# Towards View-Invariant Expression Analysis Using Analytic Shape Manifolds

Sima Taheri, Pavan Turaga and Rama Chellappa
Center for Automation Research, UMIACS
University of Maryland, College Park, MD 20742
Email: {taheri, pturaga, rama}@umiacs.umd.edu

*Abstract*— **Facial expression analysis is one of the important components for effective human-computer interaction. However, to develop robust and generalizable models for expression analysis one needs to break the dependence of the models on the choice of the coordinate frame of the camera i.e. expression models should generalize across facial poses. To perform this systematically, one needs to understand the space of observed images subject to projective transformations. However, since the projective shape-space is cumbersome to work with, we address this problem by deriving models for expressions on the affine shape-space as an approximation to the projective shape-space by using a Riemannian interpretation of deformations that facial expressions cause on different parts of the face. We use landmark configurations to represent facial deformations and exploit the fact that the affine shape-space can be studied using the Grassmann manifold. This representation enables us to perform various expression analysis and recognition algorithms without the need for the normalization as a preprocessing step. We extend some of the available approaches for expression analysis to the Grassmann manifold and experimentally show promising results, paving the way for a more general theory of view-invariant expression analysis.**

## I. Introduction

The goal of facial expression analysis is to create systems that can automatically analyze and recognize facial feature changes and facial motion from visual information. This has been an active research topic for several years and has attracted the interest of many computer vision researchers and behavioral scientists, with applications in behavioral science, security, animation, and human-computer interaction [1].

Facial expressions occur along with the head motions and pose variations, especially when there are spontaneous human-to-human interactions. Therefore, it is necessary for facial expression analysis algorithms to be able to jointly analyze the head pose and facial expressions, or in other words be invariant to pose changes. But this is a challenging task especially due to large variations in the appearance of facial expressions in different views and also the nonlinear coupling of these different sources of variations in the observed images.

While most of the proposed methods for facial expression analysis can only handle frontal-view faces, [2]–[4], there has been recent progress in designing pose-invariant facial expression recognition algorithms. Previous work treating pose invariance in facial expression analysis can be generally divided into two groups of approaches as those based on a 3D face model and those based on a 2D face model. It should be noted that since facial geometry conveys important information about a human's emotional state, one of the common approaches for analyzing facial expressions is by using face shape models. Therefore, our focus in this paper is on the geometry-based approaches.

There are several approaches that use a 3D face model and jointly estimate the rigid and nonrigid facial deformations [5]–[9]. In these approaches, the estimated rigid motion of the face is a byproduct of the system and can be used for tracking facial landmarks, while the non-rigid motion is further processed for expression analysis. The main disadvantages of these 3D shape model-based approaches is that they are computationally expensive, they require time-consuming initialization process, and the 3D model fitting techniques may not converge. Moreover in a HCI application, 2D images and 2D shapes are far more easily available than 3D shapes. Thus, the focus of our work is on using 2D facial geometries.

Since 2D face images are projections of 3D faces, the rigid head motions and non-rigid facial expressions are nonlinearly coupled in the captured 2D images. This fact has made the pose-invariant facial expression analysis based on 2D shape models hard to solve [10]. Most of the available approaches that use a 2D shape model and facial landmarks, decouple the rigid and nonrigid motions via normalization by aligning all the available configurations to a reference frame [11]–[13]. But these normalization-based approaches depend on the choice of the reference frame which is usually made arbitrarily [14]. There is also a very recent normalization-based algorithm proposed by Rudovic *et al.* [10] in which using some trained regression functions, the 2D landmark locations in non-frontal poses are mapped to the corresponding locations in the frontal pose. This method shows promising results for pose-invariant expression recognition, however, it requires a pose estimation phase before performing pose normalization and errors in pose estimation may contribute to recognition errors.

The main drawback of all these normalization-based approaches is that they ignore the intrinsic geometry of shape-space and instead they consider the aligned shapes as points

This project is partially supported by a Grant N00014-09-1-016 4 from the Office of Naval Research.

in the Euclidean space. In this paper, we emphasize the importance of understanding shapes as equivalence classes across view-changes instead of as a vector derived from features such as active shape models. This would enable expression models to generalize across views. Since the 2D face images are the projections of 3D faces, the projective shape-space, which carries information about the configuration of the facial landmarks that are invariant to the camera view point, is of most importance in expression analysis.

Equivalence classes are difficult to work with from a statistical perspective and we need a canonical representative from them that can be used for statistical analysis. For the similarity shape-space (Kendall's shape-space) [15], concepts such as pre-shape and Procrustes analysis are well-studied. The affine shape-space for $m$ landmark points in $\mathbb{R}^k$ is identified with the set of all $k$-dimensional subspaces of $\mathbb{R}^m$, [14], [16], which is a Grassmann manifold. This manifold also has well-studied mathematical structure that can be used for statistical analysis [17]–[20]. But similar advances in projective shape-space have been slow due to overemphasis on the importance of similarity shapes in image analysis [21]. Thus, suitable metrics are hard to define for comparing projective shapes.

On the other hand, projective transformations can in many cases be approximated by affine transformations. Therefore, here we perform expression analysis using the affine shape-space since its structure is well-understood. But the eventual goal is to achieve invariance to large view-changes via projective shape-spaces and this work is a small step in that direction so that the advantages of using shape spaces for pose-invariant expression analysis can be realized.

In section II, we discuss the landmark-based representation of facial geometry. We then discuss the affine shape-space and show that the facial landmark configurations can be identified as points on the Grassmann manifold. Some mathematical discussions on the Grassmann manifold are also provided. Then in section III we describe the extension of some of the facial expression analysis algorithms to this shape-space and present experimental results. Section IV concludes the paper.

**Contribution:** Our main contribution is to show the advantages of using a proper shape-space for pose-invariant facial expression analysis. Modeling the facial landmark configurations as equivalence classes on the affine shape-space, as an approximation to the projective shape-space, not only decouples the rigid and nonrigid facial motions, but also offers a well-defined underlying structure for the data. Most of the available algorithms for expression analysis can easily be extended to this shape-space.

## II. Facial Expressions on the Manifold

Non-rigid facial deformations can be encoded using facial action coding system (FACS), introduced by Ekman *et al.* [22], where each action unit (AU) determines the shape of its corresponding facial components. Figure 1 (second row) shows the face of a subject with different AUs. As it can be seen, while AU-1 implies a raised inner brow, AU-27 corresponds to a wide open mouth. As the figure shows, the geometry of facial components is a good cue for representing and recognizing most of the AUs/expressions. In our work we use the landmarks on the face to represent the facial geometry at each frame of an expression sequence.

### A. Facial Geometry

Landmark-based face shape representation is one of the most widely used approaches for geometric modeling of faces. Here we model the facial geometry using a $m \times 2$ matrix $L = [(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)]^T$ in $\mathbb{R}^2$. Figure 1 shows the locations of 2D landmarks on the faces in different databases. To model the non-rigid deformations corresponding to each expression, the first step is to decouple the rigid and non-rigid deformations of the landmarks. Since the shape observed in an image of a face is a perspective projection of the 3D locations of the landmarks, projective shape-space is an appropriate choice to realize invariance with regard to the camera angle. Modeling the facial geometry as equivalence classes in the projective shape-space introduces a new way of statistical analysis of 2D faces which is independent of the face poses. But advances in statistical analysis of projective shapes are still preliminary. On the other hand, projective shapes in constrained situations can be approximated with affine shapes. Therefore, we limit our discussions to affine shape-spaces.

All the affine transformations of a shape can be derived from the base shape simply by multiplying the centered shape matrix, $\tilde{L}_{\text{base}}$, by a $2 \times 2$ full-rank matrix on the right (translations are removed by centering). Multiplication by a full-rank matrix on the right preserves the column space of the matrix, thus the 2D subspace of $\mathbb{R}^m$ spanned by the columns of the centered shape, i.e. $\text{span}(\tilde{L}_{\text{base}})$, is invariant to affine transformations of the shape. Subspaces such as these can be identified as points on a Grassmann manifold [17].

Given a sequence of a face performing an expression, we would like to model the facial deformations that generate such a sequence. Based on the above discussions, a sequence of faces is represented as a sequence of points on a Grassmann manifold. So, we can model the facial deformations via geodesics on the manifold, where a geodesic is a path of shortest length on the manifold between two given points. The geodesic emanating from a point on the manifold can be characterized by a velocity vector on the tangent plane at that point. Therefore, we parametrize the facial deformations corresponding to each expression/AU as a velocity (or sequence of velocities) with which a point on the manifold (neutral face) should move in order to reach the final point (apex) in unit-time. In the following, we briefly describe how to compute these parameters on the Grassmann manifold. The readers are referred to [19], [20] for a more in-depth discussion of the mathematical details.

### B. Geometry of the Grassmann Manifold

A Grassmann manifold, $\mathcal{G}_{m,k}$, is the space of $k$-dimensional subspaces in $\mathbb{R}^m$. By fixing $m, k$ throughout

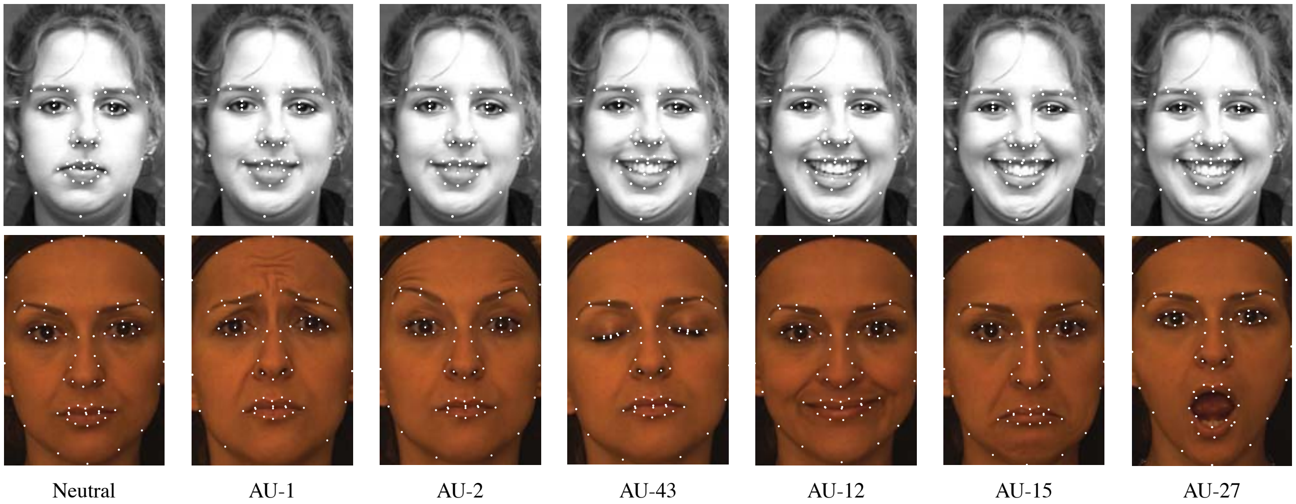


Fig. 1. A sequence from the Cohn-Kandade database (first row), and a subject in the Bosphorus database performing various AUs (second row). The landmark locations are shown on the faces.

the paper we avoid adding suffixes to index the set $\mathcal{G}$. Each element of $\mathcal{G}$ can be identified by a unique $m \times m$ projection matrix, $P$, onto the $k$-dimensional subspace of $\mathbb{R}^m$ [1]. Let $\mathbb{P}$ be the set of all $m \times m$ symmetric, idempotent matrices of rank $k$. Then, $\mathbb{P}$ is the set of all projection matrices and hence is diffeomorphic to $\mathcal{G}$. The identity element of $\mathbb{P}$ is defined as $Q = diag(I_k, 0_{m-k,m-k})$, where $0_{a,b}$ is an $a \times b$ matrix of zeros and $I_k$ is the $k \times k$ identity matrix.

A Grassmann manifold $\mathcal{G}$ (or $\mathbb{P}$) is a quotient space of the orthogonal Lie group, $O(m)$. Therefore, the geodesic on this manifold can be made explicit by lifting it to a particular geodesic in $O(m)$ [20]. This process is illustrated by Fig 2. Then the tangent, $X$, to the lifted geodesic curve in $O(m)$ defines the velocity associated with a curve on $\mathbb{P}$. The tangent space of $O(m)$ at identity is $o(m)$, the space of $m \times m$ skew-symmetric matrices, $X$. Moreover, in $o(m)$ the Riemannian metric is just the inner product $\langle X_1, X_2 \rangle = \text{trace}(X_1 X_2^T)$. This property is inherited by $\mathbb{P}$ as well.

The geodesics in $\mathbb{P}$ passing through the point $Q$ (at time t=0) are of the type $\alpha : (-\epsilon, \epsilon) \mapsto \mathbb{P}$, $\alpha(t) = \exp(tX)Q\exp(-tX)$, where $X$ is a skew-symmetric matrix belonging to the set $M$, where

$$M = \left\{ \begin{bmatrix} 0 & A \\ -A^T & 0 \end{bmatrix} : A \in \mathbb{R}^{k,n-k} \right\} \subset o(m) \tag{1}$$

Therefore, the geodesic between $Q$ and any $P$ is completely specified by an $X \in M$ such that $\exp(X)Q\exp(-X) = P$. We can then construct a geodesic between any two $P_1, P_2 \in \mathbb{P}$ by rotating them to $Q$ and some $P \in \mathbb{P}$.

One important concept is the parallel transport which is a smooth operation between tangent spaces that allows us to transfer tangent vectors between points while locally

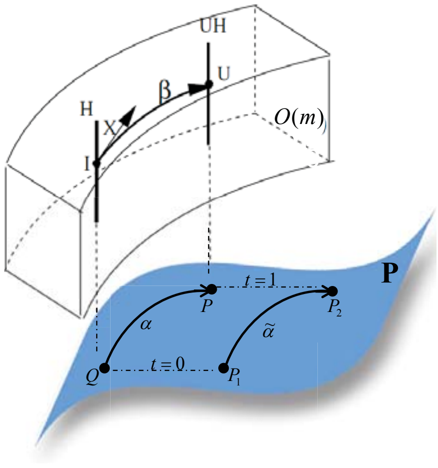


Fig. 2. Process of computing a geodesic on the Grassmann manifold by lifting it to the particular geodesic in $O(m)$, [20].

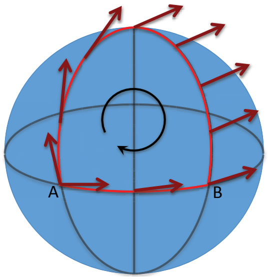


Fig. 3. Parallel transport of a vector around a closed loop on the manifold. The direction and orientation of the vector changes to match the local structure of the destination point.

[1] There are two approaches for representing points on the Grassmann manifold, either as tall-thin $m \times k$ matrices, or as square idempotent projection matrices. The former while more efficient, requires involved quotient-space interpretations. The projection matrix representation, on the other hand, has relatively simpler analytical and geometric properties, but it is computationally intensive. However, since we are using sparse landmarks, $m$ is typically in the order of $50 - 100$, thus the extra computational burden is not very significant. Therefore, we work with the projection matrix representation for the Grassmann points.

preserving direction and orientation [19]. In Euclidean space, the parallel transport is simply performed by moving the base of the arrow. However, moving a tangent vector by this technique on a manifold will not generally be a tangent vector. Figure 3 illustrates the parallel transport on the manifold. As the figure shows the result of parallel transport depends on the path along which we move the tangent vector. Readers are referred to [19] for more details on parallel transport on the Grassmann manifold. Some Grassmann related algorithms which will be of use in expression analysis are provided in the appendix.

## III. Facial Expression Analysis

In this section the goal is to perform expression analysis using the equivalence classes of face shapes on the Grassmann manifold. We show how we can extend most of the available expression analysis algorithms to the Grassmann manifold in order to perform expression analysis in an affine-invariant manner. In particular, we discuss the linear modeling of facial landmark deformations using ASM as well as modeling the nonlinear deformations using a nonlinear dimensionality reduction technique. We also discuss the AU and basic expression recognition by learning statistical models on the Grassmann manifold.

We use three databases to evaluate the strength of this approach. The first one is the Bosphorus database [23] that is composed of a selected subset of AUs as well as the six basic emotions for 105 subjects. For each subject, the neutral face and the face in the apex of various AUs and emotions are presented. In addition, 22 landmarks per face are provided by the database. However, we manually marked 75 landmarks for some of the subjects (Fig. 1). The second database is the Cohn-Kanade's DFAT-504 database (CK) [24], which consists of more than 100 subjects, each performing a set of emotions. The sequences begin from neutral or nearly neutral faces and end at the apex state of the expression. The sequences were annotated by certified FACS coders. We also manually labelled the sequences into the six basic expressions. Moreover, 59 landmarks per faces are available [25]. The third database is a sequence of a talking face [2] with 5000 frames which shows the face of a subject while talking. The facial landmarks are also provided for this database. Figure 1 shows some examples of the images in these databases.

### A. Facial deformations modeling

By representing the facial landmark configurations as equivalence classes on the affine shape-space, we transform the data to the space of nonrigid deformations. In other words, starting from a projection matrix on the Grassmann manifold corresponding to a neutral face, it is ensured that moving in each direction on the manifold is corresponding to a nonrigid deformation of the initial configuration. These nonrigid facial deformations can be statistically modeled, depending on the linear or nonlinear assumptions for the deformations, by calculating the principal directions of variations using principal geodesic analysis (PGA) or by estimating the expression manifold through nonlinear dimensionality reduction.

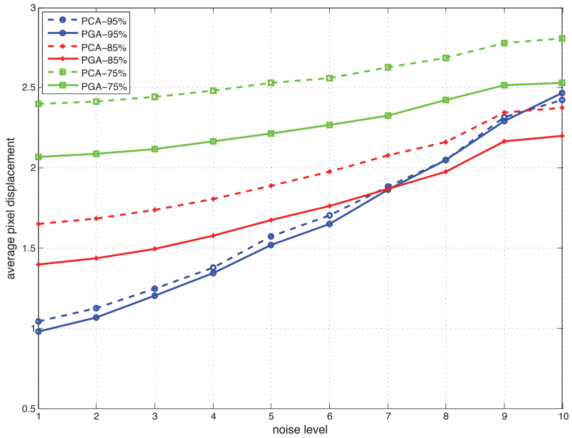


Fig. 4. PCA versus PGA bases for facial landmarks representation and reconstruction.

*1) Linear deformations modeling:* As we know, assuming a linear structure for the facial deformations, active shape models (ASM) learn the principal directions of facial geometric variations using PCA. The same idea can be extended to the Grassmann manifold using PGA, [26], which is a generalization of PCA to the manifold setting. Representing the facial landmarks with different expressions as points on the Grassmann manifold, the principal directions of variations can be learned. For this purpose, the first step is to find the intrinsic mean of the points on the manifold using the Karcher mean algorithm for the Grassmann manifold [20], [27]. Then the principal geodesic directions are calculated using the warped data to the tangent plane at the mean point.

Figure 4 compares the PCA and PGA approaches for recovering the face shape, in the talking face database, under the noise. Using manually marked 2D landmarks on the faces in this database as ground truth, we perturb the position of landmarks independently with different levels of Gaussian noise. Then we use these two approaches to reconstruct the shapes from the noisy observations. We apply the similarity alignment method on the faces to bring them to the same coordinate frame before performing PCA. As the figure shows the PGA bases have higher resilience against noise. This can be due to the fact that modeling the variations on the shape-space ensures that the variability being computed is from shape changes only and not due to rigid transformations.

*2) Nonlinear manifold learning:* Since the linear assumption for facial deformations is not always valid, there are several approaches that consider the geometric variations of facial components on a low-dimensional nonlinear manifold and learn such a manifold using nonlinear dimensionality reduction algorithms [7], [12], [13]. The nonlinear dimensionality reduction algorithms preserve the local structure of the data while reducing dimensionality. Therefore, considering the true structure of the data is important for these algorithms.

[2] http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data

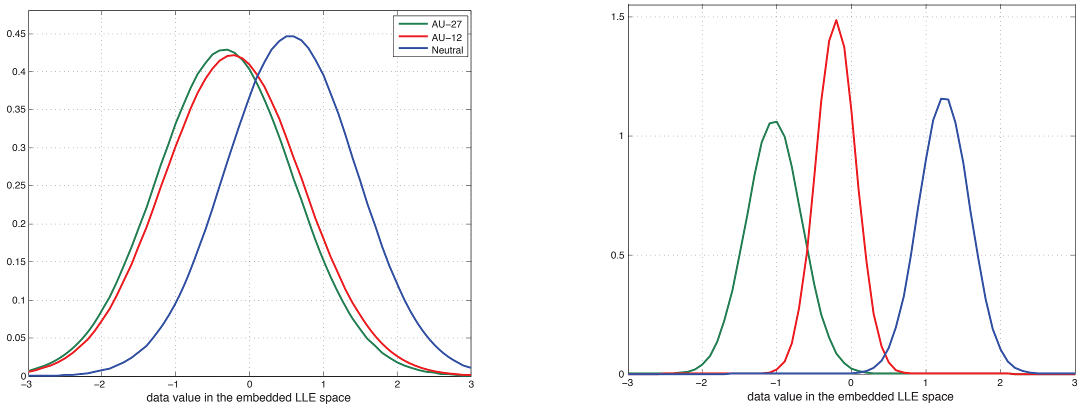


Fig. 5. Dimensionality reduction using LLE from data in the Euclidean space (left) and on the Grassmann manifold (right). Plots show the distributions of the 1D projected data separated by the classes. Better separation is seen among classes on the right.

As we emphasized earlier, the face shapes after quotienting the affine group lie on the Grassmann manifold. The local structure of the data on this space can be employed to learn the low-dimensional nonlinear expression manifold. A simple dimensionality reduction algorithm is locally linear embedding (LLE), [28], which is a neighborhood-preserving embedding of high-dimensional inputs. This algorithm can be extended to the data on the Grassmann manifold to nonlinearly project the subspaces to the lower dimensional space.

We compare the results of low-dimensional manifold learning using the data on the Grassmann manifold versus normalized data in the Euclidean space. For this purpose the dimensionality of the training data, composed of the facial landmarks for 80 subjects in the Bosphorus database having three different expressions, Neutral, AU-12, and AU-27, is reduced to one-dimension using the LLE method. Figure 5 illustrates the distribution of the data in the LLE embedded space for both Euclidean and manifold cases. As the figure clearly shows, for the data on the Grassmann manifold the one-dimensional representations are well-separated for different classes compared to that for the data on the Euclidean space.

### B. AUs template learning

Facial expressions are combinations of several AUs occurring simultaneously or sequentially in different parts of the face. Recognizing these AUs is a proper way for expression recognition. To this end, we learn a template for each AU on the Grassmann manifold. A sequence of faces performing an AU is represented as a sequence of facial landmarks $\mathcal{L} = \{L_i\}_1^n$ where $L_1$ corresponds to the neutral face and $L_n$ to the apex point (in our cases). This sequence is equivalent to a sequence of subspaces/projection matrices $\mathcal{P} = \{P_i\}_1^n$ which can be considered as samples of a curve on the Grassmann manifold (Fig. 6).

We represent each sequence as a piecewise-geodesic curve and model each piece using its velocity vector. Therefore, we have a sequence of $N$ velocity vectors $\mathcal{A} = \{A_i\}$, where $A_i = \text{velocity}(P_i \rightarrow P_{i+1})$ and $N$ is the number of segments. For the case of CK database, we choose $N$ to be equal to the number of sequence frames minus one. But for the Bosphorus database, since only the initial and final images of each sequence are available, each AU is represented using the velocity vector corresponding to the geodesic between $P_1$ and $P_n$ $(n = 2)$. Although this geodesic is an approximation to the real sequence, our experiments show that it is a good approximation since AUs are simple and represented by short sequences and the geodesic between the initial and end point is almost the same as the curve connecting the intermediate frames on the manifold.

In order to learn a template model for each AU on the manifold, an important step is to parallel transport each curve to a common tangent plane so that we can learn the statistics of the set of vectors corresponding to an AU. As we mentioned earlier, parallel transport on the manifold is different from that of Euclidean space. Figure 7 compares the results of parallel transport on the Euclidean space and the manifold. The tangent vector from the sequence in the first row is learned and then we parallel transport it to a new face in the second and third rows. Before applying the tangent vector, we affine transform the new face so that we can better show the effect of parallel transport. As the figure shows, parallel transport on the manifold generates the face with the correct deformations while the corresponding result on the Euclidean space is distorted. This again emphasizes the importance of exploiting the shape-space geometry for our problem.

The natural choice for the common tangent plane is the one at the average of neutral faces on the manifold. For this purpose, we calculate the Karcher mean for the neutral faces. Then the velocity vectors corresponding to each AU, for the Bosphorus database, is parallel transported to the mean point and the Gaussian distribution is learned as the model for each AU. It should be noted that in the Bosphorus database each AU is represented using a tangent vector. Therefore, the final model is the mean and standard deviation over the tangent vectors at the mean neutral face.

Another method for learning an AU template model is using dynamic time warping (DTW) on various sequences

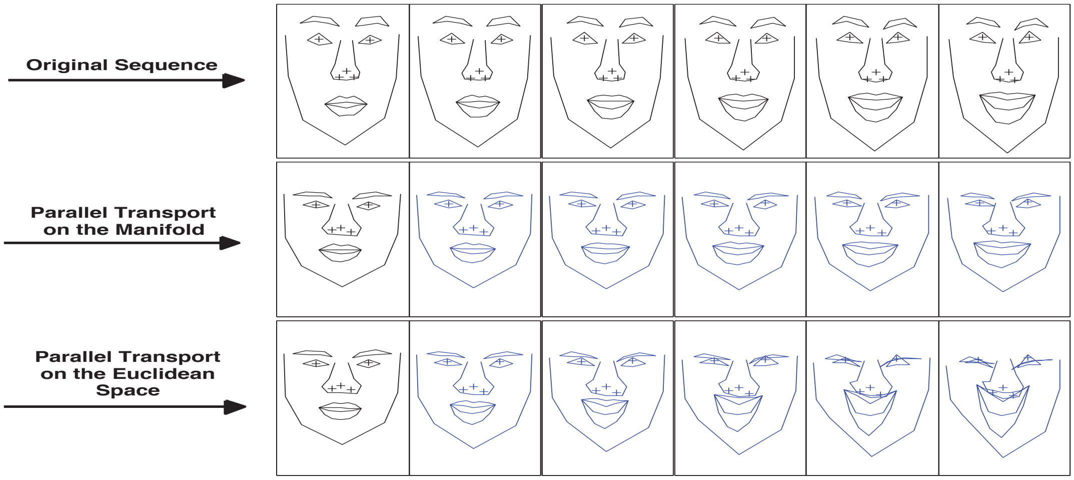

Original Sequence
Parallel Transport on the Manifold
Parallel Transport on the Euclidean Space

Fig. 7. Comparison between the results of parallel transport on the manifold versus that of Euclidean space. The first sequence (its tangent vector from the leftmost to the rightmost shape) is parallel transported to the face on the second and third row, and the new sequence is synthesized on the manifold (second row) and Euclidean space (third row).

corresponding to each AU [29]. Especially for the CK database, since each AU is represented using a sequence of projection matrices and since expressions occur at different rates, it is necessary to time warp the sequences in order to learn a rate-invariant model for them. Adapting the DTW algorithm to the sequences that reside on a Riemannian manifold is a straightforward task, since DTW can operate with any measure of similarity between the different temporal features. Here, we use the geodesic distance between the projection matrices of different sequences as a distance function and warp all the sequences (corresponding to an AU) to a randomly selected sequence. Then the final model for each AU is obtained by computing the Karcher mean of all the warped sequences. This is a simple and fast approach that works fairly well.

*1) Action Units Recognition:* Using the learned AU models for the Bosphorus and CK databases, we perform AU recognition. We report the results on seventeen single AUs in the Bosphorus database and nine single or combined AUs in the CK database. The training samples are chosen as images/sequences containing only the target AU occurring in the corresponding local facial components (brow, eye, nose, and mouth). In the Bosphorus database the lack of sufficient landmarks on the faces limits our recognition capabilities. For example we cannot recognize the AU-43 since no landmarks are provided for the eyes. Also for the CK database, since the sequences are mainly corresponding to the expressions and not AUs, we only chose those AUs for which enough training sequences are available. We divide both databases into eight sections, each of which contains images from different subjects. Each time, we use seven sections for training and the remaining sections for testing so that the training and testing sets are mutually exclusive. The average recognition performance is computed on all the sections.

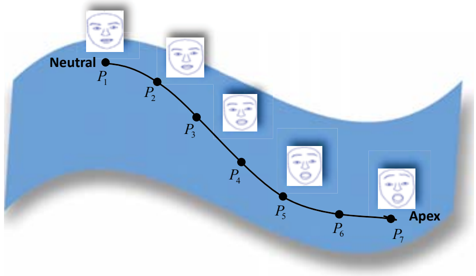
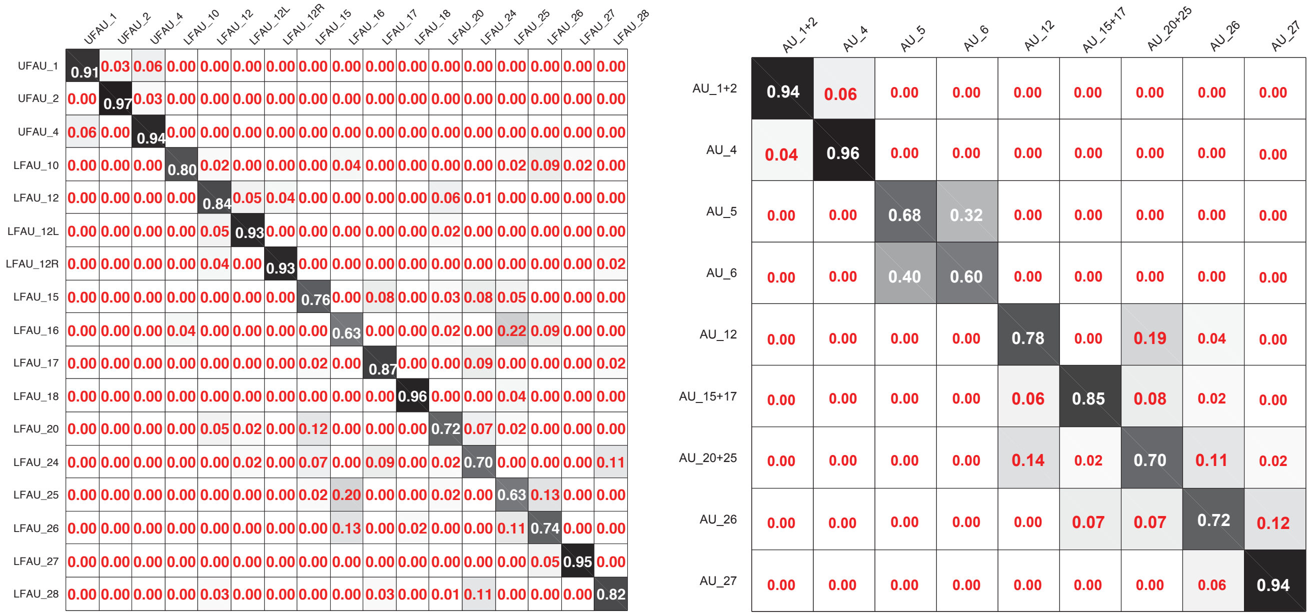


Fig. 6. A sequence of facial expression is a curve on the Grassmann manifold.

For the Bosphorus database, we perform maximum likelihood (ML) recognition where we find the probability of each test velocity vector comes from the learned Gaussian distribution. But for the CK database, we first warp each test sequence to the learned template using DTW and then use the distance between the two sequences for recognition. Figure 8 shows the confusion matrices for both databases. As the results indicate, for the AUs that are mainly identified by their facial deformations the recognition rate is high, e.g. AU-2, AU-4, and AU-27. However, for AUs whose distinction is more due to the appearance deformations than the geometries, the algorithm may confuse them with the AUs with similar geometries, e.g. AU-16 and AU-25. In these cases, AUs occurring in other parts of the face can be used as cues to remove the ambiguity and improve the recognition.

We also performed a recognition experiment using the Bosphorus database on the Euclidean space, where the normal parallel transport is performed before learning the distributions. While the average recognition rate for AUs on the Grassmann manifold is 83%, this value is 79% on the Euclidean space. Although the recognition rate is improved on the Grassmannian, but it is not considerable. A possible reason can be the fact that in the Bosphorus database the faces are almost always frontal and there are not significant



|  | UFAU_1 | UFAU_2 | UFAU_4 | LFAU_10 | LFAU_12 | LFAU_12L | LFAU_12R | LFAU_15 | LFAU_16 | LFAU_17 | LFAU_18 | LFAU_20 | LFAU_24 | LFAU_25 | LFAU_26 | LFAU_27 | LFAU_28 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UFAU_1 | 0.91 | 0.03 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| UFAU_2 | 0.00 | 0.97 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| UFAU_4 | 0.06 | 0.00 | 0.94 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| LFAU_10 | 0.00 | 0.00 | 0.00 | 0.80 | 0.02 | 0.00 | 0.00 | 0.00 | 0.04 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.09 | 0.02 | 0.00 |
| LFAU_12 | 0.00 | 0.00 | 0.00 | 0.00 | 0.84 | 0.05 | 0.04 | 0.00 | 0.00 | 0.00 | 0.00 | 0.06 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 |
| LFAU_12L | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 | 0.93 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| LFAU_12R | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 | 0.00 | 0.93 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 |
| LFAU_15 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.76 | 0.00 | 0.08 | 0.00 | 0.03 | 0.08 | 0.05 | 0.00 | 0.00 | 0.00 |
| LFAU_16 | 0.00 | 0.00 | 0.00 | 0.04 | 0.00 | 0.00 | 0.00 | 0.00 | 0.63 | 0.00 | 0.00 | 0.02 | 0.00 | 0.22 | 0.09 | 0.00 | 0.00 |
| LFAU_17 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.87 | 0.00 | 0.00 | 0.09 | 0.00 | 0.00 | 0.00 | 0.02 |
| LFAU_18 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.96 | 0.00 | 0.00 | 0.04 | 0.00 | 0.00 | 0.00 |
| LFAU_20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 | 0.02 | 0.00 | 0.12 | 0.00 | 0.00 | 0.00 | 0.72 | 0.07 | 0.02 | 0.00 | 0.00 | 0.00 |
| LFAU_24 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.07 | 0.00 | 0.09 | 0.00 | 0.02 | 0.70 | 0.00 | 0.00 | 0.00 | 0.11 |
| LFAU_25 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.20 | 0.00 | 0.00 | 0.02 | 0.00 | 0.63 | 0.13 | 0.00 | 0.00 |
| LFAU_26 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.13 | 0.00 | 0.02 | 0.00 | 0.00 | 0.11 | 0.74 | 0.00 | 0.00 |
| LFAU_27 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 | 0.95 | 0.00 |
| LFAU_28 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.00 | 0.01 | 0.11 | 0.00 | 0.00 | 0.00 | 0.82 |

|  | AU_1+2 | AU_4 | AU_5 | AU_6 | AU_12 | AU_15+17 | AU_20+25 | AU_26 | AU_27 |
|---|---|---|---|---|---|---|---|---|---|
| AU_1+2 | 0.94 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| AU_4 | 0.04 | 0.96 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| AU_5 | 0.00 | 0.00 | 0.68 | 0.32 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| AU_6 | 0.00 | 0.00 | 0.40 | 0.60 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| AU_12 | 0.00 | 0.00 | 0.00 | 0.00 | 0.78 | 0.00 | 0.19 | 0.04 | 0.00 |
| AU_15+17 | 0.00 | 0.00 | 0.00 | 0.00 | 0.06 | 0.85 | 0.08 | 0.02 | 0.00 |
| AU_20+25 | 0.00 | 0.00 | 0.00 | 0.00 | 0.14 | 0.02 | 0.70 | 0.11 | 0.02 |
| AU_26 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.07 | 0.07 | 0.72 | 0.12 |
| AU_27 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.06 | 0.94 |

Fig. 8. Confusion matrices for AU recognition on the Bosphorus (left) and CK (right) databases. LFAU and UFAU refer to the lower and upper face AUs.

affine variations between the faces.

### *C. Basic Expression Recognition*

To have a comparison with baseline results, we perform the basic expression recognition using the CK database. For this purpose, we manually label the sequences into one of six basic expressions, namely, {Happy, Sad, Fear, Surprised, Disgust, Angry} and then from each sequence we select the last five frames. We apply the linear discriminant analysis (LDA) and multi-class SVM to the training data and perform leave-one-subject-out cross-validation over 89 subjects. Table I compares the results of applying LDA and SVM to the normalized data in the Euclidean space as well as subspace representations on the Grassmann manifold. For the Grassmann data, we use the projection Grassmann kernel, [30], to perform SVM as well as kernel LDA. The table also shows the latest results reported by Cohn and Kanade *et al.* [31]. However, it should be noted that these results are reported on the extended Cohn-Kanade dataset (CK+) which has more subjects and accurate expression coding. The results show some improvements in using the geometry of the Grassmannian over performing the analysis in the Euclidean space. Although we see only modest improvements, but as discussed before, while performing normalization in the Euclidean space suffers from being arbitrary and thus is highly sensitive to noise, the analysis on the Grassmann is more stable and resilient to noise.

TABLE I
BASIC EXPRESSION RECOGNITION ON THE CK DATASET USING ALGORITHMS ON BOTH EUCLIDEAN (E) AND GRASSMANN (G) SPACES. THE LAST ROW SHOWS THE RESULTS ON CK+ DATASET.

|  | Ha | Sa | Fe | Su | Di | An | Averaged |
|---|---|---|---|---|---|---|---|
| E-LDA | 91.3 | 75.0 | 70.2 | 96.1 | 76.3 | 60.0 | 78.15 |
| G-LDA | 88.9 | 78.2 | 74.4 | 97.3 | 80.5 | 68.0 | 81.2 |
| G-KLDA | 95.1 | 85.7 | 83.0 | 98.6 | 86.8 | 65.7 | 85.8 |
| E-SVM | 91.3 | 80.3 | 74.4 | 97.2 | 78.9 | 62.8 | 80.8 |
| G-SVM | 95.0 | 85.7 | 74.5 | 97.2 | 78.9 | 65.7 | 82.8 |
| SVM [31] | 98.4 | 4.0 | 21.7 | 100.0 | 68.4 | 35.0 | 54.6 |

## IV. CONCLUSION

This paper is a step toward breaking the dependence of facial expression analysis systems to the choice of the coordinate frame of the camera. We discussed that using the equivalence class of shapes in a proper shape-space, one can remove the need for a pre-processing step to align the data to a common coordinate frame. While we claim that the projective shape-space is the proper space to model the facial variations, we have limited our discussions to the affine shape-space since it is mathematically well understood compared to the projective space. We showed that the affine shape-space for our facial landmark configurations has Grassmannian properties and therefore nonrigid facial deformations due to various expressions can be represented as points on the Grassmann manifold. By modeling the facial expressions on this manifold we ensure that the variability being computed is from shape changes only and not the coordinate frame. We extended some of the available statistical algorithms for facial expressions, e.g. ASM, nonlinear manifold learning, and expression template learning, to the Grassmann manifold and showed the benefits of this representation. It should be noted that while similarity alignment in the Euclidean space can remove the effect of camera coordinate frame to a good extent, working with equivalence class of shapes in the shapes-spaces is a systematic way of dealing with alignment issue and the main benefits become more obvious when we move to the projective shape-space.

## Appendix

Here we present the solutions to some problems related to traversing the Grassmann manifold which will be of use in expression analysis.

**P1:** *Find the geodesic between two points $P_1$, $P_2 \in \mathbb{P}$.*

```
1. Let $U \in \Phi^{-1}(P_1)$ so that $P_1 = UQU^T$
2. Define $P = U^T P_2 U$
3. Find $X$ that takes $Q$ to $P$.(using P2)
4. Find the geodesic between $Q$ and $P$:
       $\alpha(t) = \exp(tX)Q\exp(-tX)$
5. Shift $\alpha(t)$ to $P_1$ and $P_2$ as:
       $\tilde{\alpha}(t) = (U\exp(tX)U^T)P_1(U\exp(-tX)U^T)$
*** Here the sub-matrix $A$, where $X = \text{cdiag}(A, -A^T)$, is
    the velocity that takes $P_1$ to $P_2$ in unit time.
```

**P2:** *Given $P \in \mathbb{P}$, find an $X \in M$ such that $\alpha(1) = \exp(X)Q\exp(-X) = P$*

```
1. Define $B = Q - P$
2. Find the eigen decomposition of $B = W\Sigma W^T$.
*** The eigenvalues of $B$ are either 0's or occur in pairs
    of the form $(\lambda_j, -\lambda_j)$ where $0 < \lambda_j \leq 1$.
    Then $Qw_j$ and $Qw_{j'}$ are chosen to be positive
    multiples of each other, where $w_j$, $w_{j'}$ are the
    columns of $W$ corresponding to the eigenvalues
    $\lambda_j$ and $-\lambda_j$. This is achieved by multiplying
    $w_j$ by an appropriate unit number.
3. Set $X = W\Omega W^T \in M$, where $\Omega$ is derived from $\Sigma$
   by replacing all the $2 \times 2$ blocks, diag$(\lambda_j, \lambda_{j'})$, by
   cdiag$(-\sin^{-1}(\lambda_j), \sin^{-1}(\lambda_j))$ and keep the rest
   unchanged.
```

**P3:** *Find $P_2 \in \mathbb{P}$ that is reached in unit time by following a geodesic starting at $P_1$ with velocity $A$.*

```
1. Let $U \in \Phi^{-1}(P_1)$ so that $P_1 = UQU^T$
2. Form a skew-symmetric matrix $X = \text{cdiag}(A, -A^T)$
3. Define $V(t) = U\exp(tX)U^T$
4. Then $P_2 = V(1)P_1V(1)^T$
```

**P4:** *Given $P_1$ and the direction vector $X \in M$, find the parallel transport of $X$ to point $P_3$.*

```
1. Let $U_1 \in \Phi^{-1}(P_1)$ so that $P_1 = U_1QU_1^T$
1. Let $U_3 \in \Phi^{-1}(P_3)$ so that $P_3 = U_3QU_3^T$
2. Define $V = U_1 exp(X) U_1^T$
3. Compute $P_4 = VP_3V^T$
4. Having $P_3$ and $P_4$, the parallel transport of $X$ to $P_3$
   is calculated using P1
```

## References

[1] A. K. Jain and S. Z. Li, *Handbook of Face Recognition*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[2] T. Simon, N. Minh, F. D. la Torre, and J. Cohn, "Action unit detection with segment-based SVMs," in *CVPR*, 2010.

[3] P. Yang, Q. Liu, and M. Dimitris, "Exploring facial expression with compositional features," in *CVPR*, 2010.

[4] Y.-L. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *TPAMI*, vol. 23, pp. 97–115, 1999.

[5] Z. Zhu and Q. Ji, "Robust real-time face pose and facial expression recovery," *CVPR*, vol. 1, pp. 681–688, 2006.

[6] Y. Tong, J. Chen, and Q. Ji, "A unified probabilistic framework for spontaneous facial action modeling and understanding," *TPAMI*, vol. 32, no. 2, pp. 258–274, 2010.

[7] W.-K. Liao and G. Medioni, "3D face tracking and expression inference from a 2D sequence using manifold learning," in *CVPR*, 2008.

[8] Y. Chang, M. Vieira, M. Turk, and L. Velho, "Automatic 3D facial expression analysis in videos," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, October 2005.

[9] T. Wang and J. Lien, "Facial expression recognition system based on rigid and non-rigid motion separation and 3D pose estimation," in *Pattern Recognition*, vol. 42, 2009, pp. 962–977.

[10] O. Rudovic, I. Patras, and M. Pantic, "Coupled gauusian process regression for pose-invariant facial expression recognition," in *ECCV*, 2010, pp. 350–363.

[11] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, January 1995.

[12] Y. Chang, C. Hu, R. Feris, and M. Turk, "Manifold based analysis of facial expression," *Image and Vision Computing*, vol. 24, no. 6, pp. 605–614, 2006.

[13] C.-S. Lee and A. M. Elgammal, "Nonlinear shape and appearance models for facial expression analysis and synthesis," in *ICPR06*, 2006, pp. II: 497–502.

[14] E. Begelfor and M. Werman, "Affine invariance revisited," in *CVPR*, 2006, pp. 2087–2094.

[15] D. G. Kendall, "Shape manifolds, procrustean metrics, and complex projective spaces," *Bull. London Math. Soc.*, vol. 16, pp. 18–121, 1984.

[16] A. Bhattacharya and R. Bhattacharya, "Nonparametric statistics on manifolds with applications to shape spaces," in *IMS Collections*, 2008, vol. 3, pp. 282–301.

[17] P. Turaga, A. Veeraraghavan, and R. Chellappa., "Statistical analysis on stiefel and grassmann manifolds with applications in computer vision," in *CVPR*, 2008.

[18] Y. Chikuse, *Statistics on Special Manifolds*. Springer, 2003.

[19] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl*, vol. 20, pp. 303–353, 1998.

[20] A. Srivastava and E. Klassen, "Bayesian and geometric subspace tracking," *Advances in Applied Probability*, vol. 36, no. 1, pp. 43–56, 2004.

[21] V. Patrangenaru, X. Liu, and S. Sugathadasa, "A nonparametric approach to 3D shape analysis from digital camera image - I," *Journal of Multivariate Analysis*, vol. 101, pp. 11–31, January 2010.

[22] P. Ekman and W. Friesen, *The facial action coding system: a technique for the measurement of facial movement*. Consulting Psychologists Press., 1978.

[23] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus database for 3D face analysis," in *Workshop on Biometrics and Identity Management (BIOID)*, 2008.

[24] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *FGR*, 2000, pp. 46–53.

[25] G. Lipori, "Manual annotations of facial fiducial points on the Cohn-Kanade database," lAIV laboratory, University of Milan, web url: http://lipori.dsi.unimi.it/download.html.

[26] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 995–1005, 2004.

[27] X. Pennec, "Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements," *Journal of Mathematical Imaging and Vision*, vol. 25, no. 1, pp. 127–154, 2006.

[28] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *SCIENCE*, vol. 290, pp. 2323–2326, 2000.

[29] A. Veeraraghavan, A. Srivastava, A. K. R. Chowdhury, and R. Chellappa, "Rate-invariant recognition of humans and their activities," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1326–1339, 2009.

[30] J. Ham and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *ICML*, 2008, pp. 376–383.

[31] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *CVPR Workshop*, 2010.