarks on face models, computing correspondences between three-dimensional shapes, and using blendshape models for facial animation.

2.1 Finding Landmarks on Face Models

Traditionally, facial features are detected in 2D images. In this setting, facial feature detection can be achieved in an unsupervised (see for instance [10, 11]), semi-supervised (see for instance [12]) or supervised (see for instance [13]) manner. Unsupervised methods do not use prior information about the geometry of target object. However, these methods only estimate a global affine transformation between the source and target object. On the other hand, semi-supervised and supervised methods estimate a shape deformation described by a set of landmarks, which provide more accurate and consistent results. To incorporate prior knowledge about landmark locations, it often suffices to annotate only a few examples manually [12].

Recent developments on 3D data acquisition have allowed to overcome the problems attached to the 3D technologies. However, only a few approaches consider 3D landmark detection, while accounting for expression and pose variations [14]. It is well-studied that facial landmarks play an important role in applications, such as face or expression recognition [15].

Ben Azouz et al. [16] propose a method to find correspondences by automatically predicting marker positions on 3D models of a human body. The method encodes the statistics of a surface descriptor and geometric properties at the locations of manually placed landmarks in a Markov network. This method works only for models with slight variation of posture. Mehryar et al. [14] introduce an algorithm to automatically detect eyes, nose, and mouth on 3D faces. The algorithm correctly detects the landmarks in the presence of pose, facial expression and occlusion variations. This method is useful as initial alignment but not for an accurate registration. Berreti et al. [17] combine principal curvatures analysis, edge detector and SIFT descriptors to find 9 landmarks on the eyes nose and mouth regions in range images. The landmarks are properly detected in the presence of facial expressions but the method relies in anthropometric facial proportions to define the search regions and

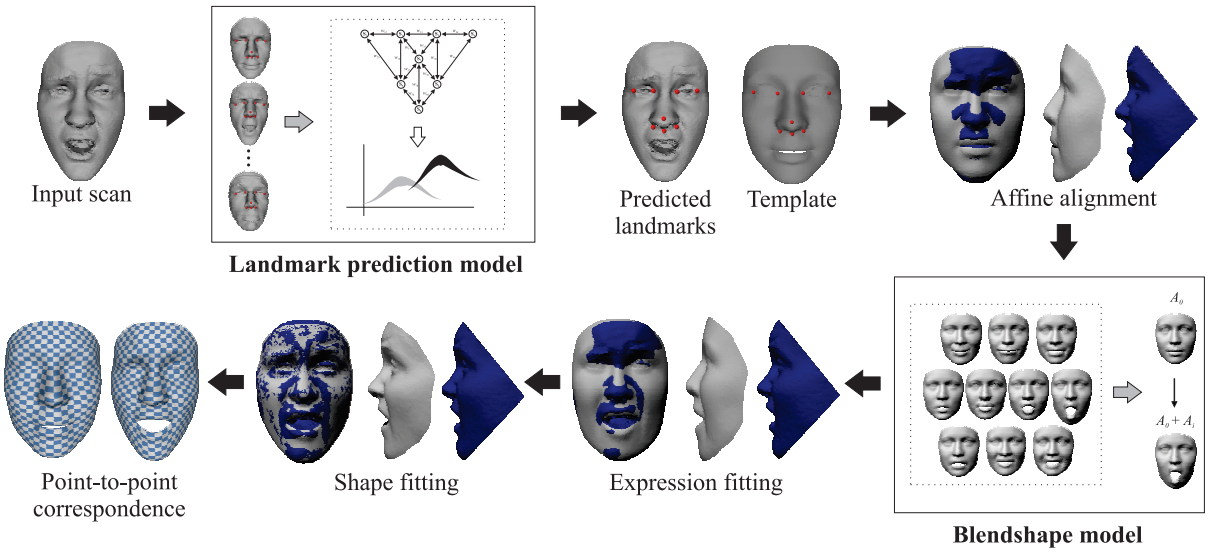


Fig. 1 Overview of the fully automatic expression-invariant face correspondence approach.

assumes that the face is upright oriented. Creusot et al. [18] present a method to localize a set of 13 facial landmark points under large pose variation or when occlusion is present. Their method learns the properties of a set of descriptors computed at the landmark locations and encodes both local information and spatial relationships into a graph. The method works well for neutral pose. However, in the presence of expression variation, the accuracy decreases considerably. Segundo et al. [19] develop a method for face segmentation and landmark detection in range images. The landmark detection method combines surface curvature information and depth relief curve analysis to find 5 landmarks located on the nose and eye regions. The landmarks are properly detected in the presence of facial expressions and hair occlusions, but the method relies on a specific acquisition setup. Perakis et al. [20,21] present a method to detect landmarks under large pose variations using a Active Landmark Model (ALM), which is a statistical shape model learned from 8 manually annotated landmarks. Using a combination of the Shape Index descriptor and Spin Images, the search space for the fitting of the ALM is defined. The final set of landmarks is defined by selecting the set of candidates that satisfies the geometric restrictions encoded in the ALM. The experiments show that the method works in the presence of facial expressions and pose variation up to 80 degrees around the y-axis. Nair and Cavallaro [22] use a point distribution model to estimate the location of 49 landmarks on the eyebrow, eye and nose regions. The method works well in the presence of expressions and noisy data. However the error in the localization of landmarks is quite high (a comparison of the results is provided in Section 5.2). Lu and Jain [23] present a multimodal approach for facial feature extraction. The nose tip is located using only the 3D information, and the eyes and mouth corners are extracted using 2D and 3D data. As their focus is handling changes in head pose and lighting conditions, variations due to facial expressions are not considered in their experiments. This multimodal approach is used by Lu et al. [24] as part of a system for face recognition in the presence of pose and expression variation (only smiling expression variations are included in the test data). The authors claim that the expression changes decrease the accuracy of the system. However, quantitative results of the landmark detection are not provided. In addition, the requirement of the texture data is a limitation of the multimodal approaches because sometimes such information is not available.

As our aim is to obtain accurate point-to-point correspondences, we derived a landmark prediction method based on the approach of Ben Azouz et al. [16]. The surface descriptor we used is able to catch the local geometry properly [26] and, by combining it with a canonical representation [27], our approach is able to detect landmarks in the presence of facial expressions. We select a machine learning-based approach to avoid classic assumptions such as: the nose tip is the closest point to the camera [28], the inner-corners of the eyes and the tip of the nose are the most salient points [19], the 3D face scan is in a frontal upright canoni-

cal pose [17], among others. The advantage is that learning-based approaches can easily be extended to other contexts.

2.2 Correspondence Computation

Several methods have been proposed to solve the problem of establishing a meaningful correspondence between shapes. Here, we focus on computing correspondences between human face shapes. Methods that do not assume templates usually have the problem that some points are not registered accurately. To remedy this, we assume a template model. In the following, we only review approaches that use template models (for details about methods for correspondence computation see the survey of van Kaick et al. [7]).

Passalis et al. [29] proposed a 3D face recognition method that uses facial symmetry to handle pose variation and missing data. A template is fitted to the shape of the input model as follows: an Annotated Face Model [30] is iteratively deformed towards the input using automatically predicted landmarks and an algorithm based on Simulated Annealing. When dealing with facial expressions, the performance of the recognition system decreases. This is due to an incorrect registration of the mouth region. Mpiperis et al. [31] propose a method that supports both 3D face recognition and expression recognition. A template model is fitted to the shape of the input model using an elastic deformation model. Both works do not show direct evaluations of the fully-automatic registration methods as this is not the main part of these works.

Guo et al. [25] propose a multimodal approach to automatically compute correspondences between 3D face models. The approach predicts 17 landmarks using a PCA-based method and uses these features to deform a template to the input model using a thin-plate spline. Although the registration results are shown to be accurate, the method cannot compute correspondences in the presence of expression variation.

Huang et al. [32] recently presented an approach to register 3D facial models in the presence of facial expressions. They first detect a set of landmarks using texture information with the help of an active appearance model. These points are used in an iterative fitting procedure, which combines displacement mapping, point-to-surface mapping, and a regional blending algorithm to fit a template to the 3D surface. The fitting accuracy of this method is evaluated on manually selected landmarks, and a high fitting accuracy is presented, thereby demonstrating that the combined use of geometry and texture leads to good results. In contrast, our method is purely geometry-based, and could therefore in principle also be applied to 3D data of faces without reliable texture information.

Statistical learning-based approaches have been effectively used to model facial variations oriented to both the synthesis and recognition of faces. Blanz and Vetter [2] developed a 3D morphable model (3DMM) for the synthesis of 3D faces from photographs. As the registration is specific to the scanning setup, rigid alignment of the scans is assumed. Lu and Jain [33] present an approach to perform face recognition using 3D face scans. The approach builds a 3DMM for each subject in the database. When a test image becomes available, the approach matches the scan to a specific individual using the learned 3DMM. Unlike our method, their training data is parameterized using manually placed landmarks and the test scans are parameterized using individual-specific deformation models. Basso et al. [34] extend the method of Blanz and Vetter [2] to register 3D scans of faces with arbitrary identity and expression. The rigid alignment of the scans is also assumed for registration. To avoid the use of texture information, Amberg et al. [35] present a method to fit a 3DMM to 3D face scans using only shape information. They demonstrate the performance of the method in the presence of expression variation, occlusion and missing data, but do not conduct extensive evaluations of the registration.

Registration methods based on iteratively deforming a template to the data are an alternative to statistical learning-based approaches. Allen et al. [36] present an approach to parameterize a set of 3D scans of human body shapes in similar posture. To fit the template to each scan, the method proceeds by using a non-rigid iterative closest point (ICP) framework coupled with a set of manually placed marker positions. Xi and Shu [8] extend the method of Allen et al. [36] to deform a template model to a head scan. The shape fitting is carried out as in Allen et al. [36] but uses radial basis functions to speed up the deformation process. Unlike our method, this only allows for neutral expressions and uses manually placed markers to align the template to a head scan. Wuhrer et al. [37] propose a method to deform a template model to a human body scan in arbitrary posture. The method works in two stages: posture and shape fitting. Posture fitting relies on the location of different land-

marks, which are predicted in a fully automatic way using a statistical model of landmark positions learned from a population. Our method can be viewed as an extension of this approach, but instead of fitting the posture, we fit the expression using blendshapes (see Section 2.3).

Methods that compute a correspondence between two surfaces by embedding the intrinsic geometry of one surface into the other one by using Generalized Multi-Dimensional Scaling (GMDS) [38] are another alternative to deal with variations due to facial expressions [39]. The performance of these methods has been demonstrated for face recognition. As GMDS methods do not take care that close-by points on one surface map to close-by points on the other, the results are often spatially inconsistent. This prevents such methods from being used for shape analysis.

2.3 Use of Blendshape Models

Modeling expressions using blendshape models is an alternative to approaches based on statistical models where a comprehensive database annotation process has to be carried out to extract variational information. In a blendshape model, movements of the different facial regions are assumed to be independent. Any expression is then modeled as a linear combination of the differences between a set of basic expressions, called *blendshapes*, and a neutral expression. That is, to produce an expression, the displacements causing the movement are linearly combined. Using a representative set of blendshapes, this simple model is effective to model facial expressions.

Li et al. [9] propose a method to transfer the expression of a subject to an animated character. Their framework allows to create optimal blendshapes from a set of example poses of a digital face model automatically. Weise et al. [40] present a framework for real-time 3D facial animation. The method tracks the rigid and non-rigid motion of the user's face accurately. They incorporate the expression transfer approach of Li et al. [9] in order to find much of the variation from the example expressions. The registration stage requires offline training where a generic template is fitted to the face of a specific subject. To obtain the results, manual marking of features has to be carried out.

Because of the advantages of modeling expression using linear blendshapes, we use it to aid the shape matching. We only optimize a blending weight per expression. This reduces the dimensionality of the optimization space drastically. Since our database of blendshapes is small, the expression fitting stage of our algorithm is efficient and helps to improve the results significantly.

3 Landmark Prediction

This section outlines how to predict a set of landmark positions on a face scan. To establish the correspondences across the whole database, we fit a template to each model. The fitting process begins with the extraction of the locations of eight landmarks shown as red spheres in Fig. 2. The locations of the landmarks were selected based on the fact that in the presence of facial expressions, the corners of the eyes, and the base and tip of the nose do not move drastically. Each landmark is located automatically on the face surface by means of a Markov network following the procedure proposed by Ben Azouz et al. [16]. The network learns the statistics of a property of the surface around each landmark and the structure of the connections shown in Fig. 2.

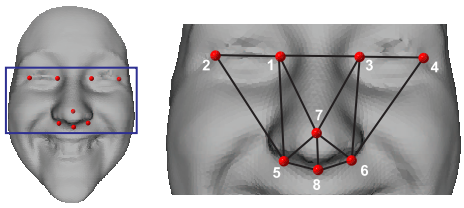


Fig. 2 Face model with landmarks. Locations and landmark graph structure.

3.1 Learning

Two important aspects have to be defined for the training of the Markov network. First, each landmark l_i $(i = 1, 2, \ldots, L)$, represented by a network node, is described using a node potential ϕ_i. We use a surface descriptor based on the *Finger Print* (*FP*) [26]. The descriptor uses a measure related to the area of a geodesic circle centered at the point to be characterized. The descriptor at a point p_k $(k = 1, 2, \ldots, N$, N is the number of vertices in the model) is obtained by computing the distortion of the geodesic disks with respect to Euclidean disks of the same radius. More specifically, the distortion of the area $A(c)$ of the geodesic disk c of radius r centered at p_k is computed as $d(r) = A(c)/(\pi r^2)$. We use as descriptor of p_k a

vector of distortions $d(r_i)$ obtained by varying the radius r_i of the geodesic disk (see Fig. 3). The reason we use FP as node potential is because it is isometry-invariant. Hence, in scenarios where the surface undergoes changes that preserve isometry, FP has been effective to encode the surface information of an object. FP is used to predict landmarks on human models in varying poses [41].

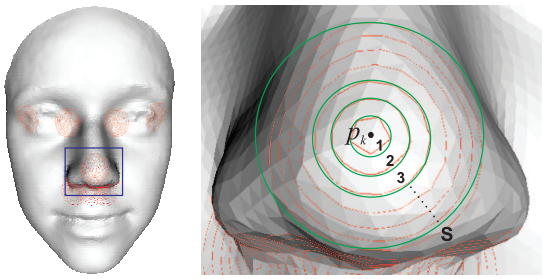


Fig. 3 Circles used to compute the Finger Print descriptor. Red and green circles correspond to the Geodesic and Euclidean circles, respectively.

Second, a link between landmarks l_i and l_j, represented by a network edge, is described using an edge potential $\psi_{i,j}$. Although we selected the locations of the landmarks based on the observations that nose and eye regions do not change much in the presence of expressions, some distortions along the edges of the Markov network may occur. To minimize the effects of the face movements, we compute the canonical form [27] of each model and define the edge potential as the relative position of landmark l_i with respect to landmark l_j in the canonical form space. We compute the canonical form as the embedding of the intrinsic geometry of the face surface to $\mathbb{R}^3$. To compute this embedding, we perform least-squares multi-dimensional scaling [42] with geodesic distances between vertices as dissimilarities. That is, we find the embedding coordinates x_i in $\mathbb{R}^3$ corresponding to vertices p_i on the scan F that minimize the energy

$$E_{MDS} = \sum_{i,j} (|x_i, x_j| - dist_F(p_i, p_j))^2, \qquad (1)$$

where $|\cdot,\cdot|$ corresponds to the Euclidean distance and $dist_F(p_i, p_j)$ denotes the geodesic distance between p_i and p_j on F. The geodesic distances are computed using fast marching [27]. We choose these standard techniques as they are efficient. The choice of the potentials $\psi_{i,j}$ ensures that the model is isometry-invariant.

The Markov network training process learns the distributions of both node and edge potentials for each individual node and egde of the network, respectively. We assume Gaussian distributions for both the node and edge descriptors in this paper, and we learn the distributions using maximum likelihood estimation. We choose this commonly used distribution to derive an efficient algorithm that is easy to implement. While this distribution may not be satisfied in practice, we found experimentally that using this simplified assumption yields satisfactory results.

3.2 Prediction with Belief Propagation

The estimation of the location of landmarks on a test model is carried out by using probabilistic inference over the Markov network. That is, we aim to find landmark locations l_i, such that the joint probability

$$p(l_1, \ldots, l_L) = \frac{1}{Z} \prod_i \phi_i(l_i) \prod_{i,j} \psi_{i,j}(l_i, l_j) \qquad (2)$$

is maximized, where Z is a normalizing factor. In practice, we find an approximate solution using the loopy belief propagation algorithm [43]. This algorithm requires a set of possible labels for each node. In our case, this means we need to provide a number of candidate locations for each landmark.

Wuhrer et al. [37] use canonical forms to learn the average locations of the landmarks, but because of the flipping-invariant property of the canonical forms, it is necessary to compute eight different alignments and select the one that leads to the minimum distance between the scan and the deformed template. In this work, we design a method to restrict the search space based on a rough template alignment. In this way, only one fitting process has to be computed, reducing the computing cost by a factor of eight.

3.3 Restricting the search region

There are two reasons to reduce the search space for the landmarks: to increase the efficiency of the landmark prediction and to eliminate the ambiguity caused by the facial symmetry. We treat the problem of restricting the search region for the landmarks as a 3D face pose estimation problem. In our case, the estimated pose does not have to be so accurate since the Markov network refines the position of the landmarks, but it has to be accurate enough to identify the left and right sides of the face. The proposed face pose estimation method

finds four landmarks located on the nose region and extracts the information of the face symmetry planes by using a template of the landmark graph. Once the nose landmarks are labeled, the final position of the entire set of landmarks is obtained by transforming the template to the coordinate system of the test model. Fig. 7 shows the main steps of the proposed search space restriction method.

3.4 Classification of Vertices

Before explaining the rough template alignment procedure, we introduce a method to classify a vertex of a 3D model into a specific class. In our case, the classes correspond to the nodes of the Markov network and the 3D model corresponds to a 3D face model. The decision rules are derived from a clustering procedure over the Principal Components Analysis (PCA) projections of a surface feature and a pre-selection method based on the surface primitives.

As the value of the FP descriptor at each landmark l_i was computed during the Markov network training process, we can model the distributions of the surface descriptors and use them to classify a vertex v_k on the face surface into a class i (each landmark corresponds to a class). PCA is a useful tool to compress a high-dimensional space into a linear low-dimensional space. When the space corresponds to a multidimensional feature space, sometimes, depending on the distinctiveness of the features, it is possible that elements of the same class form clusters in the PCA space. In our case, the FP descriptor can be viewed as S-dimensional vector and PCA is used to reduce the dimensionality to D. In this work we choose $D = 3$. Fig. 4 shows the results of applying PCA to the data from the subjects in neutral and performing six expressions (for information about the database, see Section 5.1).

Although samples of the same class tend to form groups in the PCA space, some groups overlap due to symmetric landmarks. In order to improve the separation between classes, we define a new cluster, denoted as *M-cluster*, by removing the samples which are farther than M ($M \in \mathbb{R}^+$) times the standard deviation from the cluster medoid. Medoids are representative objects of a cluster whose average dissimilarity to all the objects in the cluster is minimal [44]. For instance, Fig. 4 shows the *M-clusters* formed by setting $M = 1.5$. With this value, the clusters corresponding to the landmarks nose tip and subnasal (points 7 and 8 in Fig. 2) do not overlap any of the clusters. We will show in Sections 3.5 and 3.6 that with a good separation between these two classes, a proper landmarks prediction can be obtained.

We derive a rule E_i for a class i based on a clustering procedure. The rule E_i is defined as the minimum volume enclosing ellipsoid of a *M-cluster*$_i$ (see Fig. 4). E_i is obtained from the representation of the ellipsoid in the center form as $(p_k - C_i)^T A(p_k - C_i) \leq 1$, where C_i corresponds to the center of the ellipsoid corresponding to class i and A is the 3×3 matrix of the ellipse equation. When a new point p_k becomes available, each E_i is evaluated in order to see if the point satisfies the equation. As some *M-clusters* are overlapping, it is possible that more than one label be assigned to the same p_k. Similarly, it is possible that p_k is not assigned to any class because the point lies in a region that is not of interest. Fig. 5 shows an example of the vertex classification results obtained using the proposed method.

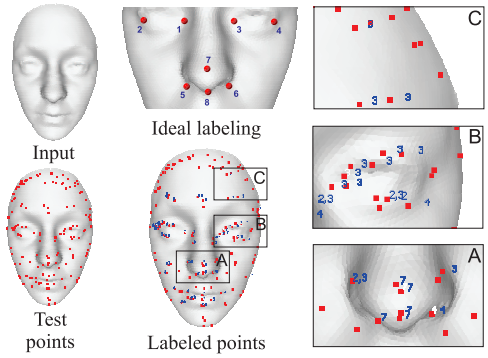


Fig. 5 Example of vertex labeling result. (A) Notice how the points on the nose tip region are correctly labeled. (B) Some vertices are assigned to two classes. This situation is because of the left-right symmetry of the features. (C) Points located far from the region of interest are discarded.

It is not efficient to compute the descriptor value and its projection to PCA space for all the vertices of the mesh. To reduce the search space, we compute samples on the surface using a curvature-based descriptor. More precisely, we use as samples all surface *umbilics* [45], which are the points on the surface where the principal curvatures are identical (that is, $k_1 = k_2$). We choose this sampling approach because it can be observed experimentally that most landmark positions are located close to a umbilic, as shown in Fig. 6.

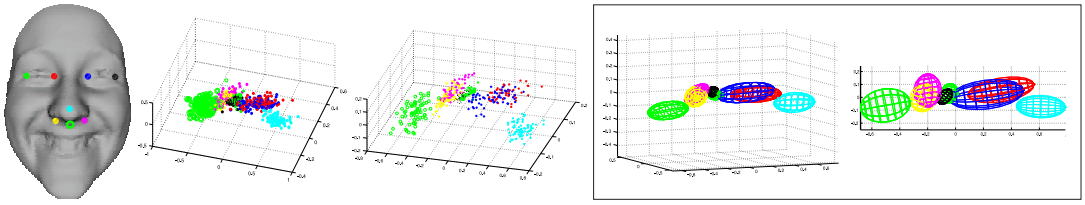


Fig. 4 PCA-based clustering. Left to right: Landmarks on a face model. Initial clusters formed with all the samples. Final cluster after removing the samples beyond a 1.5 standard deviations from the cluster medoid. Minimum volume enclosing ellipsoids (3D and upper views).

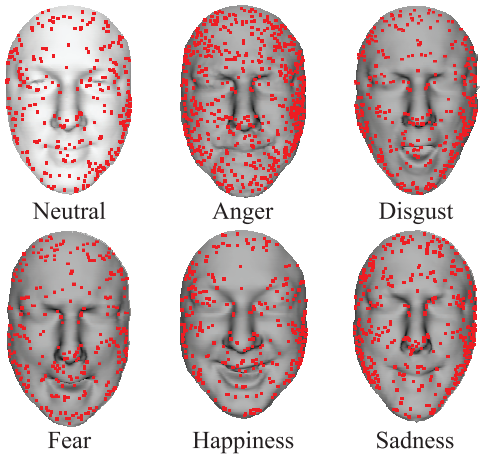


Fig. 6 *Umbilics* of different 3D facial models of the same subject performing different expressions. Notice how the *umbilics* are distributed all over the surface, and in most of the cases umbilics are present at the locations of salient facial features.

3.5 Refining the Nose Landmarks

In this section we describe the procedure to select candidates for four points on the nose area, which are used as initial guess of the landmarks: right subalare, left subalare, nose tip, and subnasal, which are labeled as 5, 6, 7 and 8, respectively (see Fig. 2). Following the classification procedure described in Section 3.4, for each umbilic of the input scan F, the FP descriptor is computed, projected into PCA space, and labeled (in the following we refer to this procedure as *FPPCA*). The result is a set of candidates for each landmark class (see first row of Fig. 7). To find an initial position of landmark l_i, we consider points in the neighborhoods of umbilics that were labeled l_i.

The search starts in the nose tip region. As starting point, we select the vertex v of F, which is the umbilic that after *FPPCA* is the closest point to the medoid of the cluster of points labeled as nose tip. The new search space corresponds to the set of vertices v_k within the geodesic circle of radius r centered at v. For each v_k, *FPPCA* is applied. In this step, we only consider points v_k that are either labeled as nose tip or subnasal. This procedure is depicted in the second row of Fig. 7.

Next, we refine the positions of the right and left subalare. To this end, we start from the point v closest to the medoid of all points that were labeled as subnasal in the previous step. The algorithm proceeds by classifying points v_k in a geodesic neighborhood of radius r of v using *FPPCA*. In this step, we only consider points v_k that are labeled as right or left subalare. Since the *M-clusters* of these two classes strongly overlap, most of the labeled points are assigned to two classes and the non-relevant points are discarded (see third row of Fig. 7). Since the labeled vertices are distributed over both sides of the nose, we split up this set of vertices into two sets by performing a $k-means$ clustering with $k = 2$. The two new sets of vertices still have both labels, and we find the point closest to the medoid of each cluster as a possible candidate (see fourth row of Fig. 7). It remains to determine which of these points corresponds to the right subalare, and which one to the left.

3.6 Aligning Landmark Graph to Scan

So far, four points on the nose region have been selected and labeled. Due to the face symmetry, two of the points have the same labels. To solve this problem, a template P_a of the upper part of the face with the same structure as the landmark graph (see Fig. 2) is roughly aligned to the input scan F. This helps also to estimate the initial guess of the remaining landmarks: right inner eye corner, right outer eye corner, left inner eye corner, and left outer eye corner, which are labeled as 1, 2, 3 and 4, respectively.

We compute a rigid alignment $\mathbf{T}$ that best aligns the point set v_a from P_a with the point set v_b from F. The point set v_a corresponds to the points labeled 5 to 8 of P_a, and v_b corresponds to the four points on the nose region of F. As the labels 5 and

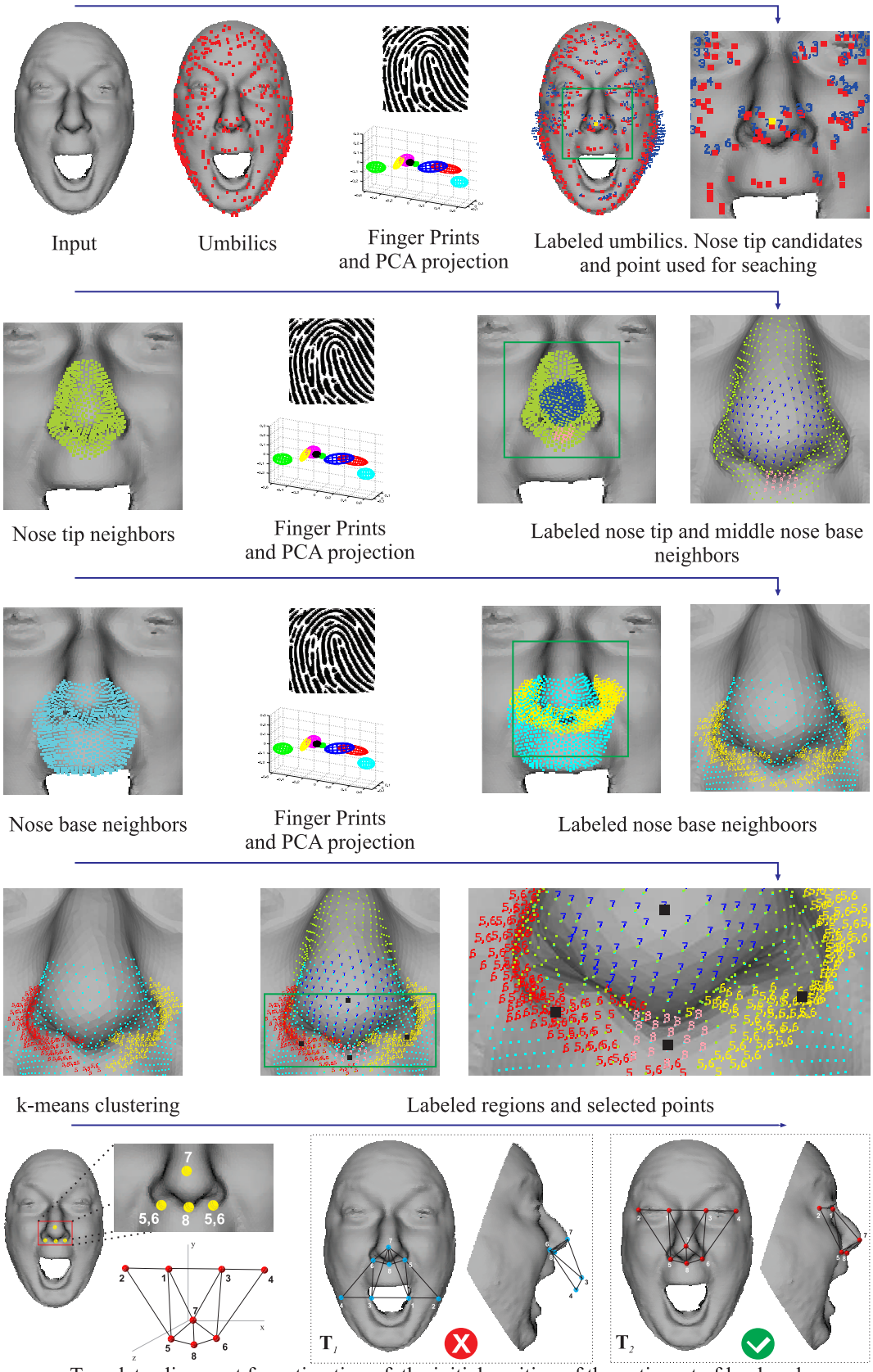


Fig. 7 Framework of the proposed initial alignment method.

6 of the points in v_b are unknown, there are two possible configurations for the alignment. As a result two linear transformations $\mathbf{T}_1$ and $\mathbf{T}_2$ are obtained. In order to select the transformation that produces a valid result, the transformed point sets $P_1 = \mathbf{T}_1 P_a$, and $P_2 = \mathbf{T}_2 P_a$, are computed. One of the transformations produces a vertical "flip" of the template, resulting in a wrong estimation of the coordinates of the points in the eye region. Therefore, the point set P_i that minimizes the sum of Euclidean distances to closest points on F is the correct transformation. This procedure is depicted in the fifth row of Fig. 7.

The locations of the transformed template vertices are used to define the search space region on which statistical inference is performed, as discussed in Section 3.2. The regions are defined as all points within distance r from the transformed points P_i.

4 Registration

In this section, we describe how a template is fitted to a 3D scan of the face. The input scan corresponds to a face of a subject performing a facial expression. Fitting a template to this scan is challenging because the facial geometry has large variations due to different face shapes and facial muscle movements. We propose a registration method, where the expression and the shape are fitted separately in order to handle the complexity of the problem. Fig. 8 shows an overview of the proposed method.

We address the facial expression fitting problem as a facial rigging problem. In facial rigging, a facial expression is produced by changing a set of parameters associated with the different regions of the face modeled using blendshapes. Conceptually, to generate a facial shape from a 3D rest pose face template, we just move a set of vertices to a new location, e.g., lift an eyebrow or open the mouth (see Fig. 9). In this sense and similar to the approach proposed by Li et al. [9], we model a facial expression as a linear combination of facial blendshapes (denoted by A_i), which are expressed as vectors of displacements from the rest pose (denoted by A_0).

4.1 Affine Alignment

To solve the fitting problem, the template A_0 in neutral pose is aligned to a scan F as follows. Both A_0 and F contain a set of landmarks denoted by $\bar{l}_i$ and l_i, respectively. The landmarks l_i were predicted using the method described in Section 3. The alignment is carried out by finding a 3×4 transformation matrix $\mathbf{T}_A$ that minimizes the energy

$$E_{lnd} = \sum_{i=1}^{L} \left(\mathbf{T}_A \bar{l}_i - l_i\right)^2, \tag{3}$$

with respect to the 12 parameters in $\mathbf{T}_A$ using a quasi-Newton approach starting from $\mathbf{T}_A$ as identity matrix.

4.2 Expression Fitting

We now outline how to fit the expression of the blendshape model to F. The aim of this step is to model expression variations using a small number of basis shapes. An expression can be generated using a small number of parameters as

$$P(\alpha_i) = A_0 + \sum_{i=1}^{j} \alpha_i A_i, \tag{4}$$

where A_0 corresponds to the rest pose, $A_i, i > 0$ correspond to the blendshape displacements, and α_i $(0 \leq \alpha_i \leq 1)$ are the blending weights of expression $P(\alpha_i)$. For each blendshape A_i, Fig. 9 shows the corresponding expressions. The 3D models used in both the creation of A_0 and the generation of A_i were obtained using a commercial software. Notice that mostly mouth displacements are considered. As the expressions are generated as a linear combination of displacements, to avoid exaggerated undesired expressions, it is important that no two blendshapes add the same kind of displacement. By using a blendshape model, the facial expression fitting problem is transformed into an optimization problem, where the value of each α_i has to be estimated.

Recall that A_0 and F are affinely aligned. We find the α_i that best match the expression of F by dividing $P(\alpha_i)$ into three regions: chin, mouth, and remaining face (as shown in Fig. 10). The division is motivated by the fact that the chin and lip regions vary drastically from one expression to another (mostly in terms of displacements). Thus it is desirable to inspect the quality of the fitting in each of these regions separately by assigning higher weights to points in these regions than to points in the remaining face.

To fit the expression, we use the energy

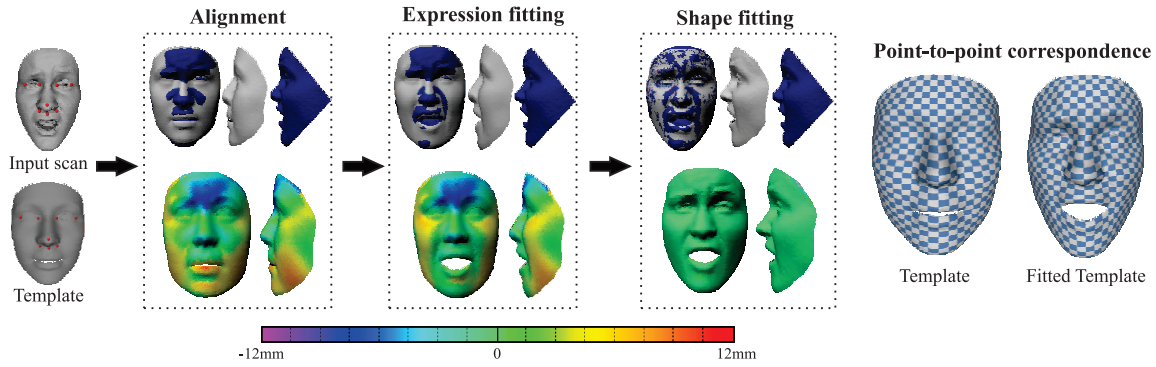


Fig. 8 Registration procedure. First, the template and the scan are aligned using the predicted landmarks. Second, the expression is fitted using a blendshape model. Finally, an energy-based surface fitting method is used to fit the shape. At the end, the overlap between the scan and the template is maximized and a point-to-point correspondence for the face shapes in different expressions is obtained.

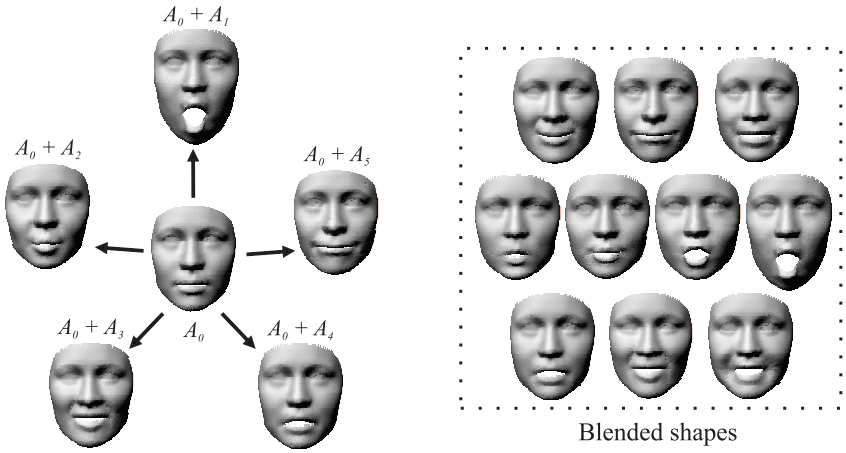


Fig. 9 Left: template rest pose A_0 and a set of blendshapes A_i. Right: examples of models generated as linear combinations of blendshapes.

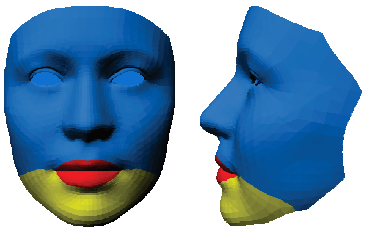


Fig. 10 Regions used in the expression fitting procedure.

$$E_{expr} = \sum_r \omega_r \langle (NN(p_r(\alpha_i)) - p_r(\alpha_i)), \mathbf{n}(NN(p_r(\alpha_i)))\rangle^2, \quad (5)$$

where $p_r(\alpha_i)$ are the vertices of $P(\alpha_i)$, $NN(p_r(\alpha_i))$ indicates the nearest neighbor point of $p_r(\alpha_i)$ on F, $\mathbf{n}(NN(p_r(\alpha_i)))$ is the unit outer normal vector of $NN(p_r(\alpha_i))$, $\langle .,. \rangle$ denotes the dot product of two vectors, and ω_r is a weight associated with $p_r(\alpha_i)$. The energy pulls each vertex of the template to the nearest point on the tangent plane of its nearest neighbor on F. The weight ω_r is used for two purposes: to give different weight to the mouth, chin, and remaining regions of the model, and to make the method more robust to both the presence of outliers and mis-oriented surfaces. To achieve the first goal, we set ω_r to either ω_{mouth}, ω_{chin}, or $\omega_{remaining}$, depending on the region containing $p_r(\alpha_i)$. To achieve the second goal, we only consider the nearest neighbor if the angle between the outer normal vectors of $p_i(\alpha_i)$ and $NN(p_r(\alpha_i))$ is small. Specifically, we set $\omega_{remaining}$ to zero if the angle is larger than φ. To force the fit to be exact, we set ω_{chin} and ω_{mouth} to zero if the angle is larger than $\varphi/2$. The expression is fitted by minimizing Eq. 5 with respect to the blending weights α_i. In our experiments we set φ to 80 degrees.

The minimization of E_{expr} is carried out in two stages. In the first stage, we inspect if some movement occurs in the chin. Once we know the position

of the chin, to refine the match with the expression of the input model, we need to inspect the positions of the lips. Based on this, the expression fitting procedure proceeds as follows: First, the weight ω_{mouth} is set to zero, thus the minimization is only guided by vertices that are not in the mouth region. In this step $\omega_{remaining}$ is set to one and ω_{chin} is defined as $1 - \left(V_{chin}^{valid}/V_{chin}\right)$, where V_{chin} is the number of vertices in the chin region and V_{chin}^{valid} is the number of valid nearest neighbors in this region. The second step begins when at least 80% of the vertices in the chin region have valid nearest neighbors. At this time, ω_{mouth} is set to $1 - \left(V_{mouth}^{valid}/V_{mouth}\right)$, where V_{mouth} is the number of vertices in the mouth region and V_{mouth}^{valid} is the number of valid nearest neighbors in this region. The minimization process ends when at least 60% of the vertices in the mouth region have valid nearest neighbors. This weight variation scheme ensures that the chin and mouth regions of $P(\alpha_i)$ match the expression of F. The threshold values for ω_{chin} and ω_{mouth} were chosen based on experimental observations.

This step fits the expression of the template to the expression of the scan. However, since the deformations are modeled by a small number of parameters, the deformation during this step is restricted, and fine shape details cannot be modeled by this step.

4.3 Shape Fitting

To find a more accurate local fitting, we next fit the shape of $P(\alpha_i)$ to the shape of F. For ease of notation, we use $P = P(\alpha_i)$ in the following.

The shape fitting is, again, treated as an optimization problem similar to the method proposed by Allen et al. [36] and extended by Li et al. [46]. The goal is to find a set of 3×4 transformation matrices $\mathbf{T}_i$ for each vertex p_i of P such that p_i is moved to the new location $\tilde{p}_i = \mathbf{T}_i p_i$ to fit the shape of F. The transformed version of P is denoted $\tilde{P}$. The transformation matrices $\mathbf{T}_i$ are obtained by minimizing an energy function, which is a weighted sum of three energy terms.

The first term is the data term

$$E_{data} = \sum_i \omega_i \left\langle \left(NN\left(\tilde{p}_i\right) - \tilde{p}_i\right), \mathbf{n}\left(NN\left(\tilde{p}_i\right)\right)\right\rangle^2, \tag{6}$$

where $NN\left(\tilde{p}_i\right)$ indicates the nearest neighbor of $\tilde{p}_i$ on F, and $\mathbf{n}\left(NN\left(\tilde{p}_i\right)\right)$ is the normalized outer normal of $NN\left(\tilde{p}_i\right)$. The weight ω_i is set to one if the angle between the outer normal vectors of $\tilde{p}_i$ and its nearest neighbor is at most 80 degrees, and to zero otherwise. The data term ensures that the template is deformed to resemble the input scan.

The second energy is a regularization term that encourages smooth transformations between neighboring vertices of the mesh. We call this energy regularization energy E_{reg} and define it as

$$E_{reg} = \sum_{(i,j)\in E^{(\tilde{P})}} (\mathbf{T}_i - \mathbf{T}_j)^2, \tag{7}$$

where $E^{(\tilde{P})}$ is the set of edges of $\tilde{P}$. This term prevents adjacent parts of P from being mapped to disparate parts of F, and also encourages similarly-shaped features to be mapped to each other [36].

The final energy term encourages the transformation matrices to be rigid. The rigid energy E_{rigid}, which measures the deviation of the column vectors of $\mathbf{T}_i$ from orthogonality and unit length, is defined as

$$\begin{aligned} E_{rigid} = \sum_{i=1}^{r} \Bigg(& \left(\left(\mathbf{a}_1^i\right)^{\mathrm{T}} \mathbf{a}_2^i\right)^2 + \left(\left(\mathbf{a}_1^i\right)^{\mathrm{T}} \mathbf{a}_3^i\right)^2 + \\ & \left(\left(\mathbf{a}_2^i\right)^{\mathrm{T}} \mathbf{a}_3^i\right)^2 + \left(1 - \left(\mathbf{a}_1^i\right)^{\mathrm{T}} \mathbf{a}_1^i\right)^2 + \\ & + \left(1 - \left(\mathbf{a}_2^i\right)^{\mathrm{T}} \mathbf{a}_2^i\right)^2 + \left(1 - \left(\mathbf{a}_3^i\right)^{\mathrm{T}} \mathbf{a}_3^i\right)^2 \Bigg), \end{aligned} \tag{8}$$

where $\mathbf{a}_1^i$, $\mathbf{a}_2^i$, $\mathbf{a}_3^i$ are the first three columns vectors of $\mathbf{T}_i$.

The energy terms described above are combined in the weighted sum

$$E_{shape} = \omega_{data} E_{data} + \omega_{reg} E_{reg} + \omega_{rigid} E_{rigid}. \tag{9}$$

The shape is fitted by minimizing E_{shape} with respect to the parameters $\mathbf{T}_i$. We start by encouraging smooth and rigid transformations by setting $\omega_{data} = 1$, $\omega_{reg}^0 = 20000$, and $\omega_{rigid}^0 = 10$. Similar to Li et al. [46], whenever the energy change is negligible, we relax the weights as $\omega_{reg}^t = 0.5\omega_{reg}^{t-1}$ and $\omega_{rigid}^t = 0.5\omega_{rigid}^{t-1}$ to give more weight to the data term. This allows the template to deform towards the scan. The algorithm iterates until the relative change in energy $(E_{shape}^{i-1} - E_{shape}^i)/E_{shape}^{i-1}$, where i is the iteration number, is less than 0.0001. For each set of weights, we use a quasi-Newton approach [47] to solve the optimization problem, and we perform at most 1000 iterations.

As our template only includes the shape of the face and the template can be free deformed during the shape fitting, in both expression and shape fitting procedures, the boundary points of the input model are ignored to prevent that the fitting results include noise shapes from the hair or ears of the input model.

5 Experiments and results

5.1 Database

We use the BU-3DFE [48] database for our experiments. The database consists of 3D face models from 100 subjects (56 Females and 44 Males) in neutral pose and with the following facial expressions: *surprise*, *happiness*, *disgust*, *sadness*, *anger* and *fear*. There are four scans of each facial expression, corresponding to different levels of intensity from *low* to *highest*. As a file containing the raw data of each scan is also available, there are a total of 50 files per subject, 25 raw scans and 25 corresponding to the cropped faces. Fig. 11 shows snapshots of different scans from the BU-3DFE database. In this work, we use a subset of 700 3D models corresponding to the cropped faces of the subjects performing the expressions in the highest level.

5.2 Landmark prediction accuracy

We use two different subsets of models of 50 subjects (25 females and 25 males) to train the landmark prediction model. First, we use a subset T_n consisting of 50 models of subjects in neutral pose as training set. Second, we use a subset T_e consisting of 350 models of the same 50 subjects in neutral pose and performing six different facial expressions as training set. As T_n covers the shape variability and T_e covers both shape and expressions variability, we are able to evaluate the importance of the variabilities considered in the training sets. The accuracy of the landmark prediction algorithm is evaluated over the remaining 50 subjects of the database (31 females and 19 males). The test database corresponds to 350 models of subjects in both neutral pose and when performing six different facial expressions.

To evaluate the accuracy of the landmark prediction algorithm, we compute the error of the Euclidean distance between a manually located landmark l_i and its corresponding estimation $\hat{l}_i$. We compute the mean, the standard deviation and the maximum of the error. We also compute the detection rates by counting the percentage of test models where the landmark $\hat{l}_i$ was predicted with an error below $10mm$ ($T10$), $20mm$ ($T20$), and $30mm$ ($T30$). Tables 1 and 2 show the results of the evaluation for the test with T_n and T_e as training databases, respectively.

The best landmark prediction results were obtained when T_e is used for training. In both experiments, the landmarks located in the nose region are better predicted than the ones located in the eye region. The tip of the nose is predicted with the lowest error and the outer corners of the eyes are predicted with the highest error. One of the reasons that the outer corners of the eyes are not predicted as well as the other landmarks is that the initial position is found based on the alignment of the landmark template (see Fig. 7). This adds an estimation error that is reflected in the values of the standard deviation. The values of the detection rates show the improvement in accuracy of the landmark prediction when T_e is used as training set. This indicates that for the configuration of the landmark prediction model used in this work, the variations due to both shape and expression have to be considered.

We compared our results of landmark prediction with two approaches where the BU-3DFE database is also used for testing. Segundo et al. [19] used 2500 range images obtained from the raw data, and Nair and Cavallaro [22] used 2350 of the 2500 3D cropped face models available. Table 3 shows the mean of the error of the landmark prediction. For all the landmarks, our approach outperforms the approach of Nair and Cavallaro [22]. Compared to Segundo et al. [19], for all the landmarks but the nose tip the mean error is similar. Recall however that Segundo et al. [19] use a more challenging dataset for testing.

Although the obtained landmark prediction error appears to be high, it is still possible to obtain a proper point-to-point correspondence since the landmarks are only used to align the template to the scan. Afterwards, a non-rigid iterative closest point framework is used to deform the expression and shape of the template. Fig. 12 shows some examples of the landmark prediction results over models of subjects with different facial shapes and performing different expressions.

In the following, we use T_e as training dataset. Furthermore, we only consider the models where

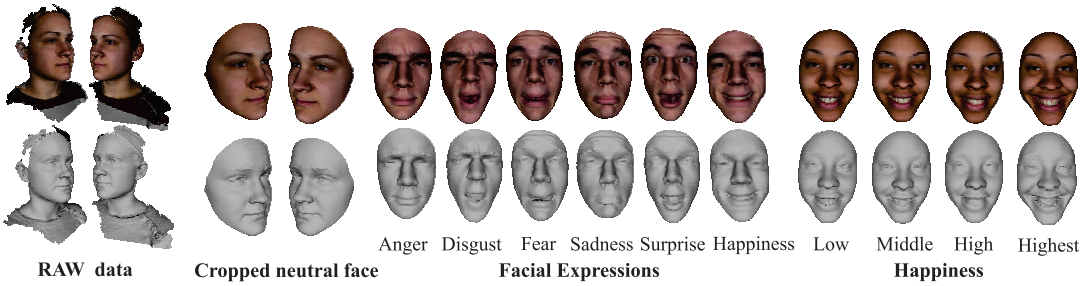


Fig. 11 Characteristics of the BU-3DFE database.

Landmark	**Mean ± Std** [mm]	**Max.** [mm]	$T10$ [%]	$T20$ [%]	$T30$ [%]
Right inner eye corner	10.35 ± 6.13	33.93	53.71	87.14	92.57
Right outer eye corner	11.79 ± 7.77	34.73	27.71	85.71	93.43
Left inner eye corner	11.63 ± 6.82	34.16	44.57	86.57	94.00
Left outer eye corner	12.57 ± 7.23	34.29	31.43	89.14	95.71
Right subalare	9.96 ± 6.59	33.49	66.00	86.86	98.00
Left subalare	10.93 ± 6.87	34.15	55.14	87.43	94.29
Nose tip	7.42 ± 5.64	32.03	82.57	92.00	96.86
Subnasal	7.12 ± 5.87	33.75	84.57	87.43	95.43

Table 1 Error of landmark prediction with training set T_n. $T10$, $T20$, and $T30$ correspond to the detection rates with a tolerance of $10mm$, $20mm$ and $30mm$, respectively.

Landmark	**Mean ± Std** [mm]	**Max.** [mm]	$T10$ [%]	$T20$ [%]	$T30$ [%]
Right inner eye corner	6.14 ± 4.54	34.39	80.86	95.14	97.43
Right outer eye corner	8.49 ± 6.12	34.54	62.29	95.14	97.71
Left inner eye corner	6.75 ± 4.21	33.75	84.00	96.57	98.29
Left outer eye corner	9.63 ± 5.82	34.63	63.14	93.43	98.86
Right subalare	7.17 ± 3.3	32.23	85.43	95.14	97.43
Left subalare	6.47 ± 3.07	32.3	89.71	96.86	97.43
Nose tip	5.87 ± 2.7	29.91	93.71	97.43	100
Subnasal	5.57 ± 2.03	30.26	95.43	98.29	99.71

Table 2 Error of landmark prediction with training set T_e. $T10$, $T20$, and $T30$ correspond to the detection rates with a tolerance of $10mm$, $20mm$ and $30mm$, respectively.

Fig. 12 Examples of the landmark prediction results. Red and green spheres correspond to the manually placed and predicted landmarks, respectively. First row: female subjects; Second row: male subjects.

Landmark	[19] [mm]	[22] [mm]	Our Method [mm]
Right inner eye corner	6.33	20.46	6.14
Right outer eye corner	N.A.	12.11	8.49
Left inner eye corner	6.33	19.38	6.75
Left outer eye corner	N.A.	11.89	9.63
Right subalare	6.49	N.A.	7.17
Left subalare	6.66	N.A.	6.47
Nose tip	1.87	8.83	5.87
Subnasal	N.A.	N.A.	5.57

Table 3 Comparison of mean errors of our method and the approaches of Segundo et al. [19] and Nair and Cavallaro [22].

all landmarks are predicted within $30mm$ of the ground truth (332 of the 350 models).

5.3 Registration

We tested our dense point-to-point correspondence algorithm on 332 models.

5.3.1 Landmark Fitting Accuracy

To evaluate the accuracy of the registration, we compute the error in the location of manually placed landmark points present in the BU-3DFE database that are not considered for the alignment. The error corresponds to the Euclidean distance between a manually placed point and its corresponding location after registration. The set of points considered for the evaluation (see Fig. 4) includes 20 points on the eyebrows (10 left, 10 right), 12 points on the eye contours (6 left, 6 right), 12 points in the nose region, 12 points on the outer contour of the lips, 3 points on the chin, and 12 points on the face contour (6 left, 6 right).

Table 4 shows the mean, the standard deviation, and the maximum of the error, as well as the detection rates. In this case, we compute the mean and standard deviation over all points in a region and over all 332 models used for correspondence computation. Furthermore, we compute the detection rates by counting the percentage of test models where all the points belonging to the same region were predicted with an error below $10mm$ ($T10$), $20mm$ ($T20$), and $30mm$ ($T30$).

The points on the eye contour and the nose region were found with lower mean error and variation than the points on the mouth, chin, and eyebrows regions. This situation is expected because the movements in the eyebrows and mouth are more pronounced than in the other areas. The big difference between the error on the face contour points with respect to the other regions is mainly because of there are no strong anatomical attributes that help to define the face contour, which results in highly inconsistent manually placed markers across the database.

Next, we discuss the quality of the results after the final shape fitting step. Fig. 13 shows the cumulative distribution of the number of models where the error at the landmark points not used for registration is below a threshold (due to noise, the set of ground truth points on the face contour was not included). Note that even when the error at some points is slightly high, we found that both the face regions and the surface geometry of the input models are consistently matched with their counterparts in the deformed template.

5.3.2 Surface Fitting Accuracy

To evaluate the accuracy of the fitting, we compute the Modified Hausdorff Distance (*MHD*), which is a metric for shape comparison that measures the degree of mismatch between two points sets. Therefore, it is useful to demonstrate the quality of a registration algorithm [29]. The *MHD* is defined as [49]:

$$MHD(P,F) = \frac{1}{N_p}\sum_{i=1}^{N_p} \min_{\mathrm{f}_j \in F} |p_i, \mathrm{f}_j| \,, \tag{10}$$

where $|p_i, \mathrm{f}_j|$ is the Euclidean distance between vertices of the template P and the vertices of the input model F, and N_p is the number of vertices of P. The *MHD* represents the average of the minimum Euclidean distance of the vertices of P, to

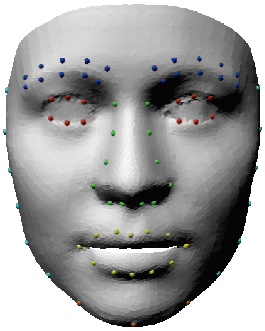
Points		Mean ± Std [mm]	Max. [mm]	$T10$ [%]	$T20$ [%]	$T30$ [%]
	Left Eyeb.	6.28 ± 3.30	25.36	52.87	98.79	100
	Right Eyeb.	6.75 ± 3.51	23.59	45.62	98.19	100
	Left Eye	3.25 ± 1.84	12.53	98.19	100	100
	Right Eye	3.81 ± 2.06	12.24	96.07	100	100
	Nose	3.96 ± 2.22	16.97	87.61	100	100
	Mouth	5.69 ± 4.45	45.36	52.57	94.26	98.79
	Chin	7.22± 4.73	33.80	58.01	95.47	99.39
	L. Face	18.48± 8.52	52.17	0.60	22.36	64.05
	R. Face	17.36± 9.17	58.36	0.30	22.96	60.12

Table 4 Error at landmark points not used for registration. Left: set of points. Right: summary of errors.

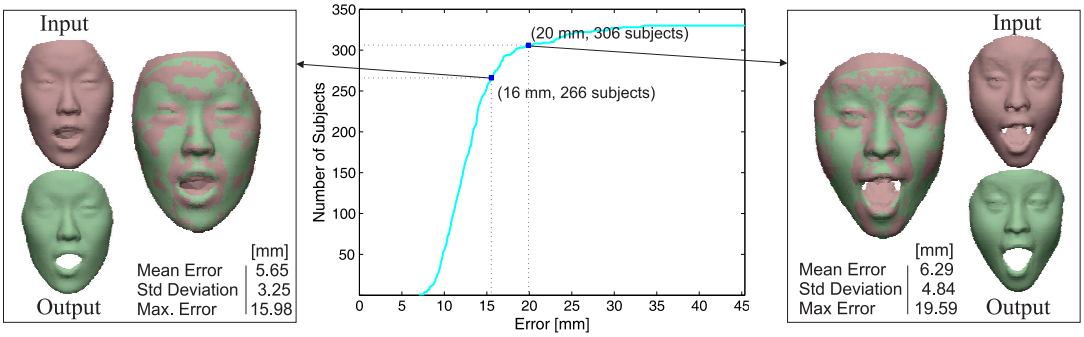


Fig. 13 Cumulative distribution of the number of models where the error at all the landmark points not used for registration is below a threshold. Example of registration results (left and right). Error distribution (center).

which F is registered [29]. The values of the average, standard deviation and maximum of the MHD for the 332 tested models were $1.42mm$, $0.56mm$ and $3.66mm$, respectively. This shows that our method has the ability of keeping the overall shape during the fitting.

In addition, the bottom row of Fig. 17 shows the histograms and the false color visualization of the mean magnitude and standard deviation of the distance between the surfaces F and P computed over all 332 models. For every point p_r on P, its nearest neighbor $NN(p_r)$ on F is determined, the distance from p_r to the tangent plane of $NN(p_r)$ corresponds to distance between the surfaces. As most of the values of the distances are concentrated between 0 and $1mm$, in order to improve the visualization, the color map was clamped to this range. Notice the variation in the lower lip and chin area, which are the regions where the surface is deformed most due to the facial expressions.

5.3.3 Visual Evaluation

Next, we show some examples that summarize the results of the expression and shape matching stages of the registration process. The third column of Fig. 14 shows examples of the expression fitting results for six different kinds of facial expression. In all cases, the expression of the mouth region of the input model is properly matched after linear blending. The fourth column of Fig. 14 shows examples of the shape fitting results. The models are color-coded with respect to the signed distance from the input scan. Note that most points on the models are within $2mm$ of the scan. Furthermore, notice how the different expressions in the eyebrows are properly fitted. In order to visualize the quality of the correspondences, a chess-board texture (with some facial features colored) was applied to the template model (see right of Fig. 14). Results of the texture transferring show that in most of the face regions, the shape of the deformed template matches the shape of the input model.

For our method, which uses nearest neighbors to guide the deformation, the highest level of expression is the most difficult to register. All of the experiments outlined so far have considered this case. Fig. 15 illustrates two examples of registering different levels of the same subject in the same ex-

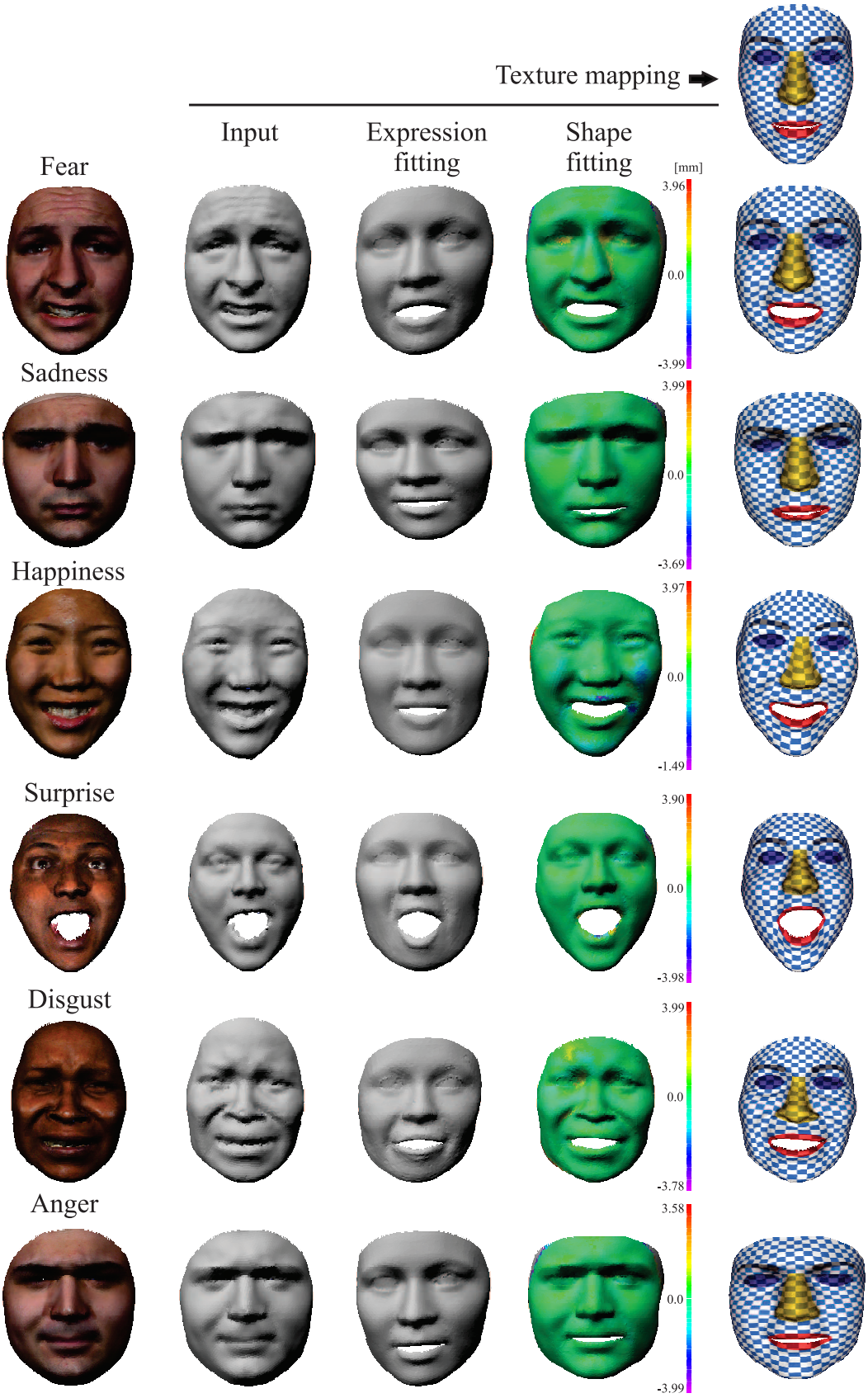


Fig. 14 Examples of registration results. The input, fitted expression, error mapped, and texture mapped models are provided for each example.

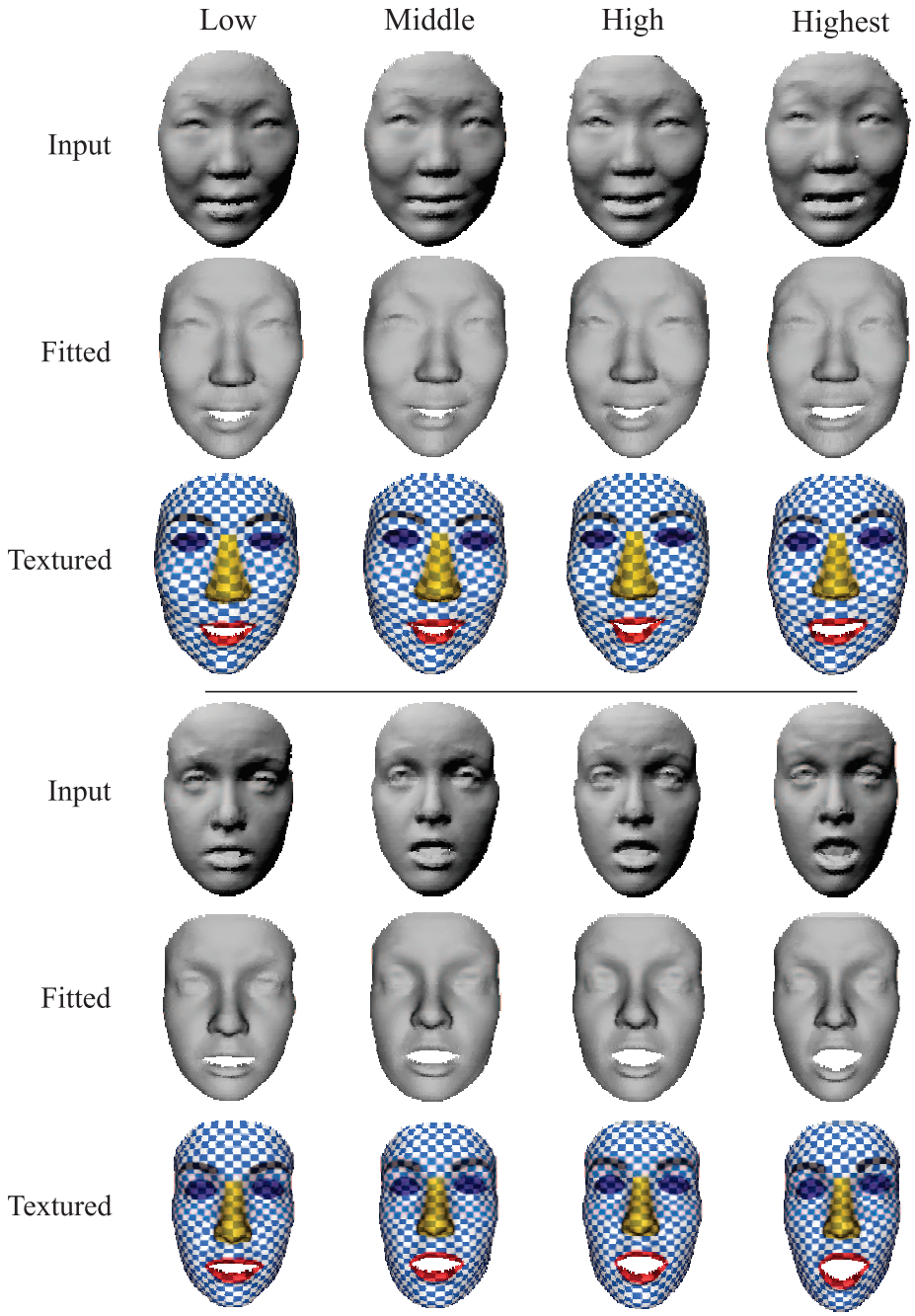


Fig. 15 Examples of fitting to models of the same subject performing an expression in different levels. Fear (first three rows). Surprise (last three rows). For each example, first, second, and third rows are the input, output, and textured models.

pression. Note that the visual differences between the quality of the results are insignificant.

Finally, we discuss the running time of our method. On a standard PC (2.4 GHz processor), the typical time to predict the set of landmarks for the initial alignment is about 5 seconds for rough alignment and about 176 seconds for the refinement of the position. The typical time for expression and shape fitting is about 6 seconds and 28 seconds, respectively.

5.4 Comparison to 3D Morphable Model

We compare our registration results to the results obtained using the commonly used 3D morphable model (3DMM) [2], which is a statistical model that encodes information about a set of training

shapes. In order to use the morphable model for fitting, we first need to build such a model. To this end, we use the 50 subjects in highest expression levels that were used for training for the landmark detection part. Before computing the model, we first need to parameterize the training shapes. We achieve this using manually placed marker positions that guide a non-rigid iterative closest point deformation. This step to parameterize a training set in a semi-automatic way is time-consuming. We analyzed the morphable model and found that retaining 50 principal components yields a compact, yet general model.

We fit the morphable model to the data by first using the landmarks predicted by our method to rigidly align the scan to the model, and by subsequently minimizing the energy E_{data} defined in Equation 6 with respect to the model parameters.

Note that unlike 3DMM, our method does not require a parameterized training set as a start. Furthermore, in the future, the method proposed in our work could help building statistical models without the need to parameterize a training set in a semi-automatic way.

We compare our results to 3DMM in two ways. First, we provide an evaluation of the obtained fitting results. Since for the 3DMM, the amount of displacement during the fitting is restricted to the one learned from the training data, our method can fit local shape details more accurately than 3DMM, as can be seen in the four examples shown in Fig. 16. As most of the values of the distances are concentrated between 0 and $1mm$, in order to improve the visualization, the color map was clamped to this range. Fig. 17 compares the histograms and the false color visualization of the mean magnitude and standard deviation of the distance between the surfaces F and P computed over all 332 models. Notice that while both methods lead to good fitting results overall, our method has lower mean error in localized areas such as the tip of the nose or the eyebrows. The reason is that unlike our method, 3DMM cannot fit to localized shape detail such as raised eyebrows, because 3DMM restricts the search space for the correspondence search to the variations observed in the training data.

Second, we compare the results for the application of expression recognition. Note that this experiment is primarily intended to give a comparative evaluation between 3DMM and our method, and not to introduce a new method for expression recognition.

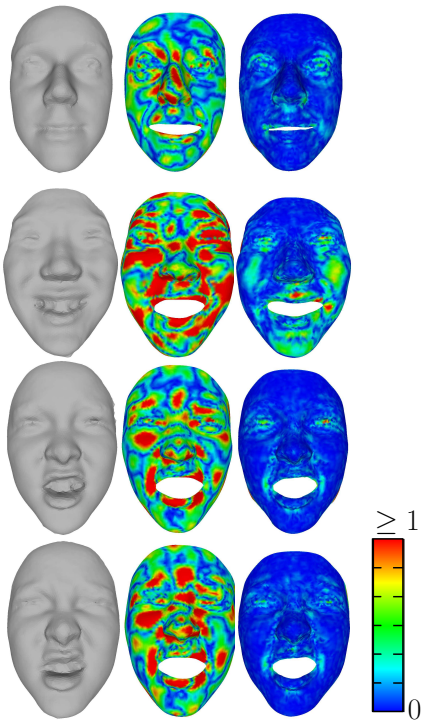


Fig. 16 Comparison of shape distance (in mm) of 3DMM fitting and our results. Left to right: input scan, 3DMM fitting, our result.

In the following experiment, we aim to recognize (the highest expression levels of) the expressions anger, happiness, and surprise. The features used for our experiment are based on anatomical facial landmarks and are computed following the methodology described in Rabiu et al. [50]. The feature selection, classification and evaluation is carried out using the pattern recognition tool developed by Duin et al. [51] with a support-vector classifier based on a 2nd order polynomial kernel. For training, we use features derived from the ground truth landmarks of the 50 subjects that were used for training for the landmark detection part. For testing, we use all fitting results (with expressions anger, happiness, or surprise). The overall expression recognition rate using the models fitted with 3DMM is 61.1%, while the overall expression recognition rate using the models fitted using our method is 77.7%. While neither of these results is competitive with human experts, who achieve a recognition rate of 98.1% [48], the experiment shows that our method achieves significantly higher recognition rates than 3DMM. The reason is that our method can fit better to local shape details, as discussed above.

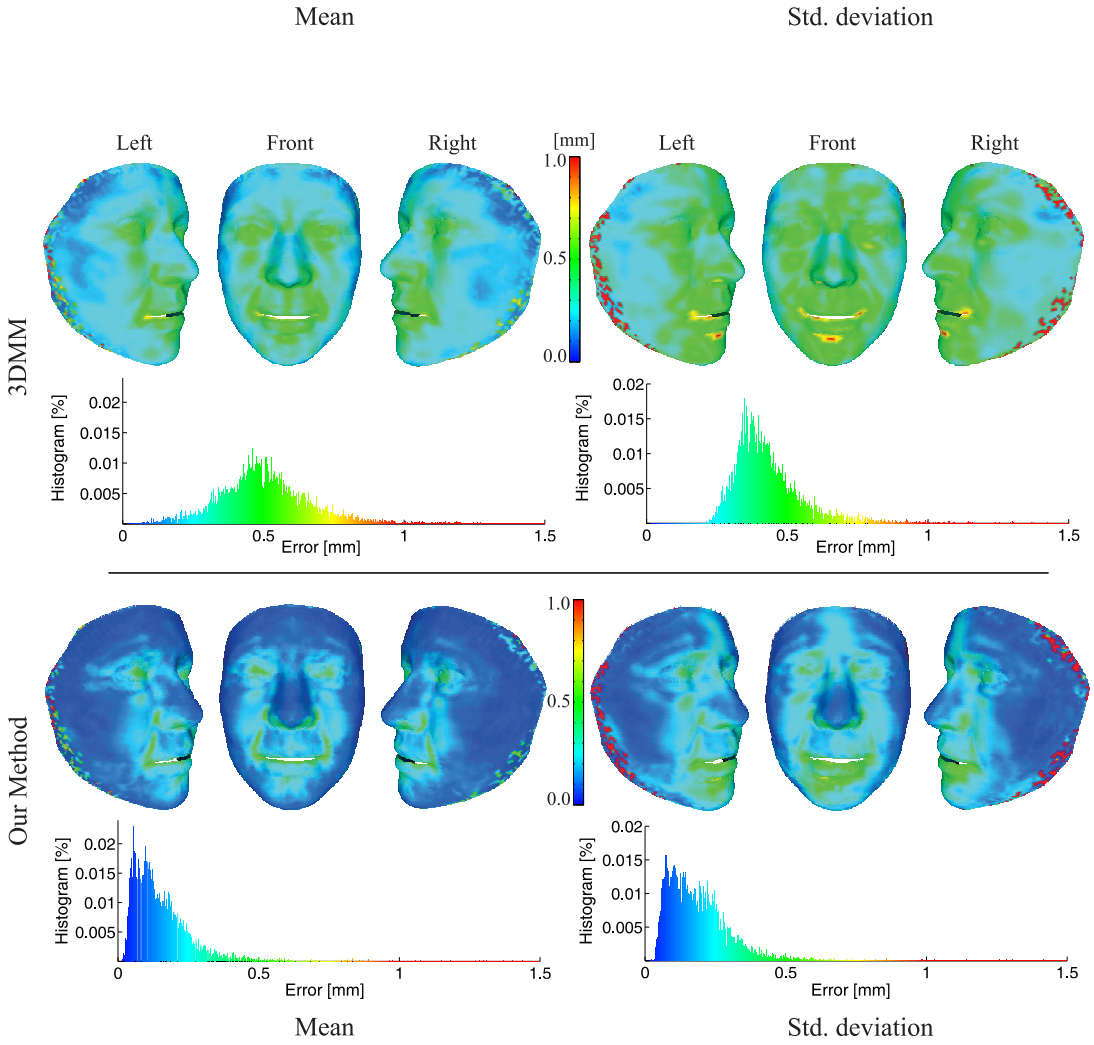


Fig. 17 Distance between the surface of the template P and the surface of input model F. Histograms and the false color visualization (different views) of the magnitude of the mean and standard deviation of the distance.

5.5 Application

Finally, we apply our fitting results to building a statistical shape space that allows to explore the identity and expression variations of a database of faces separately. To this end, we use our registration results to compute a multilinear model [52]. The multilinear model expresses each face using one weight vector ω_i for identity and a second weight vector ω_e for expression. We can modify the expression of a subject by keeping ω_i fixed while modifying ω_e. Similarly, we can modify the identity while preserving the expression by keeping ω_e fixed while modifying ω_i. This is shown in Fig. 18. Here, the faces shown in boxes are the registered faces of the database that were used to compute the multilinear model, and the remaining faces were generated by fixing ω_i to one identity and varying ω_e (top row) and by fixing ω_e to the weight of happiness and varying ω_i (bottom row). Note that in this way, realistic looking new expressions and identities can be generated, respectively.

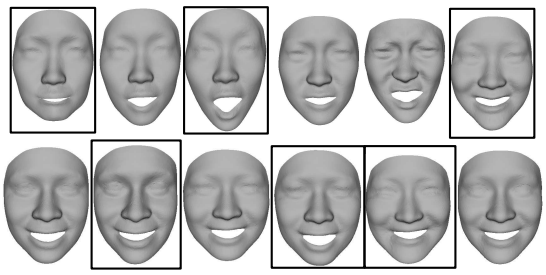


Fig. 18 Real models used to compute the multilinear model (shown in boxes) and synthetic models generated from the multilinear model.

5.6 Limitations

Our method has some limitations. Sometimes, not all local areas of a face are fitted accurately. Most of the incorrect shape fitting occurs on the inner parts of the lips. As the input scans have information in the area of the teeth, which is not considered in the template model, the algorithm converges to this region, thereby causing miscorrespondences during the shape fitting. Fig. 19 shows an example of the limitations in the shape fitting. Notice how the expression is matched correctly, but the corners of the mouth are not well located, which causes an incorrect fitting on the mouth and chin regions.

Another limitation occurs for models with occluded parts. Fig. 20 shows the result of the proposed point-to-point correspondence approach for a model of a subject where the mouth is occluded by a hand. In this case, the template is correctly fitted to areas not affected by the occlusion, but occluded regions cause unlikely face shapes.

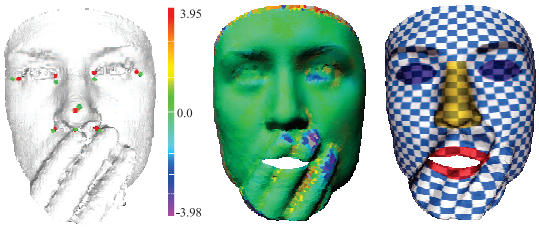


Fig. 20 Challenging test scenario. Mapped error models correspond to the fitting result. Test was carried out over one model of the Bosphorus database [53].

6 Conclusions

This paper presented a fully automatic method to compute dense point-to-point correspondences between a set of human face scans with varying expressions. The proposed approach proceeds by learning local shape descriptors and spatial relationships for a set of landmark points. For a new scan, the approach first predicts the landmark points by performing statistical inference on the learned model. The approach then fits a template to the scan in two stages. The first stage fits the expression of the template to the expression of the scan using the predicted landmark points. The second stage fits the shape of the template to the shape of the scan using a non-rigid iterative closest point technique. We applied our approach to 350 models of the BU-3DFE database, and evaluated the results both qualitatively and quantitatively. We showed that for 94.9% of the models, the landmarks are predicted with an error below $30mm$, and that for most of the models, a consistent correspondence is found. Furthermore, we evaluated the algorithm on a challenging case of a face with occlusion.

The failure cases of the algorithm are mostly caused by noisy data in the mouth area. For future work we plan to design algorithms that can handle this challenging scenario. It is also of interest to test the algorithm on a large database of models with different types of occlusion, such as models wearing eyeglasses (e.g., models from Bosphorus database [53]) and on data acquired using different types of sensors. Finally, with the availability of inexpensive depth cameras, dynamic data is becoming increasingly important. Interesting future work includes to extend the proposed algorithm to compute correspondences of dynamic facial data in a fully automatic framework.

Acknowledgements This work was supported by the program "*Créditos condonables para estudiantes de Doctorado*" from *COLCIENCIAS* - Colombia, by the program "*Convocatoria de apoyo a tesis de posgrado - Doctorados*" from *Dirección de Investigaciones de Manizales* - National University of Colombia, and by the Cluster of Excellence *Multimodal Computing and Interaction* within the Excellence Initiative of the German Federal Government. We thank Timo Bolkart for help in conducting the comparison to 3DMM, and Jonathan Boisvert, Timo Bolkart, Alan Brunton, and Pengcheng Xi for helpful discussions.

References

1. I. Dryden and K. Mardia. *Statistical Shape Analysis*. Wiley, 2002.
2. V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Conference on Computer Graphics and Interactive Techniques*, pages 187–194, 1999.
3. D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang, and W. Gao. Efficient 3D reconstruction for face recognition. *Pattern Recognition*, 38(6):787–798, 2005.
4. S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3D morphable model using linear shape and texture error functions. In *IEEE International Conference on Computer Vision*, pages 3–19, 2002.
5. S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *IEEE International Conference on Computer Vision*, volume 1, pages 59–66, 2003.
6. A. Brunton, C. Shu, J. Lang, and E. Dubois. Wavelet model-based stereo for fast, robust face reconstruction. In *Canadian Conference on Computer and Robot Vision*, pages 347–354, 2011.

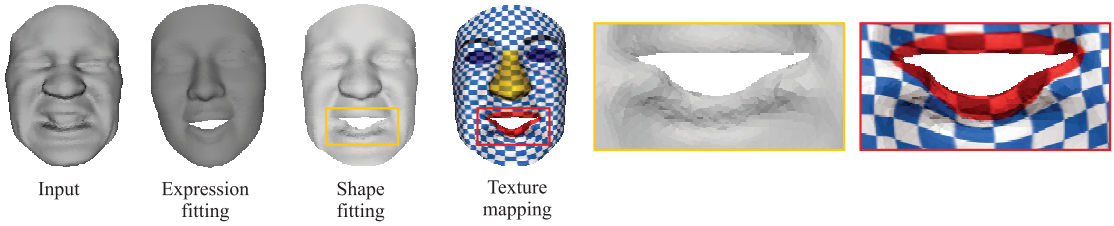


Fig. 19 Incorrect shape fitting. The differences in topology of the input and template meshes cause incorrect expression and shape fitting.

7. O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 3(6):1681–1707, 2011.
8. P. Xi and C. Shu. Consistent parameterization and statistical analysis of human head scans. *The Visual Computer*, 25(9):863–871, 2009.
9. H. Li, T. Weise, and M. Pauly. Example-based facial rigging. *ACM Transactions on Graphics (SIGGRAPH)*, 29(4):32:1–32:6, 2010.
10. E. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2006.
11. M. Cox, S. Sridharan, S. Lucey. and J. Cohn. Least Squares Congealing for Unsupervised Alignment of Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8 , 2008.
12. Y. Tong, X. Liu, F. Wheeler, and P. Tu. Semi-supervised facial landmark annotation. *Computer Vision and Image Understanding*, 116:922–935, 2012.
13. T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.
14. S. Mehryar, K. Martin, K. Plataniotis, and S. Stergiopoulos. Automatic landmark detection for 3D face image processing. In *IEEE Congress on Evolutionary Computation*, pages 1–7, 2010.
15. E. Vezzetti and F. Marcolin. 3D human face description: landmarks measures and geometrical features. *Image and Vision Computing*, 30(10):750–761, 2012.
16. Z. Ben Azouz, C. Shu, and A. Mantel. Automatic locating of anthropometric landmarks on 3D human models. In *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 750–757, 2006.
17. S. Berretti, B. Ben Amor, M. Daoudi, and A. del Bimbo. 3D facial expression recognition using SIFT descriptors of automatically detected keypoints. *The Visual Computer*, 27(11):1021–1036, 2011.
18. C. Creusot, N. Pears, and J. Austin. 3D face landmark labelling. In *Proceedings ACM workshop on 3D object retrieval*, pages 27–32, 2010.
19. M. Segundo, L. Silva, O. Pereira, and C. Queirolo. Automatic face segmentation and facial landmark detection in range images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 40(5):1319–1330, 2010.
20. P. Perakis, T. Theoharis, G. Passalis, and I. Kakadiaris. Automatic 3D facial region retrieval from multi-pose facial datasets. In *Eurographics Workshop on 3D Object Retrieval*, pages 37–44, 2009.
21. P. Perakis, G. Passalis, T. Theoharis, and I. Kakadiaris. 3D facial landmark detection & face registration: A 3D facial landmark model & 3D local shape descriptors approach. Technical Report TP–2010–01, Computer Graphics Laboratory, University of Athens, 2010.
22. P. Nair and A. Cavallaro. 3-D face detection, landmark localization, and registration using a point distribution model. *IEEE Transactions on Multimedia*, 11(4):611–623, 2009.
23. X. Lu and A. Jain. Automatic feature extraction for multiview 3D face recognition. In *International Conference on Automatic Face and Gesture Recognition*, pages 585–590, 2006.
24. Xiaoguang L., A. Jain, and D. Colbry. Matching 2.5D face scans to 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):31–43, 2006.
25. J. Guo, X. Mei, and K. Tang. Automatic landmark annotation and dense correspondence registration for 3D human facial images. *BMC Bioinformaics*, 14:232, 2013.
26. Y. Sun and M. Abidi. Surface matching by 3D point's fingerprint. In *IEEE International Conference on Computer Vision*, volume 2, pages 263–269, 2001.
27. A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.
28. K. Chang, K. Bowyer, and P. Flynn. Multiple nose region matching for 3D face recognition under varying facial expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1695–1700, 2006.
29. G. Passalis, P. Perakis, T. Theoharis, and I. Kakadiaris. Using facial symmetry to handle pose variations in real-world 3D face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1938–1951, 2011.
30. I. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, L. Yunliang, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):640–649, 2007.
31. I. Mpiperis, S. Malassiotis, and M. Strintzis. Bilinear models for 3D face and facial expression recognition. *IEEE Transactions on Information Forensics and Security*, pages 498–511, 2008.
32. Y. Huang, X. Zhang, Y. Fan, L. Yin, L. Seversky, J. Allen, T. Lei, and W. Dong. Reshaping 3D facial scans for facial appearance modeling and 3D facial

expression analysis. *Image and Vision Computing*, 30(10):681–796, 2012.

33. Xiaoguang L. and A. Jain. Deformation modeling for robust 3D face matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1377–1383, 2006.
34. C. Basso, P. Paysan, and T. Vetter. Registration of expressions data using a 3D morphable model. In *International Conference on Automatic Face and Gesture Recognition*, pages 205–210, 2006.
35. B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3D face recognition with a morphable model. In *IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6, 2008.
36. B. Allen, B. Curless, and Z. Popović. The space of human body shapes: Reconstruction and parametrisation from range scans. *ACM Transactions on Graphics (SIGGRAPH)*, 22(3):587–594, 2003.
37. S. Wuhrer, C. Shu, and P. Xi. Landmark-free posture invariant human shape correspondence. *The Visual Computer*, 27(9):843–852, 2011.
38. A. Bronstein, M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006.
39. A. Bronstein, M. Bronstein, and R. Kimmel. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, 16(1):188–197, 2007.
40. T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (SIGGRAPH)*, 30(4):77:1–77:10, 2011.
41. S. Wuhrer, Z. Ben Azouz, and C. Shu. Semi-automatic prediction of landmarks on human models in varying poses. In *Canadian Conference on Computer and Robot Vision*, pages 136–142, 2010.
42. T. Cox and M. Cox. *Multidimensional Scaling, Second Edition*. Chapman & Hall CRC, 2001.
43. J. Yedidia, W. Freeman, and Y. Weiss. *Understanding Belief Propagation and Its Generalizations*. Science & Technology Books, 2003.
44. J. Han and M. Kamber. *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers, 2006.
45. F. Cazals and M. Pouget. Smooth surfaces, umbilics, lines of curvatures, foliations, ridges and the medial axis: a concise overview. Technical Report RR-5138, INRIA, 2004.
46. H. Li, B. Adams, L. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 28(5):175:1–175:10, 2009.
47. D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.
48. L. Yin, X. Wei, J. Wang, Y. Sun, and M. Rosato. A 3D facial expression database for facial behavior research. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 211–216, 2006.
49. Y. Gao. Efficiently comparing face images using a modified hausdorff distance. In *IEEE Conference on Vision, Image and Signal Processing*, pages 346–350, 2003.
50. H. Rabiu, M. Saripan, S. Mashohor, and M. Marhaban. 3d facial expression recognition using maximum relevance minimum redundancy geometrical features. *EURASIP Journal on Advances in Signal Processing*, 1:213, 2012.
51. R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. Tax, and S. Verzakov. *PRTools4.1, A Matlab Toolbox for Pattern Recognition*. Delft University of Technology, 2007.
52. D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *Transactions on Graphics (Proc. SIGGRAPH)*, 24(3):426–433, 2005.
53. A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56, 2008.