Deformation Based 3D Facial Expression Representation

GIRUM G. DEMISSE*, DJAMILA AOUADA, and BJÖRN OTTERSTEN, University of Luxembourg

We propose a deformation based representation for analyzing expressions from 3D faces. A point cloud of a 3D face is decomposed into an ordered deformable set of curves that start from a fixed point. Subsequently, a mapping function is defined to identify the set of curves with an element of a high dimensional matrix Lie group, specifically the direct product of SE(3). Representing 3D faces as an element of a high dimensional Lie group has two main advantages. First, using the group structure, facial expressions can be decoupled from a neutral face. Second, an underlying non-linear facial expression manifold can be captured with the Lie group and mapped to a linear space, Lie algebra of the group. This opens up the possibility of classifying facial expressions with linear models without compromising the underlying manifold. Alternatively, linear combinations of linearised facial expressions can be mapped back from the Lie algebra to the Lie group. The approach is tested on the BU-3DFE and the Bosphorus datasets. The results show that the proposed approach performed comparably, on the BU-3DFE dataset, without using features or extensive landmark points.

CCS Concepts: • **Computing methodologies** → **Computer vision representations**; **Shape representations**; *Appearance and texture representations*; *Hierarchical representations*;

Additional Key Words and Phrases: 3D facial expression representation, expression modelling, 3D face deformation

ACM Reference format:
Girum G. Demisse, Djamila Aouada, and Björn Ottersten. 2018. Deformation Based 3D Facial Expression Representation. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (January 2018), 22 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Representing facial expressions is an integral part of what is called affective computing [Pantic 2009; Zeng et al. 2009]. These representations are used in combination with a classifier to recognize facial expressions and infer affect. Many other disciplines such as human computer interaction, computer graphics, health monitoring, also benefit from the ability to model facial expressions for the purpose of analysis, animation, and recognition. Over the last decades, several methods were proposed for facial expression representation. In general, these methods can be categorized into two main categories; 1) methods based on feature descriptors, and 2) methods based on a generic expression space learning.

In the first category, features are labelled, according to their type, as either geometric or appearance. Subsequently, both feature types are further labelled, according to their construction, as predefined [Pantic and Bartlett 2007], directly learned from a training data [Liu et al. 2014; Ranzato et al. 2011], or a hybrid thereof [Zhang and Ji 2005]. Approaches

*The corresponding author

Author's addresses: G. Demisse, D. Aouada, B. Ottersten, University of Luxembourg, 29, avenue JF Kennedy, L-1855 Luxembourg.
This work was partly funded by the European Union's Horizon 2020 research and innovation project STARR under grant agreement No.689947.

Manuscript submitted to ACM

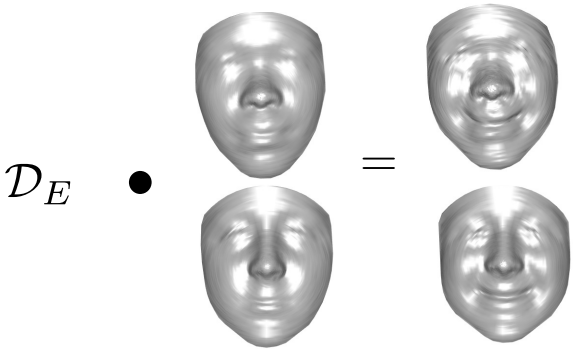


Fig. 1. Proposed 3D facial expression representation. A smile is captured by the deformation $\mathcal{D}_E$ and applied to two different subjects, preserving their specific shapes.

based on predefined features aim to detect facial action units (AU) defined in [Ekman and Friesen 1977], and defer the task of expression labelling to a higher level processing, while in learned features, the attempt is to learn a descriptive representation directly from a training data. In [Bartlett et al. 1996; Pantic 2009], methods that detect facial AU are argued to be comprehensive and robust to subjective labelling of expressions – since the detection of AU is decoupled from expression detection, a new facial expression can be discovered by combining AUs. On the other hand, systems that integrate feature learning with expression labelling have been shown to benefit from the supervised learning of strong discriminative features [Liu et al. 2014; Taigman et al. 2014]. In general, mapping functions from the raw data space to the feature space are not necessarily bijective, hence inversion of features is numerically approximated [Mahendran and Vedaldi 2015; Vondrick et al. 2013]. Consequently, it is not straightforward to translate a linear combination or scalar multiplication of features to the raw data space. Moreover, most feature based approaches depend on pre-annotated/estimated landmark points [Fang et al. 2011]. Apart from the computational overhead, methods that depend on estimated landmark points have to account for landmark estimation or annotation error. Although it is not clear how error propagates, in [Taheri et al. 2014] discrepancies in labelling the landmark/action units are presented as the major reason for low accuracy. In the second category of facial expression representations, a general space of faces and their expressions is estimated. In [Ham and Lee 2007], an L_2 norm is defined on the displacement field of faces to learn an embedding of the expression space. In [Chang et al. 2005], a 3D template face is used to match all faces and learn a generalized expression space rather than a per subject learning. However, empirically estimating the space of facial expressions from a small dataset in high dimensional space is difficult and most of the time only a small portion of the space ends up being estimated, e.g., expression space of a given subject. As such, the estimated expression space is not guaranteed to be complete or connected [Bengio et al. 2013]. One consequence of this is the need for search based computational schemes, e.g., the geodesic distance between expressions is computed using graph-based shortest path algorithms [Chang et al. 2006; Ham and Lee 2007]. Meanwhile, in [Al-Osaimi et al. 2009; Mpiperis et al. 2008; Tenenbaum and Freeman 2000], a linearity prior is introduced in the estimation of the expression space. In [Mpiperis et al. 2008; Tenenbaum and Freeman 2000], an observed facial data is formulated as a bilinear function of face and expression. Thus, expressions are modelled as linearly separable objects. In a similar spirit to our approach, in [Al-Osaimi et al. 2009] the point-to-point difference between a neutral face and a face with an expression is taken as expression residue and the expression space is estimated with PCA (principal component analysis). However, expression space is a non-linear space

and linear models fail to discover the underlying non-linear manifold [Wang et al. 2004]. In a much similar approach as is presented in this paper, in [Drira et al. 2010, 2013; Samir et al. 2006, 2009] a facial surface is decomposed into a set of facial curves and the average distance between the aggregate facial curves is used as a similarity metric between faces for identity recognition. Meanwhile, in [Kurtek and Drira 2015] instead of decomposing a facial surface into facial curves, a given face is studied as a parametrized surface.

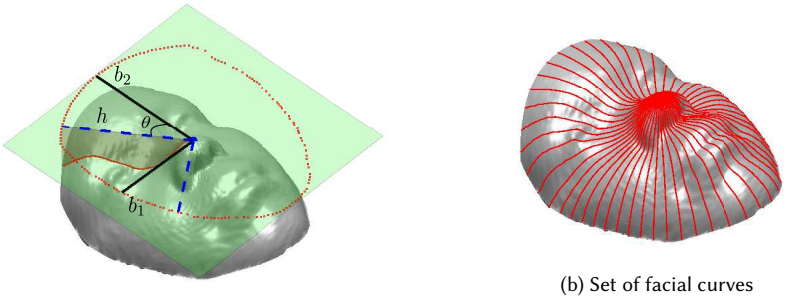
This paper proposes a novel method for facial expression representation from 3D data without appearance (texture) information or annotated landmark points. Our approach starts by sampling a 3D facial surface into a set of *facial curves*, sometimes referred to as "radial curves" [Aouada et al. 2014; Drira et al. 2013]. The sampled facial curves are then represented as an element of a matrix Lie group, as described in [Demisse et al. 2016]. Subsequently, by taking a direct product of the facial curve representations, a mapping function that identifies a set of facial curves with an element of a high dimensional matrix Lie group is defined. Given such a mapping function, facial expressions are then represented by a left action of the group on a neutral face representation, see Fig. 1. Although the idea of decoupling facial expression from a neutral face is presented in [Al-Osaimi et al. 2009], the representation space presented here, Lie group, models expression space as a non-linear space. Consequently, captures non-linear variation of facial expressions. Concurrently, the proposed representation can be linearised by mapping the expression representations from the Lie group to the Lie algebra. Hence, conventional linear models, like SVM (support vector machine), can directly be trained on the representation. The main contributions of this paper are summarized as follow:

(1) We present a non-linear facial expression representation without the need for annotated landmarks. We show the potential of the proposed representation for expression recognition by training an SVM classifier and testing it on the BU-3DFE dataset [Yin et al. 2006] and on the Bosphorus dataset [Savran et al. 2008].
(2) Using the group and differential structure of the representation space, we define a closed form solution for the geodesic distance between facial expressions, which is not the case in some manifold learning based approaches. Furthermore, we show that facial expression can be defined as a *regular* group action on the facial surface representation. Consequently, a facial expression can be identified uniquely upto the surface parametrizations.
(3) We propose a framework that reformulates optimal surface parametrization as optimal the facial curve sampling such that the deformation between faces is smooth and least costly.

The rest of the paper is organized as follows: Section 2 describes the data preprocessing and curve representation approach. In Section 3, we present the proposed facial expression representation. In Section 4, we formulate an objective functional for estimating curve correspondence between two faces and detail a dynamic programming based solution in Section 5. In Section 6, we described a basic approach for training a model with a dataset of expressions represented with the proposed method. Experimental results are discussed in Section 7. The paper concludes with final remarks in Section 8.

2 BACKGROUND

We decompose a segmented, hole-free, facial surface into a family of curves that start from a given reference point, see Fig. 2b. This decomposition allows to view a facial surface as an ordered set of curves and a facial expression as its deformation. The curve representation given in [Demisse et al. 2016] is used to represent both the set of facial curves and their deformation as an element of a Lie group, specifically a high dimensional direct product of the Special Euclidean group, SE(3) – $g \in SE(3)$ is a semi-direct product of rotation, $R \in SO(3)$, and translation, $v \in \mathbb{R}^3$. In what follows, we



(a) Parametrization

(b) Set of facial curves

Fig. 2. Facial curve extraction.

cover the data preprocessing and facial curve extraction stage, and the necessary backgrounds on representing curves in $\mathbb{R}^3$.

2.1 Preprocessing and surface decomposition

The main causes of variation in a dataset of 3D facial point clouds, with respect to a given fixed coordinate system, are deformation, scaling, translation and rotation. Among those, the most informative variation is the one due to deformation as the rest does not change the nature of the shape [Kendall 1984]. Nevertheless, transformation of the coordinate frame can be manifested as shape preserving transformations of the facial 3D point cloud, for instance different extrinsic camera parameters will lead to different measurements. Consequently, in the preprocessing stage we filter shape preserving transformations with respect to a given fixed world coordinate system.

Removing location and scale: A given facial point cloud $\Gamma = \{p_1, \cdots, p_N\}$, with $p_i \in \mathbb{R}^3$, is centred to zero mean and unit norm as follows:

$$\Gamma = \left\{ \frac{p_1 - \bar{p}}{h}, \cdots, \frac{p_N - \bar{p}}{h} \right\}, \tag{1}$$

where

$$\bar{p} = \frac{1}{N} \sum_{i}^{N} p_i, \quad \text{and} \quad h = \sqrt{\sum_{i}^{N} \|p_i - \bar{p}\|_2^2}, \tag{2}$$

Equation (2) computes the mean and the L_2 norm of the vectorized point cloud.

Filtering rotation: The head orientation of a normalized facial point cloud Γ is aligned with an arbitrarily selected reference face, using the Iterative Closest Point (ICP) algorithm. We note that ICP gives a reasonable result only when the point cloud is a segmented and hole free facial data. Subsequently, singular value decomposition (SVD) is used for estimating the coordinate orientation of the reference face and the given face to further refine the alignment. We again stress that the data should be described from a fixed reference coordinate system for the representation to be meaningful.

Facial surface decomposition: We first select the reference point p_r as the tip of the nose; this is mainly because it is relatively easer to estimate. The estimation is done by selecting the point with a maximum component in the direction

of the eigenvector with the smallest eigenvalue. That is

$$p_r = \arg\max_{\forall p \in \Gamma} \mathbf{P}_j \cdot p, \tag{3}$$

where $\mathbf{P}_j$ denotes the eigenvector with the smallest eigenvalue and the operation $\cdot$ denotes the dot product; the eigenvectors and eigenvalues are computed with SVD on the normalized facial data. Subsequently, let P_t be the tangent plane at p_r with b_1 and b_2 as its orthonormal basis, see Fig. 2a. Then for some $\theta \in [0, 2\pi]$, a facial curve on the surface Γ is given as follows

$$\begin{aligned}\Gamma(\theta) = \{p \in \Gamma \mid &(R(\theta) \times b_1) \cdot (p - p_r) = 0 \\ &\wedge ((R(\theta) \times b_2) \cdot (p - p_r) > 0)\},\end{aligned} \tag{4}$$

where R is a rotation matrix about the normal vector of P_t, see Fig. 2a. Furthermore, a given facial curve can be parametrized by $r \in [0, h]$, where h is the mean radius of the surface projection on to the rotation plane. Consequently, the parametrization of the full facial surface is given by $\Gamma(r, \theta)$ such that $\Gamma(0, \theta) = p_r, \forall \theta$. The parametrization can then be used to extract a set of curves by defining the values of θ and r.

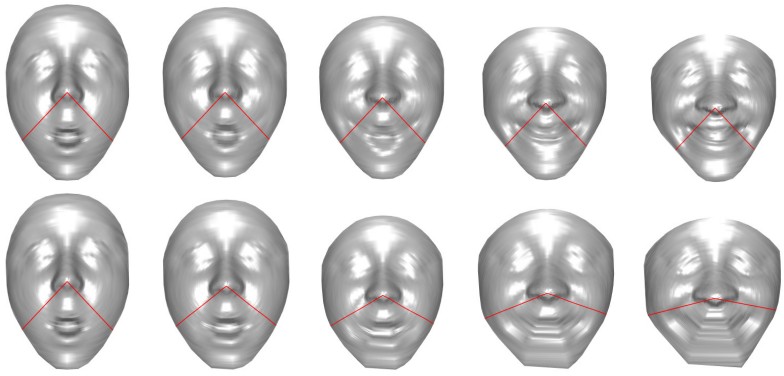
2.2 Curve representation

There are several works in curved shape representation and geodesic distance computation [Demisse et al. 2016, 2017; Mennucci 2013; Michor et al. 2007; Srivastava et al. 2011; Younes 1998]. The general approach in modelling curves is to consider a given curve as a function from $\mathbb{R}$ to $\mathbb{R}^n$. In practice, however, curves are represented by z discrete set of points [Kendall 1984]. Subsequently, statistical properties of curves, in the space of the selected points, are computed by either taking the L_2 norm in Euclidean space or using spline based distances [Dryden and Mardia 1998]. Alternatively, in [Demisse et al. 2015, 2016; Michor et al. 2007; Younes 1998] the statistical properties of curves are studied in the diffeomorphism group of the space of selected points. In this paper, we adopt the curve representation approach described in [Demisse et al. 2016, 2017]. Curves in [Demisse et al. 2016] are represented by z rigid transformation matrices such that the sequential action of the matrices on a fixed reference point reconstructs the full sequence of selected points. As a result, any subsequent geometric or statistical quantities of the curves are computed using both the group and the differential structure of the representation space. In this Subsection, we will briefly describe the curve representation framework presented in [Demisse et al. 2016, 2017] for open curves in $\mathbb{R}^3$.

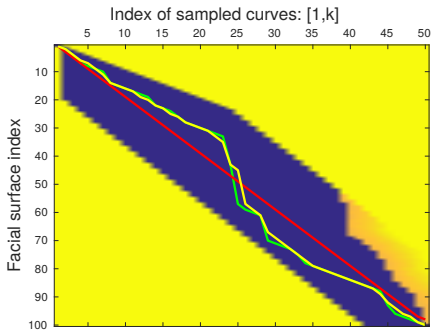
Let $\bar{\psi}_i$ be a continuous curve in $\mathbb{R}^3$ parametrized by arc-length, and let $r : [1, z] \to [0, \ell]$ be a monotonic and injective map, where z is any positive integer and ℓ is the length of the curve. Note that in the case of facial curves, ℓ is replaced by h. Then, $\psi_i = \bar{\psi}_i \circ r$ is a z-ordered distinct sample points of $\bar{\psi}_i$, with $\circ$ denoting function composition. For some fixed integer z, a family of curves is defined as $C_z = \{\psi_i | \psi_i \ni p_1 = p_r\}$ where p_r is a fixed reference point. As such, C_z is the space of all curves described by z discrete points starting from a fixed reference point p_r. Furthermore, any curve in C_z can be deformed to another by the left action of $\mathrm{SE}(3)^{z-1}$, a direct product of 3-dimensional Special Euclidean group. That is to say, $\forall \psi_j, \psi_i \in C_z, \exists \mathcal{G} \in \mathrm{SE}(3)^{z-1} : \mathcal{G}\psi_i = \psi_j$. It is worth to note that all deformations leave the first point fixed. Subsequently, a mapping $f : C_z \to \mathrm{SE}(3)^{z-1}$ is defined as

$$f(\psi_i) = \mathcal{G} = (g_1, \cdots, g_{z-1}), \tag{5}$$

such that $g_i \times p_i = p_{i+1}$.



(a)



(b)

Fig. 3. Curve matching between two different faces with “surprise", on the far left, and “happy", on the far right, expressions. The first row shows the geodesic deformation between the faces when both faces are represented by 50 uniformly sampled curves. The red curves are tracked along the deformation to illustrate the mismatch. In the second row, the “surprise" face is represented by 50 uniformly sampled curves while the “happy" faces is optimally sampled via dynamic programming, see Section 4. The cost matrix shows three solutions for different weighting factors–the blue region is the feasible set defined by the sectors size. The red curve is the optimal solution for $\alpha = 0$ and $\beta = 1$, the green path is optimal for $\alpha = 1$ and $\beta = 0$, finally the yellow path is optimal for $\alpha = 1$ and $\beta = 1$, the second deformation is based on the yellow path.

Equation (5) provides the representation of a given z-sampled curve. The inverse of the mapping, given the fixed starting point, is

$$f^{-1}(\mathcal{G}) = (p_r, g_1 p_r, g_2 g_1 p_r, \cdots, g_{z-1} \cdots g_1 p_r). \tag{6}$$

Note that, among all possible transformations between two points it is the least action transformation that is represented by the g's, see [Demisse et al. 2016, 2017]. Deformation in the representation space is given by the right or the left action of the group on itself. For example, deformation of the curve ψ_i to the curve ψ_j is given by the left action of

$$\mathcal{G}^L = f(\psi_j) f(\psi_i)^{-1}, \tag{7}$$

or the right action of

$$\mathcal{G}^R = f(\psi_i)^{-1} f(\psi_j), \tag{8}$$

where $f(\cdot)^{-1}$ denotes a pair-wise inversion[1]. Moreover, the Riemannian and group structure of the representation space is used to give an explicit solution to the geodesic equation, as presented in [Demisse et al. 2016, 2017; Michor et al. 2007]. To that end, the geodesic path between two curved shape representations $f(\psi)$ and $f(\psi^*)$ is given as

$$\Phi(t) = (\varphi_1(t), \cdots, \varphi_{z-1}(t)) \quad \text{where} \quad t \in [0,1] \tag{9}$$

$$\text{and} \quad \varphi_i(t) = \begin{pmatrix} R_i(R_i^{-1}R_i^*)^t & v_i + (v_i^* - v_i)t \\ 0 & 1 \end{pmatrix}. \tag{10}$$

Equation (10) computes the geodesic path between the i^{th} transformation matrix of the two shape representations, and the parameter t is a time step one takes along the geodesic path. Subsequently, the geodesic distance between two curves is given as

$$d_c(f(\psi), f(\psi^*)) = (d(g_1, g_1^*)^2 + \cdots + d(g_z, g_z^*)^2)^{1/2}, \tag{11}$$

$$\text{with} \quad d(g_i, g_i^*) = (\|\log(R_i^T R_i^*)\|_F^2 + \|v_i^* - v_i\|_2^2)^{1/2}, \tag{12}$$

$\|\cdot\|_F$ denotes the Frobenius norm. The defined distance metric is left invariant, see [Demisse et al. 2016, 2017] for further details.

3 PROPOSED FACIAL EXPRESSION REPRESENTATION

In this section, we propose to represent face and facial expressions as elements of a Lie group.

3.1 Face representation

Let $\Psi = \{\psi_1, \cdots, \psi_k\}$ be a set of k facial curves sampled from a facial surface Γ as described in Section 2. Given the curve representation in Section 2.2, a straightforward representation $\mathcal{D}$ of Ψ is to take the direct product as $\mathcal{F} : (C_z)^k \to (\text{SE}(3)^{z-1})^k$ such that

$$\mathcal{D} \coloneqq \mathcal{F}(\Psi) = (f(\psi_1), \cdots, f(\psi_k)) = (\mathcal{G}_1, \cdots, \mathcal{G}_k). \tag{13}$$

The inverse of the mapping function is given as

$$\mathcal{F}^{-1}(\mathcal{D}) = \{f^{-1}(\mathcal{G}_1), f^{-1}(\mathcal{G}_2), \cdots, f^{-1}(\mathcal{G}_k)\}, \tag{14}$$

where $f^{-1}(\cdot)$ is as defined in (6). The geodesic path $\Lambda(t)$ and distance d_F between two facial representations, $\mathcal{F}(\Psi)$ and $\mathcal{F}(\Psi^*)$, are given by direct products using (9) and (11), respectively. Specifically,

$$\Lambda(t) = (\Phi_1(t), \cdots, \Phi_k(t)), \tag{15}$$

where $\Phi(\cdot)$ is as defined in (9), see Fig. 3. The distance d_F is defined as

$$d_F(\mathcal{F}(\Psi), \mathcal{F}(\Psi^*)) = (d_c(\mathcal{G}_1, \mathcal{G}_1^*)^2 + \cdots + d_c(\mathcal{G}_k, \mathcal{G}_k^*)^2)^{1/2}, \tag{16}$$

where $d_c(\cdot,\cdot)$ is as defined in (11).

[1]Note that $f(\cdot)^{-1}$ is different from $f^{-1}(\cdot)$; the former computes pair-wise inverse of the transformation, while the later is inversion of the map.

3.2 Expression representation

Similar to the curve representation in Section 2.2, a deformation between two facial representations $\mathcal{F}(\Psi)$ and $\mathcal{F}(\Psi^*)$ is given by the action of a group on itself. A deformation that acts from the left is

$$\mathcal{D}^L = \mathcal{F}(\Psi^*)\mathcal{F}(\Psi)^{-1}. \tag{17}$$

Alternatively, a deformation that acts from the right is

$$\mathcal{D}^R = \mathcal{F}(\Psi)^{-1}\mathcal{F}(\Psi^*). \tag{18}$$

where $\mathcal{F}(\Psi)^{-1} = (f(\psi_1)^{-1}, \cdots, f(\psi_k)^{-1})$, $f(\cdot)^{-1}$ is as defined in (7). In general, a left action of a group G on a set Y is given as $G \times Y \mapsto Y$. Moreover, the action is said to be *regular* if for every $x, y \in Y$ there exists exactly one $g \in G$ such that $gx = y$. In our case, Y is the group G itself, hence, the action of (17) is regular. To see this fact, consider $\mathcal{F}(\Psi), \mathcal{F}(\Psi^*) \in (\mathrm{SE}(3)^{z-1})^k$ such that

$$\mathcal{D}^L\mathcal{F}(\Psi) = \mathcal{F}(\Psi^*). \tag{19}$$

Subsequently, let us assume that there is another $\mathcal{D}_2^L \neq \mathcal{D}^L$ that satisfies the condition $\mathcal{D}_2^L\mathcal{F}(\Psi) = \mathcal{F}(\Psi^*)$. In that case,

$$\mathcal{D}^L\mathcal{F}(\Psi) = \mathcal{D}_2^L\mathcal{F}(\Psi^*), \tag{20}$$

since $(\mathrm{SE}(3)^{z-1})^k$ is a group (20) implies $\mathcal{D}^L = \mathcal{D}_2^L$. Consequently, given both the representations of a face with a neutral expression $\mathcal{F}(\Psi_N)$ and a non-neutral expression $\mathcal{F}(\Psi_E)$, we can uniquely identify a deformation due to the non-neutral expression, upto surface parametrization, as

$$\mathcal{D}_E = \mathcal{F}(\Psi_E)\mathcal{F}(\Psi_N)^{-1}. \tag{21}$$

Hence, we use $\mathcal{D}_E$ to represent a facial expression irrespective of subject specific facial shapes.

4 CURVE CORRESPONDENCE

The proposed expression representation, Subsection 3.2, assumes the parametrization of two different faces to be optimal when factoring deformations. That is, the index of a curve that passes through a particular region of the mouth in $\mathcal{F}(\Psi_N)$ is assumed to correspond with the index of a curve that covers the same mouth region in $\mathcal{F}(\Psi_E)$. In such a case, the factored expression reflects the deformation of a curve due to the observed expression. Such an assumption, however, is violated when there is a significantly large non-linear deformation between faces, see Fig. 3. Consequently, in case of large non-linear deformations, the factored expression includes deformations that reflect the curve mismatch rather than the observed expression. In this Section, we present a cost functional for matching facial curves optimally and propose dynamic programming based solution.

4.1 Cost of mismatching curves

Let $\theta : [0, k] \to [0, 2\pi]$, where k is the number of curves. Similar to the radial sampling r, see Section 2.2, we insist on the angular sampling θ to be injective, monotonic, and we restrict its initial and last values to $\theta(0) = 0$ and $\theta(k) = 2\pi$. Consequently, θ will define the angles used to sample facial curves from a facial surface Γ. That is, for a given facial surface Γ, different families of curves $\Gamma(r_j, \theta_i)$ are obtained for different values of i and j. Subsequently, for a fixed curve parametrization r, i.e., for a fixed radial sampling, we define the space of deformations between two facial surfaces Γ

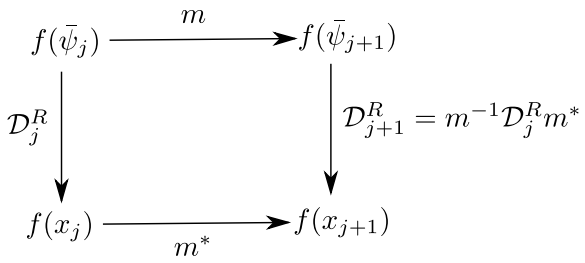


Fig. 4. Commutative diagram. Deformation between two curves is represented in terms of the deformation between previously selected curves. This formulation is used to estimate curve correspondence via dynamic programming.

and Γ^* as

$$\Omega = \{\mathcal{D}^R | \exists \theta_i, \theta_j : \mathcal{F}(\Gamma(r,\theta_i))\mathcal{D}^R = \mathcal{F}(\Gamma^*(r,\theta_j))\}. \tag{22}$$

Since the angular sampling depends on the parametrization (see Section 2.1), the distance between two face representations can be written in terms of the parameters (θ_i, θ_j) as

$$\min_{\theta_i,\theta_j} d_F(\mathcal{F}(\Gamma(r,\theta_i)), \mathcal{F}(\Gamma^*(r,\theta_j))). \tag{23}$$

Next, using the left invariance property of the distance metric, see Section 2.2, we write (23) as

$$\min_{\theta_i,\theta_j} d_F(\mathcal{F}(\Gamma(r,\theta_i)), \mathcal{F}(\Gamma(r,\theta_i))\mathcal{D}^R) = \min_{\theta_i,\theta_j} d_F(e, \mathcal{D}^R), \tag{24}$$

where $\mathcal{D}^R = \mathcal{F}(\Gamma(r,\theta_i))^{-1}\mathcal{F}(\Gamma^*(r,\theta_j))$ and e is the identity in $(\mathrm{SE}(3)^{z-1})^k$. Intuitively, the functional given in (24) attempts to find the least costly deformation from Ω, see (22). In effect, optimal curve parametrizations of the faces, θ_i and θ_j. However, our definition of Ω permits angular samplings that does not preserve geometric properties, e.g., volume of the face with respect to its support plane. Thus, a solution parametrization might give the least costly deformation, according to (24), but can deviate from the target shape, see Fig. 3. To address this problem, we add a term that penalizes parametrizations that do not preserve a particular geometric property; in our case volume.

The enclosed volume of a face Γ, parametrized with r and θ, with respect to the rotation plane is given as

$$\mathrm{Vol}(\Gamma(r,\theta)) = \int_0^{2\pi}\int_0^h r\Gamma(r,\theta)\, dr\, d\theta. \tag{25}$$

Subsequently, for a given θ the volume is approximated as

$$\mathrm{Vol}(\Gamma(r,\theta)) \approx \sum_{j=1}^{k-1} \Delta\theta(j) \sum_{i=1}^{z-1} h\Delta\Gamma(\theta(j), r(i))\Delta r(i), \tag{26}$$

where Δ denotes the forward difference. As a result, the approximate volume given in (26) depends on θ, since r is fixed for simplicity, see (22). Assuming a uniform angular sampler $\bar{\theta}$ preserves volume, the objective functional given in (24) is penalized by the difference between the volume of the face due to a candidate solution sampler and a uniform

sampler, which is formulated as

$$\begin{aligned}\arg\min_{\theta_i,\theta_j}\{&\alpha \times d_F(e,\mathcal{D}^R)\\ &+\beta \times (\mathrm{Vol}(\Gamma(r,\theta_i)) - \mathrm{Vol}(\Gamma(r,\bar{\theta})))\\ &+\beta \times (\mathrm{Vol}(\Gamma^*(r,\theta_j)) - \mathrm{Vol}(\Gamma^*(r,\bar{\theta})))\},\end{aligned} \tag{27}$$

with β and α as scalar weighting terms. Thus, a large value of β encourages volume preserving solutions, while a large value of α encourages deformation optimizing solutions.

5 DYNAMIC PROGRAMMING BASED SOLUTION

In this Section we reformulate (27) as a dynamic optimization problem. To simplify the computational cost, we fix the angular sampler θ of the surface Γ to a uniform sampler and optimize for the sampler of the other surface Γ^*. As a result, (27) is simplified to

$$\begin{aligned}\arg\min_{\theta_j}\{&\alpha \times d_F(e,\mathcal{D}^R)\\ &+\beta \times (\mathrm{Vol}(\Gamma^*(r,\theta_j)) - \mathrm{Vol}(\Gamma^*(r,\bar{\theta})))\},\end{aligned} \tag{28}$$

Next, we write (28) as a recursive function to estimate its solution via dynamic programming. To elaborate, we first write a general form of a facial surface decomposition as

$$\Gamma^*(r,\theta) = \{x_1,\cdots,x_k\}: \qquad x_j \subset U_j. \tag{29}$$

The $U_j \subset \Gamma^*$ are sectors of the face, i.e., subsets of the facial surface from which x_j can take its values. Hence, the sector size introduces a constraint in the search space of each x_j. In this work we approximate the sectors U_j by a fixed size sliding window. Subsequently, we can rewrite the first term of (28) as a dynamic optimization problem as

$$d_F(e,\mathcal{D}^R)^2 = \alpha \times \sum_{j=1}^{k-1} \phi_j(x_j, x_{j+1}). \tag{30}$$

Given $\bar{\psi}_j$ and $\bar{\psi}_{j+1}$ as the uniformly sampled curves of Γ, we define ϕ_j as

$$\phi_j(x_j, x_{j+1}) = d_c(e, m^{-1}(\mathcal{D}_j^R)m^*)^2, \tag{31}$$

such that

$$m^{-1} = f(\bar{\psi}_{j+1})^{-1} f(\bar{\psi}_j), \tag{32}$$

and

$$m^* = f(x_j)^{-1} f(x_{j+1}). \tag{33}$$

Hence, $\mathcal{D}_{j+1}^R = m^{-1}(\mathcal{D}_j^R)m^*$, see Fig. 4. Similarly, the remaining term of (28) can be written as a recursive function with optimal substructure as

$$\beta \times \sum_{j=1}^{k-1} (\mathrm{Vol}(\Gamma^*(r,\theta))\Big|_{x_j}^{x_{j+1}} - \mathrm{Vol}(\Gamma^*(r,\bar{\theta}))\Big|_{\bar{\psi}_j}^{\bar{\psi}_{j+1}}). \tag{34}$$

Consequently, by writing (28) in terms of (30) and (34) we formulate the objective function as a dynamic optimization problem and estimate the solution via dynamic programming, see Fig. 3 and Subsection 5.1. For a fixed sector size s, the time complexity of optimally parametrizing one face while the other is fixed is $O(s^2k)$. In the next Subsection, we discuss implementation details of the dynamic programming.

5.1 Implementation

In this subsection, we describe the implementation of the curve correspondence estimation between two facial surfaces, Γ and Γ^*, using dynamic programming. We assume that Γ is approximated by k uniformly sampled curves, while Γ^* is approximated by $K \gg k$ curves. In effect, the goal is to sample k curves from Γ^* such that the deformation from Γ to Γ^* is least costly, according to (28) , while preserving the volume of Γ^*.

Given a linearly recursive cost functional, dynamic programming estimates the solution by solving for the minimum and the minimizer of the cost functional's sub-problems, sequentially. We begin by selecting the first curves $\psi_1^* \in \Gamma^*$ as a matching curve for $\psi_1 \in \Gamma$, i.e., the cost of selecting the first curve is

$$C_1(x_1 = \psi_1^*) = 0. \tag{35}$$

We then explore the solution space, using the linearly separable substructures, of the cost functional which is constrained by the sectors, see Section 5. In each step we save the minimum and the minimizer of the cost functional to later use them to estimate the solution by working backwards. To be more explicit, let

$$\rho(x_j, x_{j+1}) = \underbrace{\alpha \times \phi_j(x_j, x_{j+1})}_{\text{Deformation term}} + \underbrace{\beta \times \text{Vol}(\Gamma^*(r,\theta))\Big|_{x_j}^{x_{j+1}} - \text{Vol}(\Gamma^*(r,\bar{\theta}))\Big|_{\bar{\psi}_j}^{\bar{\psi}_{j+1}}}_{\text{Volume penality term}}, \tag{36}$$

see Section 5. We then compute the minimum cost for every $x_{j+1} \in U_{j+1}$ as

$$\begin{aligned} C_{j+1}(x_{j+1}) &= \min_{x_1,\cdots,x_j} \sum_{i=1}^{j} \rho_i(x_i, x_{i+1}), \quad \forall x_{j+1} \in U_{j+1} \\ &= \min_{x_j} C_j(x_j) + \rho_j(x_j, x_{j+1}), \quad j = 1, \cdots, k-1 \end{aligned}$$

Meanwhile, the minimizers are given by

$$O_{j+1}(x_{j+1}) = \arg\min_{x_j} C_{j+1}(x_{j+1}). \tag{37}$$

In order to get a monotonic and injective sampler, we impose further restriction on the values x_j can take. To that end, let $\text{Vol}|_{[0,n]}$ be the volume of the facial surface sector up to ψ_n^*, where ψ_n^* is a facial curve of Γ^*. Subsequently, if $x_{j+1} = \psi_n^*$ then we will only consider $x_j = \psi_a^* \in U_j : \text{Vol}|_{[0,a]} < \text{Vol}|_{[0,n]}$. The proposed facial curves correspondence estimation is summarized in Algorithm 1.

Limitations: The proposed curve correspondence estimation is done in a restricted way, i.e., we are optimally sampling curves and leaving the point sampling to uniform sampling. Hence the solution to the objective functional is a curve in the search space. We believe a conjugate optimization of both curve and point sampling is more reliable and complete than optimal curve sampling, which would give a surface as a solution. However, it is highly taxing in computational time, see Fig. 5b. Alternative in [Kurtek and Drira 2015], optimal parametrization of a facial surface, as a whole, is estimated instead of the curves. Moreover, the weighting terms, β and α, on the constraint of the objective functional

Algorithm 1: Optimal sampling of curves from Γ^*

Data: $\Gamma(\bar{\theta}), \Gamma^*$
Initialization: $s = |U|$, $k = |\Gamma(\bar{\theta})|$, $j = 1$, $O_1(1) = \psi_1^*$;
for $j < k$ **do**
  for $x_{j+1} \subset U_{j+1}$ **do**
    $U = \{\psi_a^* \in U_j : (\text{Vol}|_{[0,a]} < \text{Vol}|_{[0,j+1]})\}$;
    for $x_j \subset U$ **do**
      $C_{j+1}(x_{j+1}) = \rho_i(x_j, x_{j+1}) + C_j(x_j)$;
    end
    $C_{j+1}(x_{j+1}) = \min_{x_j \in U_j} C_{j+1}(x_{j+1})$;
    $O_{j+1}(x_{j+1}) = \arg\min_{x_j \in U_j} C_{j+1}(x_{j+1})$;
  end
end
$q = k - 1$;
for $q > 1$ **do**
  $\psi_q^* = O_{q+1}(\psi_{q+1}^*)$;
end
Result: $\Gamma^* = \{\psi_1^*, \cdots, \psi_k^*\}$

are manually tuned and not estimated, see Fig. 6 and Fig 7. However, an automatic estimation of these parameters would be desirable, since one would expect the parameters to be expression-category specific. One possible solution to achieve this is to consider a combination of dynamic programming and Lagrangian multiplier methods [Bellman 1956]. Alternatively, the parameters can be estimated, from a discretized parameter space, for each expression category using brute-force search guided by, for example, cross-validation performance.

6 MODELLING EXPRESSIONS

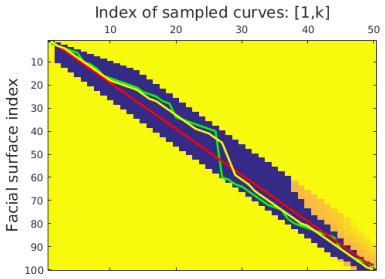
The presented approach represents expressions in a non-linear space, a matrix Lie group. However, one can easily linearize a representation of a given expression $\mathcal{D}_E$ by projecting it to the group's Lie algebra, which is a vector space. To define the projection we first define the mapping of a matrix $g \in \text{SE}(3)$ to its Lie algebra $\mathfrak{se}(3)$ by a matrix logarithm as

$$\log : \text{SE}(3) \mapsto \mathfrak{se}(3), \tag{38}$$

see [Varadarajan 2013] for further details. Subsequently, by taking the direct product of (38) we define the mapping of an expression $\mathcal{D}_E$ to the Lie algebra as $\mathcal{L}(\mathcal{D}_E) = ((\log)^{z-1})^k(\mathcal{D}_E)$. Under such linearization, an expression will be $(3+3) \times (\#\text{points} - 1) \times (\#\text{curves})$ dimensional vector, where the threes are counts of the independent components of the skew-symmetric matrix and the translation. Alternatively, an expression represented on the Lie algebra can be mapped back to the Lie group using a direct product of matrix exponentials defined as

$$\exp : \mathfrak{se}(3) \mapsto \text{SE}(3), \tag{39}$$

we use $\mathcal{E}$ to denote the direct product of (39). Thus $\mathcal{E}(\mathcal{L}(\mathcal{D}_E)) = \mathcal{D}_E$. Consequently, using $\mathcal{L}$ one can train linear discriminate models, e.g., SVM, on the Lie algebra. Alternatively, a linear combination or scaling of a linearized expression can be mapped back to the Lie group using $\mathcal{E}$, see Fig. 8.



(a) Cost Matrix

(b) Geodesic deformation

Fig. 5. Curve matching. In (a) a cost matrix with three solutions is shown that correspond to the three deformations in (b). In all the deformations, the first faces are approximated by 50 uniformly sampled curves, i.e., they are Γ, while the last faces are sampled optimally, i.e., they are Γ^*. Pair of facial curves are highlighted in red in all of the deformations to illustrate the impact of the matching. The deformation in the first row is according to the red path in the cost matrix which is the solution when $\alpha = 0$ and $\beta = 1$, that is when both faces are sampled uniformly. The deformation in the second row is based on the green path which is the solution when $\alpha = 1$ and $\beta = 0$, that is when there is no volume based constraint. The last deformation is based on the yellow path which is computed for $\alpha = 1$ and $\beta = 4$. In this particular example, regardless of the good curve matching solution given by the yellow path, the deformation of the lower lip is not smooth. *This is mainly because we are only matching curves and disregarding point matching, see supplementary video.*

7 EXPERIMENTS

In this section, we evaluate the proposed approach in recognizing what are known as the universal expressions–*anger* (AN), *happiness* (HA), *surprise* (SU), *fear* (FE), *sadness* (SA), and *disgust* (DI). In [Yin et al. 2008; Zhang et al. 2014], dynamic 3D data facial expression datasets are provided. However, in order to evaluate the proposed approach on a dynamic 3D facial dataset one needs to have a time series model, which is beyond the scope of this paper. Consequently, we evaluate our approach on the BU-3DFE dataset [Yin et al. 2006] and the Bosphorus dataset [Savran et al. 2008]. In the subsequent sections, we detail the experimental protocol and results on each of the above datasets.

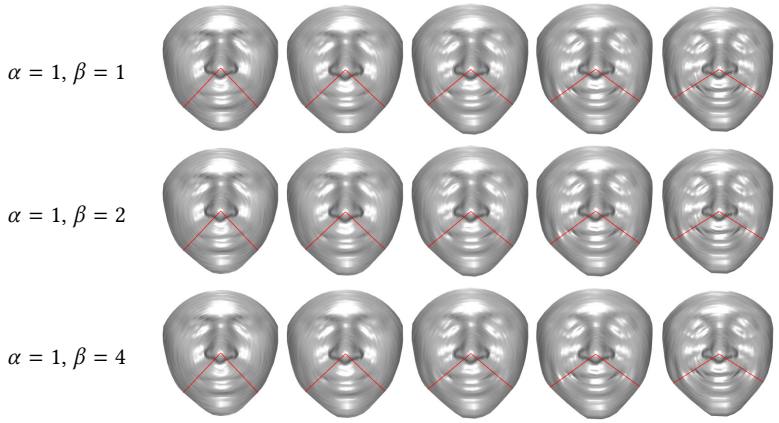


Fig. 6. Curve matching for a relatively small deformation under three different β values. Note that, the impact of β is minimal on the matching results since the deformation is relatively small, see supplementary video.

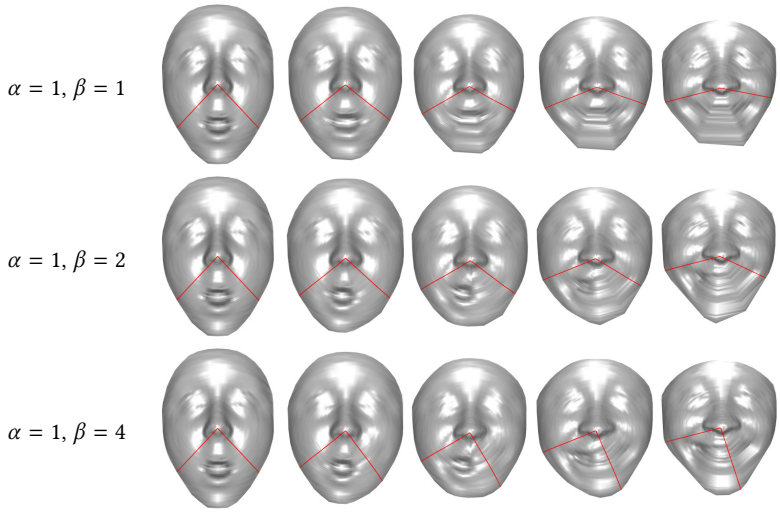


Fig. 7. Curve matching for a relatively large deformation under three different β values. Contrary to Figure 6, the value of β has a significant impact on the matching result since the deformation is relatively large, see supplementary video.

7.1 BU-3DFE experimental setup

BU-3DFE is composed of the six universal expressions with neutral face. The expressions are collected from 100 subjects (56 female and 44 male) of different race and age. Each expression has different levels of intensity, ranging from 1 to 4; the most intense one is labelled 4. In following subsections, we detail the experimental setup and the experimental results.

There are usually three main experimental scenarios that are performed using the BU-3DFE dataset for evaluating a modelling approach in expression recognition. The scenarios are summarized as follows:

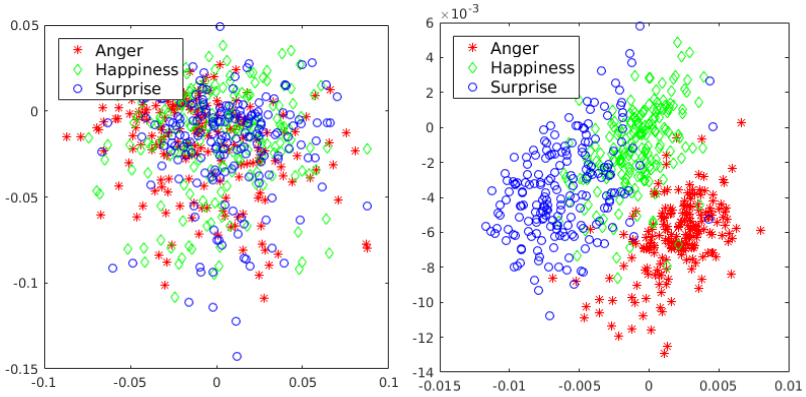
(a) $S = 0$ (b) $S = 0.5$ (c) $S = 1$ (d) $S = 2$

Fig. 8. Given a vector form of an expression $\mathcal{L}(\mathcal{D}_E)$, the figure shows the action of $\mathcal{E}(S \times \mathcal{L}(\mathcal{D}_E))$ on two different neutral faces for different scales S. In (a) the scale $S = 0$, thus the face remains neutral.

i. Select 60 subjects, split the dataset into 10 sections and use the 54 out of 60 for training and the remaining 6 for testing.
ii. Select 60 subjects, split the dataset into 10 sections and use the 54 out of 60 for training and the remaining 6 for testing. Repeat the experiment 100 times.
iii. Randomly select 60 subjects, split the dataset into 10 sections and use the 54 out of 60 for training and the remaining 6 for testing. Repeat the experiment 100 times. Except in this case the subjects are randomly select in each round.

In this paper, we are conducting experiment type (iii) 20 times instead of 100. Consequently, we will mainly compare our approach with methods evaluated with experimental type (ii) and experimental type (iii).

Given neutral faces of subjects, we duplicate the experimental scenario discussed in [Berretti et al. 2011]. We select 60 subjects randomly with all the 6 expressions in two intensities (3 and 4). Out of the 60 subjects 54 subjects are selected as a training data and the left out 6 are reserved for testing. The process is repeated 10 times by dissecting the 60 subjects into different training and testing groups, similar to 10 fold cross validation. Next, the overall process is repeated 100 times. In our case, we repeat the overall process 20 times. Hence, we perform, in total, 200 times training and testing. Once we randomly select 60 subjects for 20 times, we prepare the dataset under different parameters and curve sampling settings. In all facial surface representations, we fix the number of points representing curves $z = 50$. However, we prepare all facial surfaces for two different numbers of curve values, i.e., for $k = 50$ and $k = 100$. We call these datasets D-1 and D-2, respectively. Subsequently, for both datasets, D-1 and D-2, we select the k curves optimally, as described in Section 4, and uniformly as described in Section 2. Hence in total, we prepare 4 datasets, i.e., D-1-U, and D-1-O (D-1 with uniform and optimal curve sampling, respectively), and D-2-U and D-2-O (D-2 with uniform and optimal curve sampling, respectively). In the optimal sampling case, all neutral faces are optimally sampled to k curves with respect to a randomly selected and uniformly sampled neutral reference face. Next, faces with expression are optimally sampled with respect to their respective optimally sampled neutral faces. In all of the optimal sampling $\alpha = 1$ and $\beta = 0.7$. For every training and testing phase, SVM with linear kernel is trained on the Lie algebra, see Section 6. Classification is done on the Lie algebra in one-vs-all classification scenario.



(a) Expression residue (b) Our approach

Fig. 9. Two-dimensional expression space computed from three expressions with PCA. (a) PCA on expressions extracted as expression residues. (b) PCA on the Lie algebra of expressions extracted with our approach.

Expression representation	# Principal components	classifier	performance
$\mathbf{E}_{res}$	N/A	linear SVM	34.55%
$\mathbf{E}_{res}$	60	linear SVM	39.12%
$\mathbf{E}_{res}$	30	linear SVM	43.79%
$\mathbf{E}^*_{res}$	N/A	linear SVM	64.15%
$\mathbf{E}^*_{res}$	60	linear SVM	58.33%
$\mathbf{E}^*_{res}$	30	linear SVM	50.02%
$\mathcal{D}_E$	N/A	linear SVM	76.45%

Table 1. Comparision between Linear vs proposed expression representation. $\mathbf{E}_{res}$ represents expression residue from faces with ICP alignment only. $\mathbf{E}^*_{res}$ represents expression residue from faces decomposed into 50 uniformly sampled curves with 50 uniformly sampled points. $\mathcal{D}_E$ represents the decomposed expression using our approach.

7.2 Linear vs proposed expression space

Most methods, evaluated on the BU-3DFE dataset, are based on features that are extracted from annotated/estimated landmark regions. As a result, to demonstrate the representativeness of our approach, in comparison with landmark-free linear expression representation, we replicate a simple PCA based expression representation as described in [Al-Osaimi et al. 2009]. To build a PCA based expression space, we take facial surfaces that are aligned and parametrized as described in Section 2. Facial expressions are then decoupled from neutral faces by taking the point-to-point difference, which gives the expression residue. To that end, we will consider two types of alignment to compute the expression residues: First, we consider facial surfaces that are scaled and aligned with ICP. In such a case, for a neutral face Γ_N and a face with an expression Γ_E, the expression residue $E_{res} = \Gamma_E - \Gamma_N$. Secondly, we will consider facial surfaces that are scaled and aligned with ICP and *decomposed into uniformly sampled 50 curves and points*. In such a case, we denote the expression residue with $*$ as $E^*_{res} = \Gamma_E - \Gamma_N$. Subsequently, the expression space is estimated with a subspace spanned by different number of principal components computed from the expression residues with PCA. All expression residues are then projected on to the expression space where SVM is used, with linear kernel, to train and classify expressions in a similar experimental scenario as described in Section 7.1; a comparison of the average performance of the expression residues

is shown in Table 1. Note that the expression space estimation is done in each training and testing phase. In comparison, our approach performs much better, see Table 3. The main reason for such a large difference in performance is the Lie group based expression representation which disentangles different expressions when mapped to the Lie algebra. To illustrate this, we compute a two-dimensional, with PCA, expression space of *anger*, *happiness*, and *sadness* from the whole BU-3DFE dataset with 3 and 4 intensities. As shown in Fig. 9, expressions extracted with our approach are more separately clustered as compared to expression residues.

7.3 Results on BU-3DFE

In this subsection, we present results of our approach and existing expression recognition methods that follow the same experimental scenario as [Berretti et al. 2011]. As shown in Table 3, our approach outperformed [Berretti et al. 2011], when an SVM is trained on expressions extracted from D-2-O dataset, see Subsection 7.1. However, our approach performed worse than [Yang et al. 2015] and [Zhen et al. 2016]. Nevertheless, both approaches [Yang et al. 2015; Zhen et al. 2016] are based on localized features that are hierarchically decomposed [Yang et al. 2015] and hand crafted [Zhen et al. 2016]. Consequently, unlike our approach, it is not clear how one can invert computational results from the feature space, constructed using both methods, back to the data space. The lowest accuracy rate of our approach is on expressions extracted from D-1-U. However, accuracy rate improves as the number of curves is increased, see Table 4 and Table 6. This is mainly because a dense set of curves approximate the facial surface more closely. Consequently, subtle details of an expression are more likely to be captured from a dense set of curves than sparse. Additionally, optimal sampling of the curves improves performance, regardless of the curve number, see Table 5 and Table 7. As argued in Section 4, expressions extracted from optimally sampled curves are more representative than uniformly sampled ones. Compare Table 4 against Table 5, and Table 6 against Table 7. Nevertheless, in all our experiments, *fear* is largely confused with *hapiness*.

Regardless, however, almost all of the methods that we have compared against, Table 3, are based on geometric or appearance features. Meanwhile, our approach is based on curve based facial surface representation. Subsequently, to compare our approach with another facial surface based representation that resembles what is presented in this paper, we replicate the experimental scenario described in [Kurtek and Drira 2015]. Hence, we select the first 11 subjects, with six expression of the highest intensity, from the BU-3DFE dataset. The dataset is preprocessed similar to D-2-O, i.e., $k = 100$ optimally sampled curves that are approximated with $z = 50$ points. Similar to [Kurtek and Drira 2015], we perform a leave-one-subject-out experimental scenario with 1-nearest neighbour classifier. In effect, we only use the distance metric defined in (16) to classify expressions. The proposed approach achieved 65.15% recognition accuracy. While the method presented in [Kurtek and Drira 2015] achieved 62.12%, when optimal surface parametrization is not estimated, and 74.24% under optimal surface parametrization. Our approach performed lower than what is presented in [Kurtek and Drira 2015]. As indicated in the limitations of the curve correspondence estimation, we are estimating optimal curve correspondence and leaving the point sampling to uniform arc length sampling. Meanwhile, in [Kurtek and Drira 2015] the parametrization is estimated for the whole facial surface leading to high retrieval accuracy.

7.4 Bosphorus dataset

The Bosphorus dataset is composed of face and facial expression data collected from 105 subjects. Nevertheless, it is only the 65 subjects that have labelled six universal expressions. Furthermore, the dataset is composed of facial point clouds with erroneous measurements, especially in facial expressions where opening of the mouth is involved. In this

%	HA	SU	FE	SA	AN	DI
HA	**100.00**	0.00	0.00	0.00	0.00	0.00
SU	0.00	**75.00**	15.38	3.85	3.85	1.92
FE	7.69	39.62	**39.23**	3.85	7.69	1.92
SA	3.85	3.85	0.00	**58.08**	24.62	9.62
AN	3.85	0.00	0.00	15.38	**75.00**	5.77
DI	25.77	3.85	0.00	9.62	5.77	**55.00**
	Average= 67.05%					

Table 2. Confusion matrix on Bosphorus dataset.

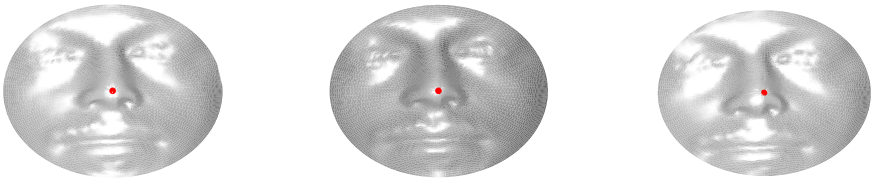
section, we evaluate the proposed approach on the Bosphorus dataset to assess its tolerance towards erroneous facial deformation measurements and tip of the nose estimation.

Subsequently, similar to [Taheri et al. 2014], we use a leave-one-subject-out experimental scenario to test the proposed approach on the 65 subjects with labelled 6 facial expressions. That is, in each iteration we use the 59 subjects, with all the six expressions, for training and use the left out one subject for testing. Similar to the evaluation on BU-3DFE, we train an SVM classifier on the Lie algebra of the representation space and use a one-vs-all classification scenario. Each facial surface is represented by 100 uniformly sampled curves, i.e., $k = 100$, that are approximated with 50 uniformly sampled points, i.e., $z = 50$. In other words, the dataset is preprocessed in a similar manner as D-2-U. The proposed approach achieved an average recognition rate of 67.05% accuracy. The confusion matrix, see Table 2, shows that the main difficulty of our approach, in the Bosphorus dataset, was in discerning *fear* from *surprise*. We believe the main accuracy difference between the Bosphorus and the BU-3DFE dataset is due to erroneous deformation measurements of 3D facial surfaces in the Bosphorus dataset. Since the proposed approach assumes a consistent 3D facial surfaces and deformations, erroneous outliers degrade its performance. Meanwhile, in [Taheri et al. 2014] an approach based on action unit composition, from an RGB input data, achieved an average 69.78% accuracy in a similar experimental protocol. Although our method and what is presented in [Taheri et al. 2014] are different approaches based on different input data, our method achieved a comparable result despite the erroneous point cloud measurements.

Meanwhile, the proposed approach also assumes a consistent estimation of tip of the nose to decompose a facial surface. As a result, we evaluate the error tolerance of the proposed approach on a dataset with noisy nose tip estimates. To that end, we add a noise vector drawn from a Gaussian distribution with zero mean vector and $\Sigma = 10 \times I_3$ covariance matrix, where I_3 represents a 3×3 identity matrix, to the estimated nose tip point. Note that the noise is randomly drawn each time a facial surface is processed, see Fig. 10. Subsequently, the leave-one-subject-out experiment on the Bosphorus dataset, with the noisy nose estimate, achieved an accuracy of 65.95%. Meanwhile, the same experimental scenario achieved 67.95% accuracy using the ground truth nose label that is provided along with the dataset. As described earlier, the performance with estimated nose tip, using (3), is 67.05%. Hence, the proposed approach is reasonably tolerant to erroneous tip of the nose estimations.

8 CONCLUSION

The paper introduced a new deformation-based facial expression representation. The representation is based on a mapping function that identifies a set of facial curves with an element of a high dimensional matrix Lie group. Furthermore, an algorithm for facial curve correspondence estimation is proposed. To validate the proposed representation, SVM is trained on the Lie algebra of the expression representation space. The results outperformed state of the art



(a) Ground truth. (b) Estimated using (3) (c) Estimated + Gaussian noise

Fig. 10. Illustration of estimated nose tip points, depicted with red point.

Methods	Landmarks	Expression features	Classifier	Performance		
				i	ii	iii
[Wang et al. 2006]	Yes	Primitive surface features	LDA	83.60%	61.79%	-
[Soyel and Demirel 2007]	Yes	Distance of landmarks	MLP	91.30%	67.52%	-
[Gong et al. 2009]	No	Region based depth difference	SVM	-	76.22%	-
[Berretti et al. 2010]	Yes	SIFT features	SVM	-	-	77.54%
[Berretti et al. 2011]	Yes	SIFT features	SVM	-	-	78.43%
[Zeng et al. 2013]	Yes	Mean curvature + Conformal factor	SRC	-	-	70.93%
[Li et al. 2015]	Yes	2D + 3D features	SVM	-	86.32%	-
[Yang et al. 2015]	No	Geometric feature based scattering representation	SVM	-	84.80%	82.73%
[Zhen et al. 2016]	No	Localized geometric features	SVM	-	84.5%	83.2%
Ours on D-1-U	No	No features	SVM	-	-	76.45%
Ours on D-1-O	No	No features	SVM	-	-	77.91%
Ours on D-2-U	No	No features	SVM	-	-	78.14%
Ours on D-2-O	No	No features	SVM	-	-	**79.16**%

Table 3. Comparison between performances of the proposed approach and recent results. We highlight the top score at the bottom.

methods evaluated on the BU-3DFE dataset. Nevertheless, there are areas where the approach can be improved. First, accurate estimation of a world coordinate system (alignment) impacts the performance of the proposed representation significantly, hence improving the coordinate alignment of a dataset is important, e.g., Semi-rigid ICP. Alternatively, a better rotational alignment can be achieved by alternatively optimizing curve correspondence estimation and rotational alignment until convergence. Second, improving the time complexity of the correspondence estimation algorithm is very important so that correspondence between faces can be computed without restriction, i.e., correspondence estimation for both curves and points. Such a solution can, theoretically, lead to dense point correspondence estimation between faces without the need for training dataset.

REFERENCES

F Al-Osaimi, Mohammed Bennamoun, and Ajmal Mian. 2009. An expression deformation approach to non-rigid 3D face recognition. *International Journal of Computer Vision* 81, 3 (2009), 302–316.

Djamila Aouada, Kassem Al Ismaeil, Kedija Kedir Idris, and Björn Ottersten. 2014. Surface up-sr for an improved face recognition using low resolution depth cameras. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*. IEEE, 107–112.

Marian Stewart Bartlett, Paul A Viola, Terrence J Sejnowski, Beatrice A Golomb, Jan Larsen, Joseph C Hager, and Paul Ekman. 1996. Classifying facial action. *Advances in neural information processing systems* (1996), 823–829.

Richard Bellman. 1956. Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences* 42, 10 (1956), 767–769.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

%	AN	HA	SU	FE	SA	DI
AN	**67.96**	1.17	0.58	1.50	18.67	10.12
HA	0.04	**95.92**	1.08	2.71	0.00	0.25
SU	0.00	1.04	**95.83**	1.25	1.42	0.46
FE	5.42	22.21	10.25	**47.75**	8.21	6.17
SA	12.21	2.04	2.58	3.79	**78.83**	0.54
DI	6.58	6.21	6.46	4.92	3.42	**72.42**
	Average= 76.45%					

Table 4. Confusion matrix on expressions extracted from faces with uniformly sampled 50 curves (D-1-U).

%	AN	HA	SU	FE	SA	DI
AN	**65.83**	0.83	0.00	3.33	19.17	10.83
HA	0.00	**95.00**	2.50	0.83	0.00	1.67
SU	0.00	0.83	**95.00**	1.67	1.67	0.83
FE	3.33	15.00	10.83	**56.67**	9.17	5.00
SA	13.33	0.83	2.50	3.33	**80.00**	0.00
DI	5.00	8.33	6.67	3.33	1.67	**75.00**
	Average= 77.91%					

Table 5. Confusion matrix on expressions extracted from faces with optimally sampled 50 curves (D-1-O).

%	AN	HA	SU	FE	SA	DI
AN	**65.83**	0.83	0.00	0.83	22.50	10.00
HA	0.00	**97.50**	1.67	0.83	0.00	0.00
SU	0.00	0.83	**94.17**	1.67	2.50	0.83
FE	5.83	19.24	10.83	**54.64**	4.17	5.00
SA	15.83	0.83	1.67	3.33	**77.50**	0.83
DI	4.17	7.50	3.33	3.33	2.50	**79.17**
	Average= 78,14%					

Table 6. Confusion matrix on expressions extracted from faces with uniformly sampled 100 curves (D-2-U).

%	AN	HA	SU	FE	SA	DI
AN	**72.50**	0.00	0.83	4.17	16.67	5.83
HA	0	**93.33**	0.83	5.00	0.00	0.83
SU	0.00	0.83	**95.83**	2.50	0.83	0.00
FE	5.00	23.33	5.83	**50.83**	7.50	7.50
SA	13.33	1.67	2.50	1.67	**80.83**	0.00
DI	3.33	5.00	5.83	1.67	2.50	**81.67**
	Average= 79,16%					

Table 7. Confusion matrix on expressions extracted from faces with optimally sampled 100 curves (D-2-O).

Stefano Berretti, Boulbaba Ben Amor, Mohamed Daoudi, and Alberto Del Bimbo. 2011. 3D facial expression recognition using SIFT descriptors of automatically detected keypoints. *The Visual Computer* 27, 11 (2011), 1021–1036.

Stefano Berretti, Alberto Del Bimbo, Pietro Pala, Boulbaba Ben Amor, and Mohamed Daoudi. 2010. A set of selected SIFT features for 3D facial expression recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 4125–4128.

Ya Chang, Changbo Hu, Rogerio Feris, and Matthew Turk. 2006. Manifold based analysis of facial expression. *Image and Vision Computing* 24, 6 (2006), 605–614.

Ya Chang, Marcelo Vieira, Matthew Turk, and Luiz Velho. 2005. Automatic 3D facial expression analysis in videos. In *International Workshop on Analysis and Modeling of Faces and Gestures*. Springer, 293–307.

Girum G Demisse, Djamila Aouada, and Bjorn Ottersten. 2015. Template-based statistical shape modelling on deformation space. In *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 4386–4390.

Girum G. Demisse, Djamila Aouada, and Bjorn Ottersten. 2016. Similarity Metric For Curved Shapes In Euclidean Space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Girum G Demisse, Djamila Aouada, and Bjorn Ottersten. 2017. Deformation Based Curved Shape Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

Hassen Drira, Boulbaba Ben Amor, Mohamed Daoudi, and Anuj Srivastava. 2010. Pose and expression-invariant 3d face recognition using elastic radial curves. In *British machine vision conference*. 1–11.

Hassen Drira, Boulbaba Ben Amor, Anurag Srivastava, Meroua Daoudi, and Rim Slama. 2013. 3D face recognition under expressions, occlusions, and pose variations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, 9 (2013), 2270–2283.

Ian L Dryden and Kanti V Mardia. 1998. *Statistical shape analysis*. Vol. 4. J. Wiley Chichester.

Paul Ekman and Wallace V Friesen. 1977. Facial action coding system. (1977).

Tianhong Fang, Xi Zhao, Omar Ocegueda, Shishir K Shah, and Ioannis A Kakadiaris. 2011. 3D facial expression recognition: A perspective on promises and challenges. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 603–610.

Boqing Gong, Yueming Wang, Jianzhuang Liu, and Xiaoou Tang. 2009. Automatic facial expression recognition on a single 3D face by exploring shape deformation. In *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 569–572.

Jihun Ham and Daniel D Lee. 2007. Separating pose and expression in face images: a manifold learning approach. *Neural Information Processing-Letters and Reviews* 11, 4 (2007), 91–100.

David G Kendall. 1984. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society* 16, 2 (1984), 81–121.

Sebastian Kurtek and Hassen Drira. 2015. A comprehensive statistical framework for elastic shape analysis of 3D faces. *Computers & Graphics* 51 (2015), 52–59.

Huibin Li, Huaxiong Ding, Di Huang, Yunhong Wang, Xi Zhao, Jean-Marie Morvan, and Liming Chen. 2015. An efficient multimodal 2D+ 3D feature-based approach to automatic facial expression recognition. *Computer Vision and Image Understanding* 140 (2015), 83–92.

Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. 2014. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1805–1812.

Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 5188–5196.

Andrea CG Mennucci. 2013. Metrics of curves in shape optimization and analysis. In *Level Set and PDE Based Reconstruction Methods in Imaging*. Springer, 205–319.

Peter W Michor, David Mumford, Jayant Shah, and Laurent Younes. 2007. A metric on shape space with explicit geodesics. *arXiv preprint arXiv:0706.4299* (2007).

Iordanis Mpiperis, Sotiris Malassiotis, and Michael G Strintzis. 2008. Bilinear elastically deformable models with application to 3d face and facial expression recognition. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*. IEEE, 1–8.

Maja Pantic. 2009. Machine analysis of facial behaviour: Naturalistic and dynamic behaviour. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 364, 1535 (2009), 3505–3513.

Maja Pantic and Marian Stewart Bartlett. 2007. *Machine analysis of facial expressions*. I-Tech Education and Publishing.

Marc Aurelio Ranzato, Joshua Susskind, Volodymyr Mnih, and Geoffrey Hinton. 2011. On deep generative models with applications to recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2857–2864.

Chafik Samir, Anuj Srivastava, and Mohamed Daoudi. 2006. Three-dimensional face recognition using shapes of facial curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, 11 (2006), 1858–1863.

Chafik Samir, Anuj Srivastava, Mohamed Daoudi, and Eric Klassen. 2009. An intrinsic framework for analysis of facial surfaces. *International Journal of Computer Vision* 82, 1 (2009), 80–95.

Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. 2008. Bosphorus database for 3D face analysis. *Biometrics and identity management* (2008), 47–56.

Hamit Soyel and Hasan Demirel. 2007. Facial expression recognition using 3D facial feature distances. In *International Conference Image Analysis and Recognition*. Springer, 831–838.

Anuj Srivastava, Eric Klassen, Shantanu H Joshi, and Ian H Jermyn. 2011. Shape analysis of elastic curves in euclidean spaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33, 7 (2011), 1415–1428.

Sima Taheri, Qiang Qiu, and Rama Chellappa. 2014. Structure-preserving sparse decomposition for facial expression analysis. *IEEE Transactions on Image Processing* 23, 8 (2014), 3590–3603.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1701–1708.

Joshua B Tenenbaum and William T Freeman. 2000. Separating style and content with bilinear models. *Neural computation* 12, 6 (2000), 1247–1283.

Veeravalli Seshadri Varadarajan. 2013. *Lie groups, Lie algebras, and their representations*. Vol. 102. Springer Science & Business Media.

Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. 2013. Hoggles: Visualizing object detection features. In *Proceedings of the IEEE International Conference on Computer Vision*. 1–8.

Jun Wang, Lijun Yin, Xiaozhou Wei, and Yi Sun. 2006. 3D facial expression recognition based on primitive surface feature distribution. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. IEEE, 1399–1406.

Yang Wang, Xiaolei Huang, Chan-Su Lee, Song Zhang, Zhiguo Li, Dimitris Samaras, Dimitris Metaxas, Ahmed Elgammal, and Peisen Huang. 2004. High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 677–686.

Xudong Yang, Di Huang, Yunhong Wang, and Liming Chen. 2015. Automatic 3d facial expression recognition using geometric scattering representation. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, Vol. 1. IEEE, 1–6.

Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. 2008. A high-resolution 3D dynamic facial expression database. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*. IEEE, 1–6.

Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. 2006. A 3D facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*. IEEE, 211–216.

Laurent Younes. 1998. Computable elastic distances between shapes. *SIAM J. Appl. Math.* 58, 2 (1998), 565–586.

Wei Zeng, Huibin Li, Liming Chen, Jean-Marie Morvan, and Xianfeng David Gu. 2013. An automatic 3D expression recognition framework based on sparse representation of conformal images. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE, 1–8.

Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. 2009. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, 1 (2009), 39–58.

Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. 2014. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing* 32, 10 (2014), 692–706.

Yongmian Zhang and Qiang Ji. 2005. Active and dynamic information fusion for facial expression understanding from image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, 5 (2005), 699–714.

Qingkai Zhen, Di Huang, Yunhong Wang, and Liming Chen. 2016. Muscular Movement Model-Based Automatic 3D/4D Facial Expression Recognition. *IEEE Transactions on Multimedia* 18, 7 (2016), 1438–1450.