3D/4D facial expression analysis: An advanced annotated face model approach

Tianhong Fang, Xi Zhao, Omar Ocegueda, Shishir K. Shah, Ioannis A. Kakadiaris

Computational Biomedicine Lab, Department of Computer Science, University of Houston, Houston, TX 77004, USA

Abstract
Facial expression analysis has interested many researchers in the past decade due to its potential applications in various fields such as human–computer interaction, psychological studies, and facial animation. Three-dimensional facial data has been proven to be insensitive to illumination condition and head pose, and has hence gathered attention in recent years. In this paper, we focus on discrete expression classification using 3D data from the human face. The paper is divided in two parts. In the first part, we present improvement to the fitting of the Annotated Face Model (AFM) so that a dense point correspondence can be found in terms of both position and semantics among static 3D face scans or frames in 3D face sequences. Then, an expression recognition framework on static 3D images is presented. It is based on a Point Distribution Model (PDM) which can be built on different features. In the second part of this article, a systematic pipeline that operates on dynamic 3D sequences (4D datasets or 3D videos) is proposed and alternative modules are investigated as a comparative study. We evaluated both 3D and 4D Facial Expression Recognition pipelines on two publicly available facial expression databases and obtained promising results.

1. Introduction
Facial expression analysis/recognition (FER) has interested many researchers due to its various purposes and applications. It plays a key role in emotion recognition and can thus, contribute to the development of human–computer interaction systems. It can also be used to improve the performance of face recognition systems by providing prior knowledge on the facial motions and facial feature deformations. This is particularly intriguing considering that the mouth area contains significant amount of discriminative information [1], and yet, is the region where most of the facial deformations occur. Other applications of FER include, but are not limited to, psychological studies, tiredness detection, facial animation, robotics, and virtual reality.

Facial expressions are generated by facial muscle contractions which result in temporary facial deformations in both facial geometry and texture. The previous studies have focused primarily on the 2D domain due to the prevalence of data in the relevant modalities (i.e., images and videos). Comprehensive surveys in this area include those by Fasel and Luettin [2], Pantic et al. [3], and Zeng et al. [4]. While these 2D facial expression recognition systems have achieved remarkable performance, challenges in 2D face recognition still present themselves in 2D expression analysis (i.e., illumination and pose variations). Three-dimensional (3D) data, on the other hand, are invariant to such changes and are information-rich by nature. Recent successes in 3D face recognition [5,6] can naturally be exploited for expression recognition.

Since the public release of the BU-3DFE database [7], research on 3D FER has drawn considerable attention, as presented in a survey by Fang et al. [8]. The BU-3DFE database has become the de facto testbed where FER researchers benchmark their algorithms and hence, we can expect more effort in that direction. Furthermore, the BU-4DFE database [9] that was released two years later, contains dynamic 3D sequences and introduces the temporal expression evolution in the 3D domain. However, only limited work has been done on 4D FER despite the significance of temporal cues in facial expression studies.

The Annotated Face Model (AFM) has been successfully applied to 3D face recognition. However, fitting an AFM on a face with expression has encountered difficulties. To overcome these problems, we have presented a two-step approach to improve the fitting of AFM for FER purpose. First, the Thin Plate Splines (TPS) and AFM are combined to find the dense point correspondence among 3D faces in terms of both location and semantics. Second, a key point selection (Spin image or MeshHOG) and filtering (RANSAC) mechanism are introduced to expedite the registration of expressive 3D facial scans in a sequence. A 3D FER algorithm that uses component-based Point Distribution Model (PDM) is then introduced. Taking advantage of the dense point correspondence from AFM, sophisticated 3D PDMS consisting of up to $10^3$ points can be built. The rich shape information contained in the PDM models enables the perception of subtle shape deformations from...
expression. In addition, thanks to the semantic annotation of the AFM, we are able to divide a face scan into different components and analyze each component separately. This overcomes the intervention among different components in learning the most discriminative variation modes of expressions using PCA. The intervention always occurs when the variation modes are learnt globally from the whole face, like in the case of Active Appearance Model. In the second part of the paper, we present a dense registration of 4D data which preserves the temporal coherence and demonstrates its effectiveness on 4D FER using a variant of local binary patterns on three orthogonal planes (LBP-TOP) [10]. The flow matrix is computed as the difference between the current frame with either the neutral scan or the previous frame. Since the registration between 3D frames compensates for the pose differences, the flow matrix contains mostly the information of the expression variations. Then, histogram on LBP-TOP is applied as a descriptor for recognition. Overall, our main contributions are: (i) a component-based analysis for 3D FER, (ii) registration methods designed specifically for expressive 3D/4D data, and (iii) an adaptation of LBP-TOP for 4D FER.

The rest of the paper is structured as follows: an extensive review of previous work in 3D/4D FER is presented in Section 2. The pipelines for 3D and 4D FER are presented in Sections 3 and 4, respectively. In Section 5, we evaluate the performance of both pipelines on public available databases and compare their performance with recent methods in this area. Finally, conclusions and future work are presented in Section 6.

2. Literature review

Since the public release of the BU-3DFE database [7], considerable effort has been devoted to solve the problem of 3D FER. On the other hand, only limited work has been done to incorporate the temporal information that is either embedded in the 4D data or interpolated from 3D key-frames. We will briefly discuss the work in 3D FER and mention all the methods from groups that are more active in 3D FER. We further enumerate all the methods that have been introduced in 4D FER. Existing approaches in 3D/4D FER and their key properties are summarized in Table 1. Readers are encouraged to read the survey by Fang et al. [8] for a more comprehensive overview.

2.1. 3D FER

Wang et al. [12] used cubic-order polynomial functions to approximate the continuous surface at each vertex of the input mesh [40]. The estimated coefficients of the polynomial function at a particular vertex \( \mathbf{v} \) form the Weingarten matrix for the local surface patch. The eigenvalues and eigenvectors of this matrix, along with the gradient magnitude that can be derived by the normal direction at \( \mathbf{v} \), form a feature set that is used to assign to \( \mathbf{v} \) a unique primitive 3D surface label based on a set of classification rules. To overcome the lack of correspondence between the meshes, the authors defined seven facial regions using 64 facial landmarks. The histograms of surface labels are computed for each region and normalized by the number of vertices in the region. This treatment introduces a sense of correspondence that facilitates the subsequent classification step, where the best performance is obtained using Linear Discriminant Analysis (LDA). Note that due to the geometrical invariance of curvature-based features, no rigid transformation is required to achieve this correspondence.

Soyel and Demirel [15] selected six distance measures among a pool of landmarks that maximize the differences of facial expressions to form the feature vectors. The intuition behind this selection comes from the definition of the fundamental facial expressions by the MPEG-4 facial animation parameters (FAPs) [41]. The authors argued that among all 84 feature points specified by MPEG-4, only a small set are not static due to the contraction and expansion of facial muscles when one of the universal expressions is displayed. However, the authors did not specify how to identify this set of feature points. By utilizing facial symmetry, they were able to trim down the number of facial features to merely 11 points, from which the six characteristic distances are extracted. One of the distances, which is essentially

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Chang [11]</td>
<td>N</td>
<td>Y</td>
<td>22 semi-auto</td>
<td>Y</td>
<td>Private</td>
<td>6</td>
<td>N/A</td>
</tr>
<tr>
<td>Wang [12]</td>
<td>N</td>
<td>N</td>
<td>64 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>83.6</td>
</tr>
<tr>
<td>Yin [13]</td>
<td>Y</td>
<td>N</td>
<td>64 semi-auto</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>80.2</td>
</tr>
<tr>
<td>Ramanathan [14]</td>
<td>N</td>
<td>N</td>
<td>11 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>7</td>
<td>91.3</td>
</tr>
<tr>
<td>Soyle [15]</td>
<td>N</td>
<td>N</td>
<td>58 auto</td>
<td>Y</td>
<td>Private</td>
<td>4</td>
<td>83.0</td>
</tr>
<tr>
<td>Sun [17]</td>
<td>Y</td>
<td>Y</td>
<td>83 auto</td>
<td>Y</td>
<td>BU-4DFE</td>
<td>6/8 AUs</td>
<td>80.9/87.1</td>
</tr>
<tr>
<td>Sun [18]</td>
<td>Y</td>
<td>Y</td>
<td>83 auto</td>
<td>Y</td>
<td>BU-4DFE</td>
<td>6</td>
<td>90.4</td>
</tr>
<tr>
<td>Mipiperis [19]</td>
<td>N</td>
<td>N</td>
<td>83 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>90.5</td>
</tr>
<tr>
<td>Mipiperis [20]</td>
<td>N</td>
<td>N</td>
<td>83 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>92.3</td>
</tr>
<tr>
<td>Soyle [21]</td>
<td>N</td>
<td>N</td>
<td>27 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>7</td>
<td>87.8</td>
</tr>
<tr>
<td>Tang [22,23]</td>
<td>N</td>
<td>N</td>
<td>83 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>95.1, 87.1</td>
</tr>
<tr>
<td>Rosato [24]</td>
<td>Y</td>
<td>Y</td>
<td>22 auto</td>
<td>Y</td>
<td>BU-3DFE BU-4DFE</td>
<td>7/6</td>
<td>80.1/85.9</td>
</tr>
<tr>
<td>Venkatseths [25]</td>
<td>Y</td>
<td>N</td>
<td>68 auto</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>81.7</td>
</tr>
<tr>
<td>Soyle [26,27]</td>
<td>N</td>
<td>N</td>
<td>83 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>7</td>
<td>93.7</td>
</tr>
<tr>
<td>Gongs [28]</td>
<td>N</td>
<td>N</td>
<td>83 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>76.2</td>
</tr>
<tr>
<td>Tekguc [29]</td>
<td>N</td>
<td>N</td>
<td>83 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>7</td>
<td>88.2</td>
</tr>
<tr>
<td>Savran [30]</td>
<td>N</td>
<td>N</td>
<td>Auto</td>
<td>Y</td>
<td>Bosphorus</td>
<td>22 AUs</td>
<td>91.4</td>
</tr>
<tr>
<td>Zhao [31]</td>
<td>Y</td>
<td>N</td>
<td>19 manual</td>
<td>Y</td>
<td>Bosphorus</td>
<td>7/16 AUs</td>
<td>94.2/85.6</td>
</tr>
<tr>
<td>Savran [32]</td>
<td>Y</td>
<td>N</td>
<td>Auto</td>
<td>Y</td>
<td>Bosphorus</td>
<td>25 AUs</td>
<td>97.1</td>
</tr>
<tr>
<td>Berretti [33]</td>
<td>N</td>
<td>N</td>
<td>27 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>77.5</td>
</tr>
<tr>
<td>Zhao [34]</td>
<td>Y</td>
<td>N</td>
<td>19 auto</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>82.3</td>
</tr>
<tr>
<td>Maalej [35]</td>
<td>N</td>
<td>N</td>
<td>24 manual</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>N/A</td>
</tr>
<tr>
<td>Venkatseths [36]</td>
<td>Y</td>
<td>N</td>
<td>Not used</td>
<td>Y</td>
<td>BU-3DFE</td>
<td>6</td>
<td>85.6</td>
</tr>
<tr>
<td>Tsalakianidou [37]</td>
<td>Y</td>
<td>Y</td>
<td>81 auto</td>
<td>N</td>
<td>Private</td>
<td>5/11 AUs</td>
<td>85.0/83.6</td>
</tr>
<tr>
<td>Sandbach [38]</td>
<td>N</td>
<td>Y</td>
<td>Not used</td>
<td>Y</td>
<td>BU-4DFE</td>
<td>3</td>
<td>81.9</td>
</tr>
<tr>
<td>Le [39]</td>
<td>N</td>
<td>Y</td>
<td>Not used</td>
<td>Y</td>
<td>BU-4DFE</td>
<td>3</td>
<td>92.2</td>
</tr>
</tbody>
</table>

The rest of the paper is structured as follows: an extensive review of previous work in 3D/4D FER is presented in Section 2. The pipelines for 3D and 4D FER are presented in Sections 3 and 4, respectively. In Section 5, we evaluate the performance of both pipelines on public available databases and compare their performance with recent methods in this area. Finally, conclusions and future work are presented in Section 6.
the width of the face contour, is used to normalize other distances as an attempt to make the feature scale-invariant. Subsequently, a neural network, trained using the backpropagation algorithm, is used to classify the expressions. Note that this distance-based feature is also invariant against rigid transformations. In a follow-up work [21], the authors used a similar framework based on the FACS, but, the set of feature points are different and the distances are computed using several feature points instead of just two, which in the authors' view cancels out individual variations. This work was extended further in [26] and [27], where an automatic feature selection mechanism was introduced. Distances between all possible pairs of the 83 manual annotations of the BU-3DFE database are enumerated and normalized. Principal Component Analysis (PCA) is performed on this feature space to reduce its dimensionality and LDA is then applied to find the optimal subspace that preserves the most discriminant information. Realizing that some expression classes are close to each other in the subspace, the authors proposed to re-group these classes into clusters and perform the subspace projection followed by neural network classification in a hierarchical manner. Under the same framework, but without the coarse-to-fine scheme, Tekguc et al. [29] adopted the Non-dominated Sorted Genetic Algorithm II for feature selection and obtained a slightly lower recognition rate.

Tang and Huang [22] explored similar distance features. They proposed an automatic feature selection method based on maximizing the average relative entropy of marginalized class-conditional feature distributions. Less than 30 “best” features are automatically selected using this method from the pool of all possible line segments between the 83 landmarks. A regularized AdaBoost algorithm, with three weak classifiers (i.e., Nearest Neighbor (NN), Naive Bayes (NB) and LDA), is used for classification. As a preprocessing step, the feature distances on the neutral scan of a subject are subtracted from the features of the subject’s expressive scan. In another work [23], they adopted a manual approach and carefully devised a set of 96 discriminative features that included not only the normalized distances, but also the slopes of the line segments connecting a subset of the 83 landmarks. The distances are normalized by the corresponding facial animation parameter units (FAPUs), which are used to scale the FAPs according to the MPEG-4 standard [41], and are defined as fractions of distances between certain feature points on a face model in its neutral state. This also implies the availability of the neutral scan of the input subject. In addition, the slope features are also normalized to unit vectors. A multi-class Support Vector Machine (SVM) using the one-against-one scheme is selected for classification.

2.2. 4D FER

Chang et al. [11] built a coarse mesh model and manually fitted this model to the initial frame of the range data. A 2D tracker was then employed and the model’s projection was warped by the 22 tracked feature points. The depth of the vertex was recovered by minimizing the distance between the model and the range data. The generalized expression manifold was built on the facial deformable model approach, an input 3D face scan is represented in a uv-parameterized space by an elastically deformed Annotated Face Parameterized space by an elastically deformed Annotated Face Model (AFPM) [42]. FER was formulated as the estimation of the posterior probability for each expression category.

Yin et al. [13] extended the work of Wang et al. [12] by introducing a tracking model for estimating motion trajectories, which are used to construct a spatial-temporal descriptor. They proposed a facial expression label map (FELM) based tracking approach. In this approach, the tracking model is first aligned to the 3D face scan, and then deformed to fit the target scan by minimizing an energy function. To create the sequential models from the BU-3DFE database, 40 intermediate frames are generated using the key-frame interpolation and synthesis approach based on the four models corresponding to the four intensity levels. The FELM vector and the motion vector are concatenated to form the descriptor, which becomes the input to LDA classifier. Rosato et al. [24] took an alternative approach to Yin et al. [13] for feature tracking with a generic model by automatically establishing vertex correspondences across input scans or dynamic sequences. A deformable template approach [43] was applied to extract 22 feature points on the 2D face texture. The 3D meshes were parameterized in a 2D plane by the circle pattern approach [44]. The proposed coarse-to-fine model adaptation approach between the planar representations was used and the correspondences are extrapolated back to the 3D meshes. The composition of the descriptor and the classifier are the same as in Yin et al. [13].

Sun et al. [17] used an Active Appearance Model (AAM) to track feature points in the 2D texture frames and retrieve their 3D positions. The influence of rigid head motion was eliminated by registering each 3D frame to the initial neutral scan and displacement of the tracked points between the two was used as the feature vector. The statistical information and the temporal dynamics of the training data were learned by Hidden Markov Models (HMM) and the Bayesian decision rule was used to classify query sequences given the trained models for the prototypical expressions. This approach was further enhanced in [18]. After the 3D positions of the feature points were identified, radial basis function (RBF) based interpolation was used to adapt the generic model to these feature points. Similar to [12], geometric surface label maps were generated from the adapted models. LDA was used to achieve optimal feature space transformation and the performance of various HMM-based classifiers is evaluated.

Tsalakoudou and Malassiotis [37] presented a fully automatic FER system which is capable of operating at 4–10 frames per second utilizing both 2D and 3D data acquired from a real-time 3D scanner. They built an Active Shape Model (ASM), which is a PDM learned from 81 manually annotated 3D landmarks with a 2D gradient profile for each landmark. Given a new 2D–3D image pair, they fitted the ASM to the data using the gradient information in the neighborhood of each landmark. The feature vectors combine geometric information of the landmarks and the statistics on the density of edges and curvature around the landmarks. For classification, a specific set of rules is defined for each expression, based on the variation of each component of the feature vector w.r.t. a base feature vector computed from the data of a particular subject with neutral expression. Despite its subject-dependent nature, this is the first fully automatic, real-time 2D-3D FER system reported in the literature.

Sandbach et al. [38] modeled an expressive sequence to contain neutral, onset, apex and offset segments. They captured Free-Form Deformations (FFDs) between frames and extracted features by applying a quad-tree decomposition. The features are then collected using the GentleBoost selection method and the temporal dynamics are modeled using HMMs.

Le et al. [39] proposed to use facial level curves for 3D face analysis. The distances between the level curves form the spatio-temporal features for 4D FER. A decision boundary focus classification algorithm is presented based on universal background modeling and maximum a posteriori (MAP) adaptation for HMMs.

Compared to the previous 3D FER methods, the proposed framework is more extensible and requires only 12 landmarks, which are more likely to be automatically detected. Compared the existing 4D FER approaches, our system is fully automatic and subject-independent. Furthermore, we do not assume the availability and usability of the associated 2D texture images since only shape modality is used in our pipeline. We demonstrate the effectiveness of our systems by evaluating performance on the publicly available BU-3DFE and BU-4DFE databases.

3. 3D facial expression recognition

In this section, we present a novel and extensible framework for facial expression recognition using 3D data. Through an improved deformable model approach, an input 3D face scan is represented in a uv-parameterized space by an elastically deformed Annotated Face...
Model (AFM). Not only the resulting meshes are already in a dense correspondence, they are also smooth and semantically consistent. This information-rich representation greatly facilitates subsequent operations such as feature extraction and classification.

3.1. The annotated deformable model framework

The deformable model approach proposed by Kakadiaris et al. [5] has been proven robust and adaptive for face recognition purposes. We opt to use the subdivision surfaces as the model, since it offers greater flexibility and scalability than parametric surfaces. First, the input 3D scan is aligned with the AFM (Fig. 1) using Iterative Closest Point (ICP) [45] followed by Enhanced Simulated Annealing (ESA) [46] to minimize the Z-buffer difference [47]. After this rigid registration step, the AFM is subdivided to fit the input scan by solving an analytical formulation defined by [48]:

$$\mathbf{M} \frac{\partial^2 \mathbf{q}}{\partial t^2} + \mathbf{D} \frac{\partial \mathbf{q}}{\partial t} + \mathbf{K} \mathbf{q} = \mathbf{f}_q,$$

(1)

which takes into consideration both the internal elastic force of the model (defined by the mass matrix $\mathbf{M}$, the damping matrix $\mathbf{D}$ and stiffness matrix $\mathbf{K}$) and the external pulling force ($\mathbf{f}_q$) from the target shape. This equation is solved using a Finite Element Method approximation. The output is a 3D mesh with consistent topology that resembles the target shape while providing a mapping to a regular 2D grid. As a result, different 3D facial meshes can be brought into correspondence and a direct comparison is made possible. Different classification tasks can be carried out using this uniform representation. In fact, this framework has already been leveraged for gender and ethnicity classification using 3D data [49,50].

3.2. Improved alignment using Procrustes analysis

Although ICP algorithm is robust to some degree of pose variations, it is sensitive to initialization. When the input scan exhibits very strong pose variation, ICP may yield local minima from which ESA cannot recover and hence, the rigid registration will fail. This is illustrated in the middle column of Fig. 2.

To compensate for the sensitivity to initialization, we leverage the registration framework of Passalis et al. [51] which adds a pre-alignment stage using Procrustes analysis based on landmark correspondence between the input scan and the AFM. In that study, 12 manually annotated landmarks are used, as illustrated in Fig. 1. We also notice that preliminary work has been done to automatically detect landmarks on the 3D facial scans [52–54].

In general, Procrustes analysis is a form of statistical shape analysis used to analyze the distribution of a set of shapes [55,56]. It is commonly used to compute the mean shape of a set of point-defined shapes. In our case, it is easily adapted to estimate a rough rigid transformation from the input scan to the AFM using the corresponding landmarks. Applying Procrustes analysis is more robust against extreme poses when compared to applying ICP & ESA directly, as shown in the right column of Fig. 2. An additional benefit is that the time required for registration is significantly reduced as the input scan is already very close to the AFM.

3.3. Improved model fitting using thin plate splines

After the aforementioned rigid transformation, the AFM undergoes a nonrigid deformation to match the shape of the input data thus capturing the geometric characteristics of the subject’s face. During the original fitting process developed by Kakadiaris et al. [5], the polygonal data of the input facial scan attracts the vertices of the AFM to drive the deformation. Thus, for each AFM vertex it is necessary to locate its nearest neighbor on the input based on which this external driving force is computed. This scheme works sufficiently well in cases where the input facial scan exhibits neutral or slight expression. However, when the input scan contains an excessive amount of deformation w.r.t. its neutral state, the closest neighbors of some AFM vertices are most likely not their true semantic correspondences. Without loss of generality, consider the problem of establishing point-wise correspondence between two curves, as the case with two surfaces would be essentially the same. While the semantic correspondences may coincide with the nearest neighbors in regions with similar curvature, in regions with large difference in curvature it is obvious that using only the shortest distance from one curve to the other does not guarantee an anatomically accurate correspondence.

Due to this deficiency, the original deformation framework is not directly applicable for fitting input scans with high intensity expressions. More specifically, it means that although the shape of the fitted model may be visually correct it will not have semantically correct regions (i.e., some of the vertices in the mouth area will be falsely identified as part of the chin) (Fig. 3).

One solution to this problem is to use some constraints to guide the deformation, which can be achieved through a set of corresponding landmarks on both the input scan and the AFM. As mentioned in Section 3.2, for the purpose of this study we use manually annotated landmarks (Fig. 1) but there are already some efforts devoted to the automatic landmark detection on 3D facial scans [52,53] and hence, it is not unrealistic to assume that the set of landmarks can be detected automatically in the near future.

Schneider et al. [57,58] presented a method for registering two 3D face meshes using the Thin Plate Spline (TPS) warping followed by a re-sampling process. The TPS formalization yields a transformation that minimizes a physically interpretable bending energy, which essentially solves a three-dimensional scattered data interpolation problem. In the re-sampling process, each vertex of the reference mesh is projected onto the target mesh by intersecting its normal with the target surface. This method can easily be adapted for mesh fitting by using the AFM as the reference mesh. Nevertheless, the main drawback of this method is that the re-sampled mesh is not

![Fig. 1. Definitions of the 12 landmarks used in this work: (1) outer corner of the right eye, (2) inner corner of the right eye, (3) inner corner of the left eye, (4) outer corner of the left eye, (5) nose tip, (6) right ala, (7) left ala, (8) midpoint of the upper lip, (9) right corner of the mouth, (10) left corner of the mouth, (11) midpoint of the lower lip, and (12) chin tip. Six regions are defined on the AFM: forehead, eyes and brows, nose, cheeks, mouth, and chin.](image)
smooth and it requires a post-processing step to fix the regions where the projections of the reference vertices do not exist, as illustrated in Fig. 4.

To overcome the drawbacks in both approaches we propose to substitute the vertex re-sampling with the deformation of the AFM, after the TPS-based warping has been applied. In other words, the warping serves as an initialization to the fitting algorithm. The warping is constrained by the landmarks and hence, it is able to deform the model to roughly match the target surface. Following this, the elastic deformation will take over and drive the vertices of the AFM toward the target, while maintaining its overall smoothness. In the meshes which have open mouth, we first mask out the internal points provided the landmarks of mouth corner and then perform the warping and fitting.

For conciseness, the complete fitting pipeline is presented in Algorithm 1, along with the improved registration steps described in Section 3.2. Fig. 5 illustrates a few examples of the AFM fitted to datasets with high-intensity expressions from BU-3DFE [7]. Note that the proposed pipeline is able to handle the extreme expressions very well, producing smooth, yet semantically correct regularized meshes.

Algorithm 1. Thin Plate Spline guided surface fitting

Input: A 3D facial scan with \( K = 12 \) annotated landmarks \( y \) and the AFM with the corresponding landmarks \( x \)

1. Estimate a rough rigid transformation from the data to the AFM by Procrustes analysis using the corresponding landmarks
   a. Recenter both \( y \) and \( x \) so that their centroids are at the origin \((0,0,0)\)
   b. Estimate the optimal rotation \( R_1 \) to align \( y \) to \( x \) by minimizing the Procrustes distance between the two \(|y - x|^2\)
   c. Compute the difference between the original centers of \( y \) and \( x \) for the translation matrix \( T_1 \)
   d. Transform the input scan by \( R_1 \times T_1 \)
2. Refine the transformation using ICP and Simulated Annealing
3. Warp the AFM to roughly match the input scan, constrained by the corresponding landmarks [57]
   a. Compute the displacement vectors \( d = y - x \)

---

**Fig. 2.** (L) Initial positions of the input scan w.r.t. the AFM; (M) Rigid registration using only ICP & ESA; (R) Rigid registration using ICP & ESA with initialization from Procrustes analysis.

**Fig. 3.** (L) Fitting results using the deformable model approach proposed by Kakadiaris et al. [5]. The method fails on 10.3% of BU3D scans, mostly on those with open mouth (R) Fitting results from the improved procedure using Thin Plate Spline warping as an initialization, followed by AFM deformation. The method succeeds on fitting to all BU3D scans, with a requirement of 12 landmarks.

**Fig. 4.** Schneider’s framework [57,58] requires a restoration step to fix unreliable vertices. (L) Depiction of the input data, and (R) depiction of the re-sampled reference mesh.

**Fig. 5.** A few examples of the AFM fitted to datasets with high-intensity expressions from BU-3DFE [7].
b. Compute the weight matrix for the mesh transform

\[ W = S^{-1}[d_1 \ldots d_K 0_{3 \times 1}]^T \]

where \( S \) is the Thin Plate Spline matrix
c. Transform each vertex \( v \) on the AFM by

\[ v' = v + W^T [u_1 \ldots u_K 1]^T \]

with \( u_i = ||v - x_i|| \).

4. Refine the deformation of the AFM by iteratively subdivision and solving the analytical formulation (Eq. (1))

Output: Deformed AFM that resembles the input scan

3.4. Feature representation

After fitting the AFM to the input data with semantic correctness, we now have a dense correspondence between each of the fitted meshes. To extract compact features from the meshes, we propose to build point distribution models (PDM) for each of the 6 prototypical expressions. The features are not necessarily computed in 3D space (e.g., positions, normals, curvatures) but can also be computed in 2D space (e.g., wavelets) since each of the fitted mesh has a regularized grid representation or UV parameterization in \( R^2 \). However, these features should be continuous for building the PDM.

In order to learn the most discriminate variation modes in different face regions, we define different regions on the face. Currently, the six facial regions that have already been defined on the AFM (denoted by different colors in Fig. 1) are used. However, our ongoing work investigates automatic segmentation of the face mesh based on a given classification criterion [59]. Based on the region definitions, the mesh is divided into six components. The nose region is discarded as it is the most rigid part of the human face and is invariant to facial expressions.

In the training phase, the collection of each component is grouped by expressions. For each of the groups, we align all shapes again using Procrustes analysis, but apply additional scale normalization to minimize inter-personal differences, which we do not want the PDMs to capture. We then build \( m \times n \) PDMs, where \( m \) is the number of regions used while \( n \) is the number of expressions. When we receive a new mesh for classification, each of its components is re-aligned in the same way with the means of the PDMs of the corresponding region but different expressions. This is followed by a subspace projection onto each of those PDMs. The coefficients from all projections are concatenated together to form a feature descriptor.

The overall feature generation procedure using this component-based framework is presented in Algorithms 2 and 3 for training and testing phases, respectively, where the input meshes are expected to be the results from fitting process described in Section 3.3. Note that the framework we have described so far can be used with any feature in the continuous space, but the re-aligning step requires at least one set of PDMs to be built using the shape geometry.

Algorithm 2. Training

Input: \( m \) sets of facial component shapes \( A \), each separated into \( n \) groups

1. Re-align the shapes in each group using Procrustes analysis with scaling such that \( ||A_i|| \) is equal to 1
2. Compute features \( F \) on the aligned data
3. Build a PDM using the shape geometry by applying PCA to group \( n_i \) from region \( m_j \) and obtain \( U_j^0 \) as the mean and \( V_j^0 \) as the principal components
4. Build other PDMs for features \( F \) and obtain \( \mathbf{U}^F_i \) as the mean and \( \mathbf{V}^F_i \) as the principal components.
5. Project the features \( F \) onto the subspace of the corresponding PDM and obtain coefficients \( f_{ij} = \mathbf{V}^F_i \times (\mathbf{U}^F_j - \mathbf{U}^F_i) \).

   Output: Feature vectors obtained by concatenating coefficients from the projections onto all expressions \( \mathbf{a}_i = [f_{i,0} \ldots f_{i,n}] \)

Algorithm 3. Testing

Input: An input mesh broken down into \( m \) components \( \mathbf{A} \)
1. Re-align and normalize each component \( \mathbf{A}_i \) with the means of the geometric PDMS \( \mathbf{U}^F_i \) such that \( \| \mathbf{A}_i \| \) is equal to 1.
2. Compute features \( F \) on the aligned data
3. Project the features \( F \) onto the subspace of the corresponding PDM and obtain coefficients \( f_{ij} = \mathbf{V}^F_i \times (\mathbf{U}^F_j - \mathbf{U}^F_i) \).

   Output: Feature vectors obtained by concatenating coefficients from the projections onto all expressions \( \mathbf{a}_i = [f_{i,0} \ldots f_{i,n}] \)

4. 4D facial expression recognition

In this section, we present a fully automatic pipeline for classifying six prototypical expressions from 4D videos. Note that in this work we assume that the input contains a complete expression sequence. In other words, we expect that each expression is performed gradually; from neutral to onset, to apex, and then to offset and neutral. This is a reasonable assumption since in reality people display expressions more or less in a similar manner. Since the 4D videos run at 30 fps, we make another assumption that the deformation between two consecutive frames is reasonably small.

The proposed pipeline will still leverage the benefits from the deformable model approach, in a way that the temporal coherence of the sequence is taken into consideration and is well preserved. Fitting AFM in the dynamic sequence frame by frame as the static cases does not make use of the temporal information embedded in the time series and it is time consuming. Instead, we propose to compute the rigid transformations between each pair of consecutive frames. This facilitates the subsequent fitting step. A set of vertex correspondences between two meshes is first computed. The outliers of correspondences are then filtered out. The rest are finally used for robust estimation of the transformation via Procrustes analysis. After the registration for all frames in a sequence, LBP-TOP is applied on the flow matrix and the histogram of LBP-TOP is extracted for recognition.

4.1. Mesh matching

We present two methods to find the vertex correspondences between two meshes, one based on spin image similarities [60] and the other based on the Euclidean distances between MeshHOG descriptors [61].

4.1.1. Matching using spin images

Following the intuition that matches are usually found between “interesting” vertices with distinctive local shapes, we start by computing the shape index values of all the vertices on the mesh. The shape index, \( \rho \), is a continuous mapping of the principal curvature values at a vertex \( \mathbf{v} \) into the interval \([0,1]\) and captures the intuitive notion of local geometry. It can be computed as [62]:

\[
\rho = \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2},
\]

where \( \kappa_1 \) and \( \kappa_2 \) are the maximum and minimum values of the principal curvatures at \( \mathbf{v} \), respectively, and are the eigenvalues of the local Hessian matrix \( \mathbf{H} \). The vertices with \( \rho \) close to 1 have local shapes that resemble a cap and the vertices with \( \rho \) close to 0 have local shapes that resemble a cup. We then set two thresholds to select candidates for corner detection. Harris and Stephens [63] suggest that the corner response \( k \) can be computed as

\[
k = \frac{\text{trace}^2(\mathbf{H})}{\det(\mathbf{H})}.
\]

Subsequently, spin images [60] are computed at the vertices that exhibit a strong corner response. A spin image is a representation of the geometric neighborhood around a specific point of a 3D object. It encodes the coordinates of points on the surface w.r.t. a local basis, namely oriented point and is thus invariant under rigid transformations. The similarity between two spin images \( \mathbf{s}_i \) and \( \mathbf{s}_j \) is defined as [64]:

\[
\text{Similarity} = \left[ \frac{\text{tanh}^{-1}(f_c(\mathbf{s}_i, \mathbf{s}_j))}{\lambda} \right]^2 - \frac{1}{n-3},
\]

where \( n \) is the number of pixels that overlap and is also used in the computation of \( f_c \) (which computes the correlation coefficient between \( \mathbf{s}_i \) and \( \mathbf{s}_j \)), and \( \lambda \) is the expected overlap.

After filtering by using the shape index values and corner responses, the candidate pool becomes reasonably small. Therefore, generating and matching spin images exhaustively is avoided, thereby reducing the computation burden while simultaneously offering the guarantee that the closest match will be found. Finally, we consider only the matches that have similarity above a predefined threshold.

4.1.2. Matching using MeshHOG

Zaharescu et al. [61] studied surface feature detection and description. They revisited local feature detectors and descriptors developed for 2D images and extended the concept to accommodate 3D surfaces with associated scalar functions. Operators such as the discrete convolution and discrete gradient are formulated by taking into consideration both the differential properties of the scalar functions and the intrinsic geometry of the discrete surfaces. Based on these formulations, the authors proposed a new interest point detector and a new local descriptor. The detector, named MeshDOG, is a generalization of the difference of Gaussians (DOG) operator [65,66] and seeks the extrema of the Laplacian of a scale-space representation of any scalar function defined on the mesh. The local descriptor, named MeshHOG, is a generalization of the histogram of oriented gradients (HOG) descriptor commonly used for describing 2D images [67].

Feature detection starts with finding the extrema of the DOG of the scalar function across scales, followed by non-maximum suppression. These extrema are then thresholded by considering only the top 5% of the maximum number of vertices. Additionally, a cornerness measure (similar to the one described in Section 4.1.1) is computed at each vertex to further eliminate non-stable responses.

The descriptor is constructed over a support region, defined as the neighborhood of a vertex \( \mathbf{v} \). First, a local coordinate system is formed using the normal \( \mathbf{n}_v \) and two other unit vectors defined in the tangent plane at \( \mathbf{v} \). The gradient vectors within the support region are projected onto the three orthonormal planes of the local coordinate system. Then a two-level histogram is computed on each of the three planes. Each plane is divided into four polar slices and each vertex in the neighborhood of \( \mathbf{v} \) will fall within one of the slices when projected onto that particular plane. For each such spatial slice, an orientation histogram of the projected gradient vectors of the projected vertices is computed. The final descriptor is obtained by concatenating all the histograms from the three planes, followed by L2 normalization.

For each descriptor \( \mathbf{t}_i \) on the source mesh, the best matching descriptor \( \mathbf{t}_j^\ast \) on the target mesh is found as the one with the shortest Euclidean distance \( d_{ij} = \| \mathbf{t}_i - \mathbf{t}_j^\ast \| \) from \( \mathbf{t}_i \). Cross validation is also performed to ensure that \( \mathbf{t}_i \) is also the best match of \( \mathbf{t}_j^\ast \). To avoid confusion cases, a pair of correspondences is only accepted if its second best match is significantly worse.
4.2. RANSAC

The correspondences found by matching either spin images or MeshHOG may contain outliers (i.e., false matches). These noisy point correspondences will lead to erroneous estimation of the rigid transformation between the frames. To alleviate this problem, we use RANdom SAMple Consensus (RANSAC) [68] that minimizes the influence of the outliers.

RANSAC has been extensively used in computer vision for estimating parameters of mathematical models from a set of observed data which may contain outliers. The two basic assumptions of RANSAC are: (i) the majority of the input data are inliers and their distribution can be explained by some set of model parameters; and (ii) there exists a procedure to estimate the model parameters given only a small set of inliers. The problem we are trying to solve satisfies both assumptions. First, most of the correspondences are legitimate matches such that the points on the two meshes differ by a rigid transformation. The matches around which a small amount of non-rigid deformation takes place can be treated as outliers. Second, given a set of correspondences, we can use Procrustes analysis [69] to compute this rigid transformation in a similar fashion as the AFM pre-alignment described in Section 3.2.

4.3. Consecutive fitting

To generate a fitted sequence from a 4D dataset, we first fit the initial frame $F_0$ using the standard deformable model approach [5]. Let us denote the fitted model as $F_0$ and the transformation matrix $M_0$. We then compute the transformation as $M$ from $F_1$ to $F_0$. Therefore, $F_1$ can be registered with the AFM using $M_1 = M \cdot M_0$. Instead of deforming the original AFM to fit $F_1$, we start the deformation from $F_0$ so that its local curvature will be similar to the local curvature of $F_1$, which in turn ensures the semantically correct vertex movement. The rest of the sequence is handled in the same manner. Recall that the sequence will start from neutral expression and the difference between consecutive frames is reasonably small. Therefore, a fitted model from the previous frame incorporates the prior information and facilitates the subsequent deformation, improving both speed and accuracy of the fitting procedure.

4.4. Feature description

We adopt Local Binary Patterns from Three Orthogonal Planes (LBP-TO) for expression analysis in the fitted sequences. It was introduced by Zhao and Pietikäinen [10] for dynamic texture recognition. It is proven to be very effective in the application of FER in 2D videos. Although the descriptor was designed for 2D videos, because of the advantages of the AFM formulation, the fitted models of a sequence can be represented in uv-space as geometry images, which can be considered the frames in 2D videos. With some adjustment, we can then leverage the aforementioned descriptors for 4D FER.

Local Binary Patterns (LBP) have been used extensively in 2D FER [70], due to their effectiveness in texture analysis and computational simplicity. The original LBP operator was introduced by Ojala et al. [71]. It labels the pixels of an image by thresholding a neighborhood of each pixel with the value in the center and translates the results into binary numbers. These binary numbers codify local patterns of different types and are accumulated in a histogram over a predefined region. This histogram essentially becomes the descriptor of the region and the image can be described by a concatenation of such histograms.

Zhao and Pietikäinen [10] extend the definition of LBP to account for the temporal evolutions of the local image patterns, namely LBP-TOP. In their approach, the 2D video is viewed as a three dimensional volume with the dimensions $x$, $y$, and $t$. Therefore, instead of only computing LBP descriptors on the $xy$-plane, the authors also take into consideration the $xt$- and $yt$-planes when applying the operator. In other words, when thresholding using the value of the center pixel, its neighbors are sampled from all three planes. The resulting binary number encodes not only the spatial pattern but also the temporal evolution pattern.

Directly applying LBP-TOP to the geometry images (Fig. 6) will not work since there is no notion of texture in geometry images. This is because the values in $x$ and $y$ channels are changing uniformly according to their position in the grid. To introduce variations in these channels, we propose to compute LBP descriptors on the difference between a frame and the first frame of the sequence or between a frame and its previous frame, using the idea of flow matrix [25], which we term the flow image of that particular frame (Fig. 6). Significant features can be observed when the vertices exhibit significant amount of deformation w.r.t. their initial positions. Due to the property of the $uv$-parameterization, the same location on the $\mathbb{R}^2$ grid always refers to the same point in $\mathbb{R}^3$. Thus, we create LBP histograms of different facial components on the flow image. The components are defined by the AFM (Fig. 1), instead of simple rectangular regions [10]. Moreover, considering that most of the values on the $\mathbb{R}^2$ grid are interpolated from pixels where the control vertices of the AFM are mapped, we compute only the binary encodings at their corresponding pixels in the $uv$ space.

5. Results

5.1. 3D expression classification

To demonstrate the effectiveness of the proposed framework, we devised a simple extension to the static 3D FER pipeline. The geometric features of a 3D mesh (shape and normals) have been successfully used for face recognition [5]. We want to also explore their applicability in 3D FER, considering the fact that most of the previous studies have already investigated features that are invariant under rigid transformations (i.e., curvatures [12] and distances [15,26,22]). Therefore, we consider two features in this work. The first feature is basically a concatenation of geometric coordinates of all vertices and the other is a concatenation of their normals.

Due to its effectiveness and popularity, we choose the support vector machines (SVM) with a Radial Basis Function kernel as the classifier in our adaptation. This classifier is widely used in data analysis and pattern recognition. The SVM constructs hyperplanes to maximize the separation between different categories in a highly nonlinear space. The SVM implementation we use is LIBSVM [72]. Also, note that we did not fine tune the soft margin parameter $C$ and the kernel parameter $\gamma$ to boost the performance on any specific training set.

The results of this adaptation are reported on the BU-3DFE database [7]. Following the experimental convention of most of the previous work in this field [12,15,22,25,28,33,34], we first randomly selected 60 subjects, in which half were male and half were female. Then, the two high-intensity models for each of the expressions of the selected subjects are randomly divided into two sets. The training set contains the datasets from 54 subjects and the testing set contains the datasets.
Table 2
Confusion matrix when the vertex coordinates are used to form the feature descriptors. The average classification rate is 84.1%. AN denotes angry expression. DI denotes disgust expression. FE denotes fear expression. HA denotes happy expression. SA denotes sad expression. SU denotes surprise expression. (The same abbreviations are used in the rest of the paper).

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>DI</th>
<th>FE</th>
<th>HA</th>
<th>SA</th>
<th>SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>83.33</td>
<td>1.52</td>
<td>0.76</td>
<td>0.00</td>
<td>14.39</td>
<td>0.00</td>
</tr>
<tr>
<td>DI</td>
<td>7.58</td>
<td>81.06</td>
<td>9.85</td>
<td>0.00</td>
<td>0.00</td>
<td>1.52</td>
</tr>
<tr>
<td>FE</td>
<td>6.82</td>
<td>4.55</td>
<td>75.76</td>
<td>6.82</td>
<td>3.03</td>
<td>3.03</td>
</tr>
<tr>
<td>HA</td>
<td>0.00</td>
<td>0.76</td>
<td>7.58</td>
<td>90.91</td>
<td>0.00</td>
<td>0.76</td>
</tr>
<tr>
<td>SA</td>
<td>12.12</td>
<td>1.52</td>
<td>1.52</td>
<td>0.00</td>
<td>84.85</td>
<td>0.00</td>
</tr>
<tr>
<td>SU</td>
<td>0.00</td>
<td>1.52</td>
<td>9.09</td>
<td>0.76</td>
<td>0.00</td>
<td>88.64</td>
</tr>
</tbody>
</table>

Table 3
Confusion matrix when the vertex normals are used to form the feature descriptors. The average classification rate is 91.0%.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>DI</th>
<th>FE</th>
<th>HA</th>
<th>SA</th>
<th>SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>92.42</td>
<td>0.76</td>
<td>0.00</td>
<td>0.00</td>
<td>6.82</td>
<td>0.00</td>
</tr>
<tr>
<td>DI</td>
<td>3.03</td>
<td>91.67</td>
<td>3.03</td>
<td>0.00</td>
<td>0.00</td>
<td>2.27</td>
</tr>
<tr>
<td>FE</td>
<td>5.30</td>
<td>4.55</td>
<td>81.06</td>
<td>2.27</td>
<td>4.55</td>
<td>2.27</td>
</tr>
<tr>
<td>HA</td>
<td>0.00</td>
<td>0.00</td>
<td>1.52</td>
<td>98.48</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SA</td>
<td>8.33</td>
<td>1.52</td>
<td>1.52</td>
<td>0.00</td>
<td>88.64</td>
<td>0.00</td>
</tr>
<tr>
<td>SU</td>
<td>0.00</td>
<td>0.00</td>
<td>5.3</td>
<td>0.76</td>
<td>0.00</td>
<td>93.94</td>
</tr>
</tbody>
</table>

Table 4
Confusion matrix when using MeshHOG for mesh matching and LBP-TOP as the feature descriptor. The average classification rate is 75.82%.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>DI</th>
<th>FE</th>
<th>HA</th>
<th>SA</th>
<th>SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>68.31</td>
<td>11.17</td>
<td>3.11</td>
<td>1.11</td>
<td>16.31</td>
<td>0.00</td>
</tr>
<tr>
<td>DI</td>
<td>7.47</td>
<td>79.69</td>
<td>3.00</td>
<td>3.92</td>
<td>2.67</td>
<td>3.25</td>
</tr>
<tr>
<td>FE</td>
<td>11.28</td>
<td>3.47</td>
<td>67.89</td>
<td>6.81</td>
<td>7.08</td>
<td>3.47</td>
</tr>
<tr>
<td>HA</td>
<td>3.97</td>
<td>3.97</td>
<td>9.51</td>
<td>81.31</td>
<td>0.00</td>
<td>1.25</td>
</tr>
<tr>
<td>SA</td>
<td>23.00</td>
<td>0.00</td>
<td>4.36</td>
<td>0.00</td>
<td>71.64</td>
<td>1.00</td>
</tr>
<tr>
<td>SU</td>
<td>0.00</td>
<td>0.00</td>
<td>3.95</td>
<td>4.55</td>
<td>2.86</td>
<td>2.54</td>
</tr>
</tbody>
</table>

5.2. 4D registration

To compare different registration implementations, we introduce a simple measure of maximum per pixel deviation, which is the largest magnitude of a flow image. Since we expect smooth deformation between the frames, the deviation should be reasonably small. Matching by spin images and MeshHOG both offer similar performances in registering the whole sequence to the AFM while maintaining the temporal coherence, as can be observed from the proximity of the two curves in Fig. 7. This point is also substantiated by the insignificant differences in the FER results (Tables 4 and 5). If instead, each frame is registered separately using the ICP-based method, this coherence may break due to the fact that the transformation matrix of each mesh is computed independently. The registration error is exaggerated when we look at the sequence as a whole and is also reflected in the fitted models (Fig. 7).

The supplementary material [73] includes the video of an input sequence, input.mp4. The advantage of the mesh matching-based registration over the ICP-based method is evident when comparing fitted_mesh_matching.mp4 and fitted_ICP.mp4. Three fitted sequences are provided, which show that matching using either spin images or MeshHOG produces visually identical results (fitted_spin_images.mp4 and fitted_MeshHOG.mp4), while registration using ICP breaks the temporal coherence of the dynamic sequence (fitted_ICP.mp4).

Given that the non-rigid deformation between consecutive frames is very small, the spin images of the corresponding points would almost be the same. Therefore, the spin similarity is a very solid measure for correspondence. Matching by MeshHOG, on the other hand, produces a few more outliers than spin images (Figs. 8 and 9). Nonetheless, these outliers are subsequently neutralized by RANSAC during transformation estimation and their influence on the expression classification is negligible.
Table 5
Confusion matrix when using spin images for mesh matching and LBP-TOP as the feature descriptor extracted from flow matrix computed using the neutral frame (left) and the consecutive frame (right). The average classification rates are 74.63% (left) and 74.34% (right).

<table>
<thead>
<tr>
<th>%</th>
<th>AN</th>
<th>DI</th>
<th>FE</th>
<th>HA</th>
<th>SA</th>
<th>SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>73.53/65.00</td>
<td>7.83/14.00</td>
<td>3.36/5.00</td>
<td>1.00/0.00</td>
<td>14.28/16.00</td>
<td>0.00/0.00</td>
</tr>
<tr>
<td>DI</td>
<td>20.37/8.44</td>
<td>62.29/69.96</td>
<td>11.35/13.02</td>
<td>0.00/0.00</td>
<td>3.75/2.25</td>
<td>2.25/6.33</td>
</tr>
<tr>
<td>FE</td>
<td>11.94/0.00</td>
<td>3.54/10.00</td>
<td>61.13/65.00</td>
<td>7.19/10.00</td>
<td>11.55/11.00</td>
<td>4.65/4.00</td>
</tr>
<tr>
<td>HA</td>
<td>2.54/1.00</td>
<td>0.00/1.11</td>
<td>6.01/10.11</td>
<td>89.78/87.77</td>
<td>0.00/0.00</td>
<td>1.67/1.00</td>
</tr>
<tr>
<td>SA</td>
<td>23.08/13.00</td>
<td>0.00/0.00</td>
<td>1.11/15.00</td>
<td>1.11/0.00</td>
<td>75.70/71.00</td>
<td>0.00/1.00</td>
</tr>
<tr>
<td>SU</td>
<td>0.00/1.00</td>
<td>2.11/2.11</td>
<td>8.22/6.33</td>
<td>2.22/0.00</td>
<td>1.11/2.22</td>
<td>86.33/87.34</td>
</tr>
</tbody>
</table>

5.3. 4D expression classification

The proposed dynamic 3D FER pipeline was tested on the BU-4DFE database [9]. This database contains 606 dynamic 3D sequences from 101 subjects, with each subject performing the six prototypical expressions (i.e., anger, disgust, fear, happiness, sadness and surprise). Although in the database description [9], the authors state that each sequence contains expression performed gradually from neutral appearance, low intensity, high intensity, and back to low intensity and neutral, it is not the case for some of the sequences (Fig. 10). Following our assumption that the input sequence starts from a neutral state, we manually removed the videos in which the subjects do not start with a neutral expression. We have also identified and removed the videos containing corrupted meshes (Fig. 11) and the videos that have obvious discontinuity (i.e., not a single continuous sequence). The manual selection leaves us with 507 sequences from 100 subjects.

Noticing that most of the meshes have spike-shaped reconstruction artifacts around their borders, as a preprocessing step, we removed the two levels of boundary triangles. We then performed 10-fold cross validation on the remaining sequences. To avoid gender bias, the subjects were selected independently from the females and males. Furthermore, we guarantee that no subjects appear in both the training and the testing sets. We still use SVM with a Radial Basis Function kernel for classification [20,21,22], since the temporal information is already embedded in the LBP-TOP descriptors. Grid search was performed to find the optimal values of C and γ during training. The best results were obtained by using either spin images or MeshHOG for mesh matching and LBP-TOP as the feature descriptor computed on the flow matrix referred to the neutral, as summarized in Tables 4 and 5. The differences between the two matching methods were insignificant and hence, we will refer to the results produced by matching with spin images in the rest of the paper, unless stated otherwise. Meanwhile, we also computed the LBP-TOP between two consecutive frames so that the requirement of the neutral frame can be released. The results are presented on the right column of each cell in Tables 5, 6, 7.

To the best of our knowledge, there are three methods in the literature that directly tackle dynamic 3D FER on the same database with explicit experiment protocols [18,38,39]. Each of these methods was evaluated on a different subset of the BU-4DFE database. Sun and Yin [18] selected 60 subjects and considered all six expressions in their experiment. Sandbach et al. [38] manually selected a subset of the subjects who are considered to be posing the instructed expression accurately, and they tested only on angry, happy, and surprise expressions. Le et al. [39] also selected 60 subjects and tested only on happy, sad, and surprise expressions.

We want to note that even though Sun and Yin [18] achieved better classification performance (90.44% compared to our 74.63%) when considering all six expressions, they used significantly fewer sequences (at most 360 comparing to our 507). The effect of this particular reduced evaluation set remains unknown. Hence, the results are not directly comparable and are stated only for reference. Nevertheless, when tested on the more restricted subsets, our 4D FER framework outperforms both the approaches proposed by Sandbach et al. [38] and Le et al. [39]. When tested on angry, happy, and surprise expressions, we achieved 96.71% average classification rate (Table 6) comparing to 81.93% from Sandbach et al. Using a larger subject pool and considering only happy, sad, and surprise expressions, we obtained 95.75% average classification rate (Table 7) compared to 92.22% from Le et al. These comparisons have demonstrated the effectiveness of the proposed pipeline.

6. Conclusions

The availability of dedicated 3D FER databases and the advancement in sensor technology have enabled us to investigate different representations of the human facial expressions. In this paper, we have presented two facial expression recognition frameworks that work with 3D and 4D data. Both frameworks were built on top of the annotated deformable model approach [5] and have extended its applicability to FER. We have improved the AFM on static 3D face scans with expressions by introducing Procrustes Analysis and TPS to the fitting process. We have also designed a module for registering 4D data with temporal coherence...
and compared different implementations. To the best of our knowledge, this is the first approach that has the ability to register 4D data and bring them into dense correspondence without the need of texture information or explicit parameterization. We have also proposed a component based PDM for 3D FER and adapted the spatio-temporal descriptor LBP-TOP to the geometry flow images for 4D FER. Promising results were obtained on the two publicly available databases, BU-3DFE and BU-4DFE.

Nevertheless, there are still many interesting questions that need to be addressed. Given the formulation of the 3D and 4D FER systems, we can plug in different modules for descriptors and classifiers and study their influence. For example, combining 3D features (e.g., curvature or normal) with 2D features (e.g., wavelets) is a possible direction. It is interesting to test which kinds of feature fit our PDM based framework. Even with LBP-TOP descriptors, we can introduce boosting for feature selection [74]. Instead of encoding temporal information in the features, classifiers aware of the temporal evolutions such as the ones based on hidden Markov models can also be considered.

On the higher level, as noted by Fang et al. [8], more effort is expected to be directed to AU recognition. In order for facial expression recognition to be more practically useful, the focus of study should shift from the posed expressions to those that are elicited spontaneously, as many studies have suggested that there are significant differences between the two [4].

Supplementary materials related to this article can be found online at doi:10.1016/j.imavis.2012.02.004

Acknowledgments

This research was funded in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the Army Research Laboratory (ARL) and by the University of Houston (UH) Eckhard Pfeiffer Endowment Fund. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI, the U.S. Government, or UH.

Table 6
Confusion matrix when considering only angry, happy, and surprise expressions. LBP-TOP extracted from flow matrix computed using the neutral frame (left) and the consecutive frame (right). The average classification rates are 96.71% (left) and 96.88%.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>HA</th>
<th>SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>97.32/95.00</td>
<td>2.68/1.00</td>
<td>0.00/4.00</td>
</tr>
<tr>
<td>HA</td>
<td>2.00/2.25</td>
<td>96.33/97.75</td>
<td>1.67/0.00</td>
</tr>
<tr>
<td>SU</td>
<td>2.54/2.11</td>
<td>1.00/0.00</td>
<td>96.46/97.89</td>
</tr>
</tbody>
</table>

Table 7
Confusion matrix when considering only happy, sad, and surprise expressions. LBP-TOP extracted from flow matrix computed using the neutral frame (left) and the consecutive frame (right). The average classification rates are 95.75% and 91.41%.

<table>
<thead>
<tr>
<th></th>
<th>HA</th>
<th>SA</th>
<th>SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>97.32/93.89</td>
<td>1.43/5.11</td>
<td>1.25/1.00</td>
</tr>
<tr>
<td>SA</td>
<td>1.11/5.00</td>
<td>98.89/85.56</td>
<td>0.00/9.44</td>
</tr>
<tr>
<td>SU</td>
<td>4.61/1.00</td>
<td>4.36/4.22</td>
<td>91.03/94.78</td>
</tr>
</tbody>
</table>

References
