Image and Vision Computing 30 (2012) 762–773

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Recognition of 3D facial expression dynamics ☆

Georgia Sandbach [a,*], Stefanos Zafeiriou [a], Maja Pantic [a,b], Daniel Rueckert [a]

[a] Imperial College, Department of Computing, London, UK
[b] University of Twente, Department of Computer Science, Enschede, The Netherlands

ARTICLE INFO

Article history:
Received 15 July 2011
Received in revised form 2 December 2011
Accepted 22 January 2012

Keywords:
Facial expression recognition
3D facial geometries
Motion-based features
Quad-tree decomposition
2D/3D comparison

ABSTRACT

In this paper we propose a method that exploits 3D motion-based features between frames of 3D facial geometry sequences for dynamic facial expression recognition. An expressive sequence is modelled to contain an onset followed by an apex and an offset. Feature selection methods are applied in order to extract features for each of the onset and offset segments of the expression. These features are then used to train GentleBoost classifiers and build a Hidden Markov Model in order to model the full temporal dynamics of the expression. The proposed fully automatic system was employed on the BU-4DFE database for distinguishing between the six universal expressions: Happy, Sad, Angry, Disgust, Surprise and Fear. Comparisons with a similar 2D system based on the motion extracted from facial intensity images was also performed. The attained results suggest that the use of the 3D information does indeed improve the recognition accuracy when compared to the 2D data in a fully automatic manner.

1. Introduction

It is widely expected that in the future computing will move into the background, becoming a part of our everyday life, with the user moving into the foreground. As a part of this transition, the interactions between users and computers will need to become more natural, moving away from the traditional interface devices, and replicating human-to-human communication to a larger extent. Facial expressions constitute an important factor of communication, revealing cues about a person's mood, meaning and emotions. Therefore the requirement for accurate and reliable facial expression recognition systems is a crucial one.

Recognition of facial expressions is a challenging problem, as the face is capable of complex motions, and the range of possible expressions is extremely wide. Even recognition of the six universal expressions – happiness, sadness, anger, disgust, fear and surprise – is a difficult problem, due to the wide variations seen between subjects when expressing these emotions, and the differences between acted and naturalistic examples.

Expression dynamics are of great importance for the interpretation of human facial behaviour [1]. They convey cues for behaviour interpretation [2], and are useful for distinguishing between spontaneous and posed emotional expressions [3]. In addition, they are essential for the recognition of complex states such as pain and mood [4], as well as of more subtle emotions such as social inhibition, embarrassment, amusement and shame [5,6]. It is therefore obvious that a system capable of accurate and robust expression recognition will need to harness the information available in expression dynamics.

Methods and systems have been proposed for automatic facial expression and facial action unit (AU) recognition from 2D facial images and video. Unfortunately, these systems are highly sensitive to the recording conditions such as illumination conditions, facial pose and others changes in facial appearance like make up, sunglasses etc. More precisely, in most cases when 2D facial intensity images are used it is necessary to maintain a consistent facial pose (preferably a frontal one) in order to achieve good recognition performance. Even small changes in facial pose can reduce the effectiveness of the systems. For these reasons, it is now widely accepted that in order to address the challenge of accuracy, different capture modalities (such as 3D or infrared) must be employed. Furthermore, advances in structured light scanning, stereo photogrammetry and photometric stereo have made the high-end acquisition of 3D facial structure and motion a feasible task [7].

The use of 3D facial geometry data and extracted 3D features for expression recognition has so far not been heavily studied. Images and videos of this kind will allow a greater amount of information to be captured (2D and 3D), including out-of-plane movement which 2D cannot capture, and remove the problems of illumination and pose inherent to 2D data. There are previous research efforts that use 2D images to construct 3D models in order to extract 3D features that can be used for classification of the facial expression, such as in [8–10]. However these methods are also susceptible to the problems of illumination and pose inherent to all 2D methods. For this reason, more recently several methods have been proposed which

☆ This paper has been recommended for acceptance by Lijun Yin, Ph.D.

* Corresponding author.
E-mail addresses: gls09@imperial.ac.uk (G. Sandbach), s.zafeiriou@imperial.ac.uk (S. Zafeiriou), m.pantic@imperial.ac.uk (M. Pantic), d.rueckert@imperial.ac.uk (D. Rueckert).

0262-8856/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2012.01.006

use 3D facial geometry data for facial expression recognition, either for static analysis [11–16], to encode the temporal information [16,17], or to model the temporal dynamics of facial expressions in 3D image sequences [18,19].

Expression dynamics and 3D facial geometry data combined offer a wealth of information that can be harnessed for the analysis of facial expressions. The development of such systems will open up new avenues in facial expression recognition as 3D facial geometries ensure that all motion in the face will be captured, unlike 2D data, and analysis of full expression dynamics allows cues to be detected that are unavailable in static data. This paper proposes a method that aims to exploit the advantages in data of this kind through the extraction of 3D motion-based features and temporal modelling of the full expression dynamics for recognition purposes.

We propose a fully automatic method for facial expression recognition which consists of several stages. Firstly the 3D motion of the face appearing between frames in each image sequence is captured using Free-Form Deformations (FFDs) [20]. We extract features by applying a quad-tree decomposition of the motion fields. Features are then collected using a GentleBoost (GB) feature selection method for the onset and offset temporal segments of the expression and frame classification. Temporal modelling of the full expression is performed via neutral–onset–apex–offset hidden Markov models (HMMs). These models are then used for dynamic expression recognition. We have also conducted a comparison between the use of motion extracted from 2D facial intensity and 3D facial geometry information using a similar methodology in order to prove the superiority of the latter approach.

In summary, the novel contributions of this paper are as follows:

- Employing 3D FFDs, sets of 2D vector projections and quad-trees in order to perform 3D motion-based feature extraction.
- An extension of the method proposed in [21] to perform expression recognition using both 2D intensity images and 3D facial geometry information.
- Modelling of the temporal segments of the full expression rather than those of action units.

A comparison of the equivalent 2D and 3D methods is then performed on the same database, the BU-4DFE, in order to assess the benefits of the 3D data. To the best of our knowledge, this is the first fully automatic approach for dynamic 3D facial expression recognition.

2. Related work

The use of facial expression dynamics in expression recognition systems has recently increased dramatically. Analysis of this kind makes use of the implicit temporal encoding in expressions which have been shown to hold the key to distinguishing between different meanings and emotions. The majority of work in this field so far has made use of 2D image sequences, though a few works have started to capitalise on 3D facial geometry data. In addition research has been conducted into analysis of facial expressions from 3D static data, which is also related to the work presented here.

In this section we discuss previous 2D dynamic facial expression analysis, and then go on to look at the 3D static and dynamic work that has been completed in this area, focusing mainly on the feature extraction stage as this provides the main differences in the analysis of expressions in 3D versus 2D images and image sequences.

2.1. 2D facial expression dynamics analysis

Facial expression recognition systems generally consist of several different stages: feature extraction, feature classification and temporal modelling. This section examines the main techniques that have been used for each of these stages in previous 2D dynamic work.

2.1.1. Feature extraction

The feature extraction approaches can mostly be divided into three categories of approaches: geometric features, appearance based features, and motion based features. There have been several examples of the use of geometric features which concentrate on the shapes of particular facial components or the position of facial fiducial points. These include the works in [22–25], each of which track facial feature points, and use the movements of these points to find intermediate parameters.

Appearance-based methods have been used throughout facial expression recognition work on 2D image sequences. These include the use of Gabor wavelets in [26] to produce feature representations of each frame that can then be used for classification. The work in [27] is another example of work that used a set of Gabor wavelet coefficients, but this time facial feature points were identified before applying the filters at only these locations. An alternative filter type, exploited in [28], are morphological operators which use the processes of dilations and erosion to highlight various facial characteristics useful for expression analysis. Another type of feature descriptor that have been employed for facial expression recognition are local binary patterns (LBPs), [29]. A final alternative method used in [30] involved embedding feature vectors representing images into a manifold in order for temporal analysis to be done. The dynamics of the expression were then traced through the low-dimensional representation of the unfolded manifold.

Motion-based features were used in [21,31]. In these works, free-form deformations (FFDs) [32] were used to capture the motion between frames which is used to extract features from different regions in each image. Optical flow features, computed through a least squares approach, were also employed in [33,34]. An alternative method was employed in [35], where Gabor energy filters were used to encode the motion in image sequences.

Some works used features that implicitly encode the temporal information. These works have not generally aimed to model the temporal segments of the expression. One such method was the LBP based descriptor, LBP-TOP [36], which encodes the full image sequence as a three-dimensional image (2D + time). Alternatively, multi-linear representations of the image sequence were used in [37] for classification.

2.1.2. Classification

Several classification techniques have been used in previous dynamic 2D work. Simple rules were used as classifiers in [23,24] to discriminate between AUs from mid-level parameters determined from the features. Multi-class Support Vector Machines (SVMs) have been used in many works: in [25,38] to analyse facial action unit temporal segments, in [35] for analysis of full expression dynamics and in [22] for recognition of both. Additionally the AdaBoost algorithm has been used prior to classification to choose features that give the most information which were then passed to SVMs [38,35]. Alternatively, AdaBoost was used for classification itself in [26], and a variation on AdaBoost classification, GentleBoost [39] which uses a different update rule to AdaBoost which makes the classifier more stable and increases the speed of convergence, was used for classification in [21]. An additional method that has been used in several works such as [8,10] is Tree-Augmented Naive Bayes (TAN) classification which model the dependencies between the different features.

2.1.3. Temporal modelling

The use of temporal modelling of the dependencies between frames in either a full expression or action unit is an important step in the analysis of facial expression dynamics. It allows the information contained in the relationships between movement at different points in the expression to be harnessed for recognition. One method used for this purpose is dynamic Bayesian networks (DBNs) [27,26]. In this work static Bayesian networks were created for each time step

in the image sequence, and dependency links are added between nodes, AUs in this case, both within the time step, and between them.

An alternative model regularly used for dynamic facial expressions analysis is the HMM, a tool best known for its use in speech processing. These model the observable output as dependent on the states of hidden variables, which in this case can represent the different temporal segments of the expression or AU. These were used in [8,34] to model the full expression dynamics, and in [21] to model AU dynamics. Alternatively, a hybrid SVM-HMM classifier has also be used to model the dynamics of AUs [25].

2.2. 3D static facial expression analysis

Several previous works have used 3D static images for facial expression recognition. In these works the feature extraction stage provides the main differences in methodology over those of the 2D systems. Classification and temporal modelling is carried out using similar techniques to those used in the 2D work.

One feature type that has been used for analysis of 3D data, for example in [40,14], are characteristic distances, extracted from the distribution of facial feature points in the 3D facial geometries. In these works, the distances were then used directly as inputs to classifiers in order to distinguish the six basic expressions. In [14], the discriminative power of a range of distances were determined, in order to select which distances showed the biggest differences between the expressions.

Alternatively, the method proposed in [41] made use of 2D geometric features, 3D curvature features, and moment invariants that combine 3D geometry with 2D texture, and embedded these into an Isomap manifold in order to distinguish between normal expressions and those of people with schizophrenia.

Another method widely used is morphable models, formed from the principal components of a set of 3D faces. Examples of work that use this approach include [11,42,43], in which bilinear models are used to model both the expression and identity, and [44], in which a Morphable Expression Model (MEM) was proposed which allows any expression to be built from a weighted linear combination of components. These models provided a vector representation of the face to be formed, which allowed classification of the expression via clustering.

A final method is to map the 3D information into a 2D representation. This was employed in [45], where preprocessed 3D data was mapped into 2D curvature images with each point in the image representing the curvature of the 3D surface at that point in the 2D plane. These were then used to extract Gabor wavelet features for classification in a similar way to how they are applied to 2D texture images. Similarly, in [13], the 2D images were formed from 3D data, and used to capture the deformation between the resultant meshes and a reference mesh, by using the method developed in [46]. In this work least squares conformal mapping was used for the initial 2D mapping, and then adaptive mesh generation was applied to provide different point densities as required across the mesh. The resulting deformation estimates were used for AU classification.

In addition, some approaches, both for static and dynamic expression analysis, have used 3D models built from 2D images or videos, in order to extract 3D features to be used for expression analysis. Static examples include [47], which fitted a 3D face mesh to each image and used this to produce face texture maps that were independent of the original geometric motion. These were then used to extract high frequency components as features. Another example is [9], in which a model-based tracker was used to extract the pose and shape of the face in each frame before feature extraction was done. A generic face model based on fitting Bezier patches to the 2D images in the sequence was used in [10] and [8]. In the former this was used for static expression analysis only, whereas in the latter facial points in this model were tracked through the image sequences in order to extract features and perform analysis of the dynamics of the expression.

2.3. Encoding of 3D temporal information

One of the first works to exploit 3D motion-based features for facial expression analysis was [16] which performed recognition of the six basic expressions. This work did not aim to explicitly model the temporal dynamics of the expression, rather using motion vectors in order to classify particular expressions. Experiments were conducted on the BU-4DFE database [48], upon which the work presented in this paper is based. A deformable model was used for tracking the changes between frames and from which the motion vectors could be found. These were then classified via an extracted 3D facial expression label map which was produced for each expression.

An alternative method employed in [17,49] used the active shape model (ASM) to represent pairs of 2D and 3D images in order to track the movements of landmarks. These were then used to determine the presence of different deformations in the face that correspond to particular AUs, and classified using a rule-based approach. In this work the features were used in two ways: for a particular frame independently, and alongside the features from previous frames in order to encode the temporal information for use in detecting AUs after they have concluded.

2.4. Dynamic 3D facial expression modelling

3D data was used for analysis of expression dynamics in [18], in which a small 3D database was created which could be used for the analysis. Feature points were tracked in order to capture the deformation of the 3D mesh during the expression. Dimensionality reduction was then used to embed the video sequences into a low dimensional manifold, which then allowed a probabilistic model to be built containing the temporal information from the videos.

One of the first works to conduct experiments using the BU-4DFE database for the analysis of facial expression dynamics was [19]. The deformable model from [16] was adapted to each frame in the image, and then used to track the changes in this model to extract geometric features. Dimensionality reduction was applied via Linear Discriminant Analysis (LDA), followed by the use of 2-dimensional HMMs to model the spatial and temporal relationships between the features.

This work proposes an alternative method for dynamic facial expression analysis. We employ FFDs to model the motion between frames in the image sequence, rather than fitting a deformable model to each mesh. Quad-tree decomposition is used to allow the density of features extracted to reflect the percentage of motion present in each part of the image. The expression is modelled as consisting of four temporal segments, neutral–onset–apex–offset, with GB classifiers trained for the onset and offset segments and then used to build a HMM for the full expression, rather than training directly from the frame features. Each of these stages in our system will be described in detail in the following section.

3. Methodology

An overview of our system can be seen in Fig. 1. In the preprocessing stage, the 3D meshes in each frame are aligned to a reference frame using an iterative closed point (ICP) method [50]. The 3D motion is captured from each set of frames via FFDs [32], and the 3D vector fields are interpolated onto a uniform grid. Vector projections and quad-tree decompositions are calculated in order to determine the regions of the images in which the greatest amount of motion appears. Features are then gathered from each region in each frame, and are used to train classifiers on the onset and offset segments of the expression. The outputs are used to build a HMM of the full expression sequence.

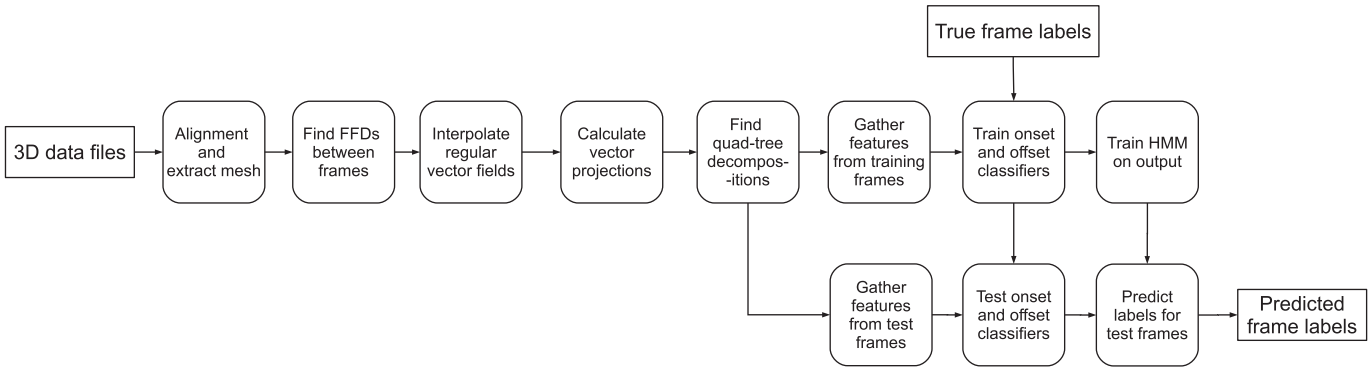

Fig. 1. An overview of the full system including motion caption, feature extraction, classification, and training and testing.

Algorithm 1. Non-rigid registration algorithm

Require: Stopping criterion ε
Require: Step size μ
Initialise the control points Φ'
Calculate the gradient vector of the cost function C with respect to the current control points Φ':

$$\nabla C = \frac{\partial C\left(\Phi'\right)}{\partial \Phi'}$$

while $||\nabla C|| >$ **do**
Recalculate the control points

$$\Phi' = \Phi' + \mu \frac{\nabla C}{||\nabla C||}$$

Recalculate the gradient vector ∇C
end while
Calculate $\Phi_\delta = \Phi - \Phi'$
Derive $\mathbf{T}(\mathbf{p})$ through B-spline interpolation of Φ_δ

3.1. Motion extraction

The motion between the frames in each image sequence was captured using 3D FFDs. FFDs [32] is a method for non-rigid registration based on B-spline interpolation between a lattice of control points. The 2D version of this method was employed for motion capture in [21]. Our aim is given two meshes, with vertices $\mathbf{p} = (x,y,z)$ and $\mathbf{p}' = (x,\acute{y},\acute{z})$ respectively, to find a vector field given by $\mathbf{T}(\mathbf{p})$ such that:

$$\mathbf{p}' = \mathbf{T}(\mathbf{p}) + \mathbf{p}. \quad (1)$$

The basic idea is to deform an object by manipulating an underlying mesh of control points. The lattice, Φ, is regular in the source image and consists of $n_x \times n_y \times n_z$ points $\varphi(i,j,k)$ with regular spacing. This is then deformed by registration of the points in the target image to become Φ' with irregularly spaced control points. The difference between the two lattices is denoted as Φ_δ. $\mathbf{T}(\mathbf{p})$ can be computed using B-spline interpolation on Φ_δ.

For any point in the 3D mesh $\mathbf{p}$, let the closest control point have coordinates (x_0,y_0,z_0) and displacement $\varphi_\delta(i,j,k)$. The transformation of this point can be given as the B-spline interpolation of the 64 closest control points:

$$\mathbf{T}(\mathbf{p}) = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(a_1)B_m(a_2)B_n(a_3)\varphi_\delta(i+l, j+m, k+n) \quad (2)$$

where $a_1 = x - x_0$, $a_2 = y - y_0$, $a_3 = z - z_0$, and B_l is the lth basis function of uniform cubic B-spline, defined as follows:

$$B_0(a) = \frac{1}{6}\left(-a^3 + 3a^2 - 3a + 1\right)$$
$$B_1(a) = \frac{1}{6}\left(3a^3 + 6a^2 + 4\right)$$
$$B_2(a) = \frac{1}{6}\left(-3a^3 + 3a^2 + 3a + 1\right)$$
$$B_3(a) = \frac{1}{6}a^3.$$

$\mathbf{T}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}), w(\mathbf{p}))$ is the vector field used in this work for expression analysis.

In order to calculate Φ_δ, a cost function C is defined. In this paper we chose C to be the sum of squared differences between the points in the target and reference meshes. The non-rigid registration algorithm then proceeds to optimise the control point lattice, Φ', by minimising this cost function. To do this, we employ an iterative gradient descent technique which takes steps with size μ in the direction of the gradient vector. The algorithm finishes when a local optimum is found, which in this case is defined as when the gradient of the cost function reaches a suitably small positive value. The difference between the optimised control point lattice and the original regular lattice is then calculated, and this is used to perform B-spline interpolation in order to find the vector field that captures the motion between the frames. The full algorithm is shown in Algorithm 1.

Fig. 2 shows an example of applying 2D FFDs to extraction of the motion in a pair of images displaying a smile. Here the lattice of control points and the B-spline interpolation between them is shown as a yellow grid. In Fig. 2a the grid is almost regular, whereas in Fig. 2b this grid has been deformed in order to capture the bulging of the cheeks and stretched lips around the mouth.

The resolution of the grid used determines the sensitivity of finely motion tracking between the two images. In this work a grid with control point spacing of 1 mm is used. Fig. 3 shows a neutral and apex mesh for a happiness expression, and the motion tracked by FFDs between these frames. The most highly concentrated areas of motion are around the corners of the mouth and the cheeks, as is expected for this expression.

3.2. Feature extraction

We used motion based features, extracted from the vector fields captured by the FFDs, to train our classifiers. In order to simplify our approach vector projections were computed for each pair of axes (x, y,z and t), in a similar manner to the method used in [21]. However, here it was necessary to compute projections for all three spatial dimensions, resulting in six different projections. Furthermore, in order to focus only on the areas in which the greatest amount of

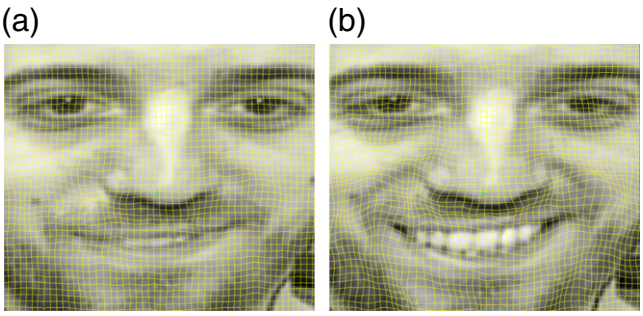

Fig. 2. Example of 2D FFDs applied to aligned face images. Grid shows the control point lattice and the B-spline interpolation of this. (a) Start of onset of Smile. (b) End of onset of Smile.

motion occurs, a quad-tree decomposition was then applied on these projections to divide the vector field into regions according to the amount of motion in every region. Finally, a set of features were extracted from each region.

3.2.1. Vector projections

Vector projections, displayed as an image, show the areas in the image in which there is a high concentration of motion in the sequences across a number of frames (or an axis). Two sets of vector projections were produced from the dataset, one built from frames in which the onset segment of the expression occurred, and other from frames in which the offset segment of the expression occurred. Six 2D vector projections were created from the 3D facial motion. These consisted of three spatial vector projections, one for each pair of spatial axes, and three time-space vector projections.

The spatial vector projections for a window width of θ were calculated as follows:

$$P^{\theta}_{xy}(x,y) = \sum_{i=1}^{M}\sum_{\tau\in\Omega_i}\sum_{t=\tau-\theta}^{\tau+\theta+1}\sum_{z} u^2_{i,x,y,z,t} + v^2_{i,x,y,z,t} + w^2_{i,x,y,z,t} \tag{3}$$

$$P^{\theta}_{xz}(x,z) = \sum_{i=1}^{M}\sum_{\tau\in\Omega_i}\sum_{t=\tau-\theta}^{\tau+\theta+1}\sum_{y} u^2_{i,x,y,z,t} + v^2_{i,x,y,z,t} + w^2_{i,x,y,z,t} \tag{4}$$

$$P^{\theta}_{yz}(y,z) = \sum_{i=1}^{M}\sum_{\tau\in\Omega_i}\sum_{t=\tau-\theta}^{\tau+\theta+1}\sum_{x} u^2_{i,x,y,z,t} + v^2_{i,x,y,z,t} + w^2_{i,x,y,z,t} \tag{5}$$

where Ω_i is the set of frames belonging to the temporal segment in the ith image sequence, M is the total number of image sequences of the current expression in the training set, and

$$u_{i,x,y,z,t} = u^i(x,y,z,t),$$

$$v_{i,x,y,z,t} = v^i(x,y,z,t),$$

$$w_{i,x,y,z,t} = w^i(x,y,z,t)$$

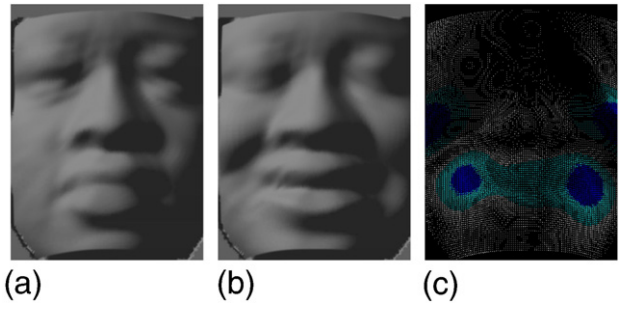

Fig. 3. Mesh representations of neutral and apex frames taken from the Happy image sequence for subject F004, along with the motion tracked between them by FFDs. (a) Mesh of the cropped neutral 3D facial geometry. (b) Mesh of the cropped apex 3D facial geometry. (c) Vector field showing the motion between these frames.

are the vector components, in the x, y and z directions respectively, at coordinates (x,y,z) and time t in the ith image sequence. Note the summation is performed over the window to be used, as well as over the sequence, to ensure all frames that will be used for gathering features influence the quad-tree decomposition.

The time-space vector projections were calculated for t values in the range $0 \le t \le 2\theta - 1$ as follows, using only the vector component in the spatial direction applicable:

$$P^{\theta}_{xt}(x,t) = \sum_{i=1}^{M}\sum_{\tau\in\Omega_i}\sum_{y}\sum_{z} u^2_{i,x,y,z,\tau-\theta+t} \tag{6}$$

$$P^{\theta}_{yt}(y,t) = \sum_{i=1}^{M}\sum_{\tau\in\Omega_i}\sum_{x}\sum_{z} v^2_{i,x,y,z,\tau-\theta+t} \tag{7}$$

$$P^{\theta}_{zt}(z,t) = \sum_{i=1}^{M}\sum_{\tau\in\Omega_i}\sum_{x}\sum_{y} w^2_{i,x,y,z,\tau-\theta+t}. \tag{8}$$

Examples of vector projections can be seen in Fig. 4a–c and Fig. 4g–i, here collected from one fold of onset of the Happy expression with window width of 4. The former shows the spatial vector projections and the latter the space-time vector projections.

3.2.2. Quad-tree decomposition

Before feature extraction could be performed on each of the image sequences, we divided the images into regions from which a set of features was acquired. Instead of dividing the images into evenly sized regions, the technique that we employed was quad-tree decomposition. Quad-tree decomposition has been widely used in computer vision and image processing for image segmentation and feature extraction. In our case we used quad-tree decompositions to divide the image into regions sized according to the amount of motion present in each part of the vector projection. The algorithm, detailed in Algorithm 2, works by measuring the percentage of total motion in the frame that is contained in each region. A region is divided into four equally sized square regions if the percentage it contains is over a certain threshold. A lower limit is set on the region size, below which the regions cannot be divided further. The division continues repeatedly until no further regions can be split. The threshold, γ, used was 6% of the average amount of motion in the blocks. This was determined to give adequate quad-tree decomposition results from preliminary testing. Two sets of quad-tree decompositions were found from the training set — one from the frames consisting of onset motion, and one from frames consisting of offset motion. These sets were then used throughout the training and testing.

Algorithm 2. Quad-tree decomposition

```
Require: Splitting threshold γ
Require: Minimum region size σ
    Define p_tot as the total sum of movement across the full image
    Initialise R with single region which is entire image
    while True do
        for region r in R do
            Set p to be sum of movement in r
            if p > γ p_tot and size of r > σ then
                remove r from R
                divide r into four equally sized square regions
                add these new regions to R
            end if
        end for
        if no region was divided then
            Stop
        end if
    end while
```

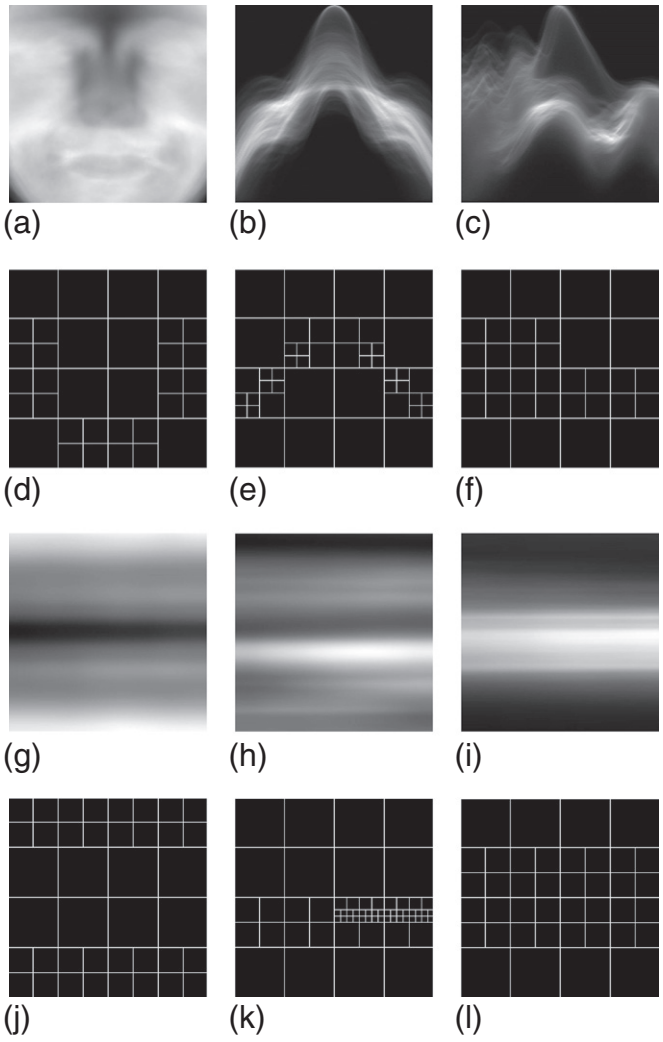

Fig. 4. Spatial and space-time vector projections and the quad-trees they produced for the onset segment of the Happy expression with window width of 4. (a) x–y vector projection. (b) x–z vector projection. (c) y–z vector projection. (d) x–y quad-tree. (e) x–z quad tree. (f) y–z quad tree. (g) x–t vector projection. (h) y–t vector projection. (i) z–t vector projection. (j) x–t quad-tree. (k) y–t quad tree. (l) z–t quad tree.

We used sliding windows throughout the quad-tree decomposition and feature extraction in order to allow information from previous or later frames to be used in the classification of the current frame. This is useful as the duration of a certain motion can help with differentiating between two or more expressions. Various window widths were tested to identify which width gave the best results for each expression. A window width of θ will produce a set of 2θ frames in total.

Examples of the quad-trees produced for each of the vector projections in Fig. 4 can be seen in Fig. 4d–f and Fig. 4j–l. For example, Fig. 4e shows the decomposition created by dividing the vector projection in Fig. 4b according to the amount of motion in the image. The smallest regions correspond to those parts of the image that contain the highest concentration of the motion, whereas the larger regions contain very little motion.

3.2.3. Features

Once the quad-trees had been produced for each vector projection they were used to extract features for every frame in the set of image sequences. For each region in the quad-tree, one set of 3D features was identified and stored. Therefore, areas where little motion was present will be covered by large regions and so produce few features, whereas areas with a large amount of motion produced small regions and so gave many features. The features used included the mean and standard deviation of the distribution of directions of the vectors in that region, the magnitude of the total motion, and the divergence and curl of the vector field in the region. The features from all the regions were concatenated into one feature vector per frame in the image sequences, and these were used for classification.

Again, a sliding window was used to allow frames before or after the current frame to influence the features gathered for that frame. Hence, the features are extracted for a window of width θ around the current frame which is at time τ in the image sequence. The vector field for the frames in this window was averaged across either space or time using a similar calculation to that used for the vector projections. The quad-trees previously computed were used to divide up each average motion image into appropriately sized regions, from which features are collected.

3.3. Classification

At the next stage, once the features for a set of image sequences had been extracted, we used GentleBoost (GB) classifiers [39], an extension to the traditional AdaBoost classification algorithm, in order to simultaneously select the best features to use, and perform the training used for classification. We used two classifiers for each expression: one for the onset temporal segment, and the other the offset segment.

Target labels were created for each classifier by setting the labels for frames belonging to the temporal segment to be 1, and all other frames to be -1. These were used, along with the features matrix produced from each set of quad-trees, as input to the classifiers. At each iteration in the training algorithm, the classifier chooses a feature that reduces the error by the largest margin, and then stores this feature and the associated parameters. This continues until the error rate no longer reduces, or the maximum number of features is reached, here set to be 200.

Once the two classifiers had been fully trained they were used to test the same set of features. This produced a set of predicted labels for the frames in the training set, along with confidence levels for these labels. The labels and confidences were multiplied together to form a distribution of values suitable which were suitable as input for the HMMs.

The test set frames were then tested against these classifiers, in order to produce emission values. An example of the output from the two classifiers throughout a test Sad sequence can be seen in Fig. 5. Values above the zero x-axis indicate frames which are labelled as belonging to the corresponding segment, onset or offset.

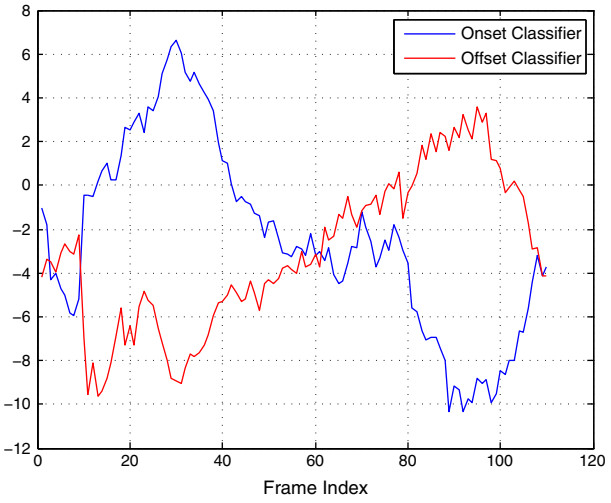

Fig. 5. Example of the onset and offset GB classifier outputs for a Sad sequence with window width 12.

3.4. Temporal modelling

We used HMMs in order to model the temporal dynamics of the entire expression. HMMs are a tool that have been well-used in the analysis of facial expression dynamics, though they are best known for their use in speech recognition. These were trained on the emission output from the GB classifiers which was formed by multiplying the labels and confidence values together.

We model a sequence which displays a full expression using four different temporal segments — neutral, onset, apex and offset. These form the basis for four possible states of the hidden variable in the HMM. The general form of the model for one expression can be seen in Fig. 6. This model allows transitions from each state to the next, as well as to itself, but also from apex back to onset, and from offset back to apex, to reflect the fact that for some expressions the subject can have multiple apexes. The actual transitions possible for each expression, the equivalent probabilities, are calculated from the labels in the training set, and so the latter two transitions may not be possible for all expressions. The model assumes that the expression will start in neutral or onset, progressing through all of the other three states, until finally returning to neutral. Hence only frames at the beginning and end of the sequence are labelled as neutral, and all other stationary frames in between are labelled as apexes. This is appropriate in these experiments, as the examples from the BU-4DFE used all contain this sequence.

The three sets of parameters of an HMM are:

- Initial probabilities — the probability distribution of the initial states across the image sequences.
- Transition probabilities — a matrix defining the probabilities of the different transitions between underlying states in the model.
- Emission probabilities — the conditional probability distribution defining how the observed values depend on the hidden states.

Each of these was determined from the results gathered from testing the trained classifiers. Let $\mathbf{L}$ be a matrix containing the state labels for the training set of frames, where each row corresponds to a different image sequence, and each column to a different frame index in this sequence. In practise this is stored as an array of cells as the image sequences are of different lengths and so contain different numbers of frames. In addition, let $\mathbf{E}^{on}$ and $\mathbf{E}^{off}$ be matrices containing the emission values produced by the onset and offset classifiers respectively. We computed the initial probability distribution, $\mathbf{P}$, by estimating the prior probabilities from the state labels of the first frame in each image sequence in the training set. The transition probability matrix, $\mathbf{T}$, was also be estimated from the state labels by using the frequency of each transition between states.

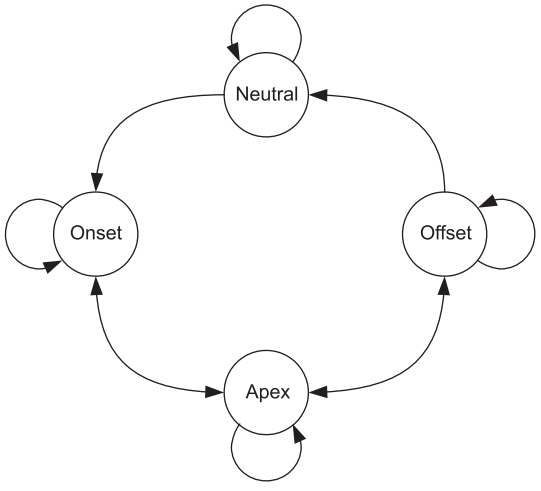

Fig. 6. The HMM transition model consisting of neutral, onset, apex and offset states and the transitions possible between them.

Finally the emission probability distribution must be calculated using the emission values and the labels. The distributions used were Gaussian, and so were represented by a mean, μ, and standard deviation, σ, for the possible emission values for each of the five possible states. Hence the distribution was represented by two matrices each with five rows corresponding to the five states, and two columns corresponding to the two classifiers, onset and offset. The mean matrix, $\mathbf{M}$, was calculated by averaging the emission values observed for each of the temporal states:

$$\mathbf{M}_{(1,s)} = \frac{1}{N_s} \sum_{(i,j)\in f(s)} \mathbf{E}^{on}_{(i,j)},$$

$$\mathbf{M}_{(2,s)} = \frac{1}{N_s} \sum_{(i,j)\in f(s)} \mathbf{E}^{off}_{(i,j)},$$

where N_s is the total number of frames in $\mathbf{L}$ with label s, and

$$f(s) = \{(i,j)|\mathbf{L}_{(i,j)} = s\}.$$

The standard deviation matrix, $\mathbf{S}$, can be calculated as:

$$\mathbf{S}_{(1,s)} = \sqrt{\frac{1}{N_s} \sum_{(i,j)\in f(s)} \left(\mathbf{E}^{on}_{(i,j)} - \mathbf{M}_{(1,s)}\right)^2},$$

$$\mathbf{S}_{(2,s)} = \sqrt{\frac{1}{N_s} \sum_{(i,j)\in f(s)} \left(\mathbf{E}^{off}_{(i,j)} - \mathbf{M}_{(2,s)}\right)^2}.$$

Once these properties of the HMM had been estimated from the training data, the model was ready to be used for testing new image sequences. Testing is conducted by collecting features from the new image sequence using the same quad-trees created from the training set, testing the classifiers on these features, and then using the observed values along with the standard Viterbi algorithm to determine the most likely sequence of states.

4. Experimental results

We conducted experiments using the BU-4DFE database [48]. This database consists of 4D data (3D plus time) collected by asking 100 subjects to act out the six basic expressions. The 3D data collected consists of the 2D image, with an added depth map showing the height of each point throughout the sequence. The image sequences available in the database were filtered to remove any expressions that were deemed to be inaccurate representations, or any sequences that did not start and end in the neutral expression. This resulted in the following numbers of examples for being used for each expression: Happy — 90, Sad — 58, Angry — 53, Disgust — 59, Surprise — 87 and Fear — 50.

The testing was done using 6-fold cross-validation. For each fold to be tested, a training set was created for each expression from the other 5 folds. The method employed for construction of a suitable training set involved taking all available positive examples from the folds, and then an equal number of negative examples by randomly selecting from the remaining expressions available. These training sets were used to train one classifier to model each expression.

The next stage in the testing was choosing a suitable window width for each expression. This was done by performing a validation test for each of the expressions and window widths. This test looked at the ability of each classifier to discriminate between positive and negative examples of the expression for which it has been trained. The window width which gave the best validation F_1-measure could then be chosen as the most suitable window width to use in the

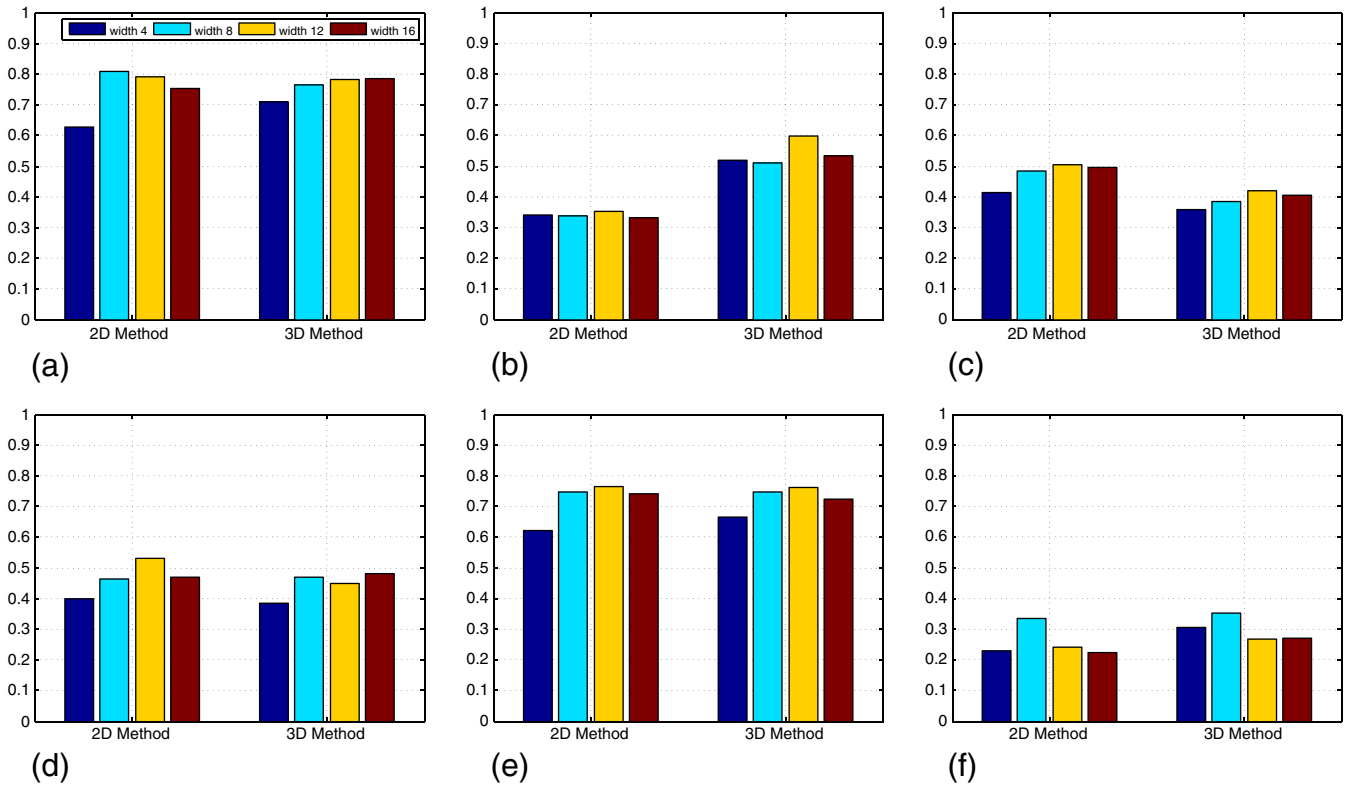

Fig. 7. Validation F_1-measures for each expression for all window widths. (a) Happy Validation F_1-measures. (b) Sad Validation F_1-measures. (c) Angry Validation F_1-measures. (d) Disgust Validation F_1-measures. (e) Surprise Validation F_1-measures. (f) Fear Validation F_1-measures.

six-way expression decision. The validation F_1-measures can be seen in Fig. 7 for each of the expressions. Initially, the window width chosen for each expression was automatically selected as being the one that produced the highest F_1-measure in the validation tests. Then, for expressions that had more than one window width that gave a similar F_1-measure, an alternative window width was manually chosen if it improved the results of the six-way classification process.

Then each of the sequences in the test set was tested against all six of the classifiers, and the outputs used to make a decision about which expression the sequence represented. The method for determining the predicted expression was as follows. Firstly, only the sequences for which one or more frames were labelled as the apex state were considered. Finally the most appropriate expression label was chosen by taking the expression for which the sequence containing the apex was most likely compared to an equivalent sequence with no apex.

An additional test was run in order to measure the benefit of using 3D facial geometries over 2D image sequences for facial expression recognition. Here the 2D facial intensities available from the BU-4DFE were used along with an adapted version of the system proposed in [21] which could model the full expression and make a six-way decision suited to recognition rather than a validation result which is more suitable for detection. The differences in the 2D tests as compared to 3D were: the alignment used between image sequences required manual eye detection as opposed to that used with the 3D method which was fully automatic. 2D FFDs were used to compute the motion between frames in each sequence. For feature extraction and classification similar methods were employed as in [21], and then HMMs were used to model the full expression as done for 3D. Hence a comparison between 2D and 3D facial expression analysis was possible.

4.1. Performance

The recall rate, precision rate, and F_1-measure, the balanced F-measure [51], were calculated for each expression. This was first conducted using the automatically selected window widths, and then using the manually optimised window widths which show a significant increase in the average F_1-measure achieved. The full results achieved, including average, can be seen in Table 1. This table shows that the recognition rates for the different expressions varies widely. The best result achieved is for Surprise, with an F_1-measure of 82.56%, with the lowest rate found for Fear, with an F_1-measure of only 46.15%. The average rate found was 64.46%.

The confusion matrix produced for the six universal expressions using the manually selected window widths can be seen in Table 2. This shows the percentage of each expression that was correctly classified, along with where the misclassifications occurred. The

Table 1
F_1-measures achieved with 2D and 3D testing. WW = Window width, RR = Recall rate, PR = Precision rate, and F_1 = F_1-measure. The best F_1 performance achieved for each expression is show in bold.

Expression	2D system				3D system			
	WW	RR	PR	F_1	WW	RR	PR	F_1
2*Automatic window width selection performance (%)								
Happy	8	76.40	77.27	76.84	16	71.91	84.21	**77.58**
Sad	12	44.83	42.62	43.70	12	70.69	57.75	**63.57**
Angry	12	61.54	58.18	**59.81**	12	48.08	45.45	46.73
Disgust	12	57.63	65.38	**61.26**	16	54.24	52.46	53.33
Surprise	12	79.07	78.16	78.61	12	79.07	85.00	**81.93**
Fear	8	38.46	37.50	37.97	8	43.59	42.50	**43.04**
Average		59.65	59.85	59.70		61.26	61.23	**61.03**
Manual window width selection performance (%)								
Happy	8	76.40	80.00	78.16	12	75.28	88.16	**81.21**
Sad	16	44.83	44.83	44.83	12	68.97	57.14	**62.50**
Angry	12	61.54	56.14	**58.72**	12	51.92	48.21	50.00
Disgust	12	50.85	68.18	58.25	8	62.71	66.07	**64.35**
Surprise	8	89.53	73.33	80.63	8	82.56	82.56	**82.56**
Fear	8	38.46	44.12	41.10	8	46.15	46.15	**46.15**
Average		60.27	61.10	60.28		64.60	64.72	**64.46**

Table 2
Confusion matrices for 2D and 3D testing. Recall rates for each expression are shown in bold.

	Happy	Sad	Angry	Disgust	Surprise	Fear
2D experimental results						
H	**76.40**	8.99	1.12	4.49	5.62	3.37
Sa	18.97	**44.83**	13.79	1.72	6.90	13.79
A	1.92	13.46	**61.54**	15.38	5.77	1.92
D	6.78	8.47	15.25	**50.85**	10.17	8.47
Su	0.00	4.65	3.49	0.00	**89.53**	2.33
F	2.56	20.51	10.26	2.56	25.64	**38.46**
3D experimental results						
H	**75.28**	2.25	1.12	6.74	4.49	10.11
Sa	1.72	**68.97**	17.24	5.17	3.45	3.45
A	0.00	28.85	**51.92**	11.54	1.92	5.77
D	3.39	5.08	16.95	**62.71**	5.08	6.78
Su	2.33	4.65	4.65	2.33	**82.56**	3.49
F	10.26	15.38	10.26	5.13	12.82	**46.15**

confusion matrix shows where the main errors are introduced. Fear is often classified as Surprise, which is an expected result due to the similarities in the way these two expressions are acted — subjects often stretch their mouths, raise their eyebrows and open their eyes in both cases. However in addition, it is also regularly misclassified as Sad. This could be due to creasing around the eyes in both expressions which gives some similarities. The main confusion for Angry comes from incorrect classification as Sad, though this expression is often misclassified as Disgust as well. This could be due to the similarities in the creases in the forehead for these three expressions, especially in examples where the subject does not have much movement in other areas of the face such as the mouth. Disgust is most often misclassified as Angry, and to a lesser extent Sad, showing again the similarities in these expressions as seen by the 3D system. Happy is misclassified as Fear most often, which could be explained by the fact that the corners of the mouth move horizontally outwards in several examples of fear in this database.

The window widths used for each expression are also shown in Table 1. These show that Happy, Sad and Angry perform best with a window width of 12, whereas the remaining three expressions give the best performance with a window width of only 8. In order to determine if these window widths are what we would expect, we compare these to the distributions of length of the onset and offset segments for each expression, as shown in Fig. 8. These plots show that the window width chosen in each case generally falls within one standard deviation of the mean of either the onset or offset mean, and in almost all cases it lies within this range for both segments. Only for Disgust do we see that the window width is below this range for the onset segment, and in this case it is only just within the range for the offset, though it does lie very close to the mode in this case. This plots show that the window widths chosen generally appear to be sensible compared to the onset, offset, or both, lengths.

4.2. Comparison to 2D

The F_1-measures achieved in the six-way decision when employing the 2D method, for both the automatic and manually selected window widths, can be seen in Table 1. The corresponding confusion matrix for the manually selected widths using the 2D method can be seen in Table 2. With both automatic and manual window width selection, the 3D system achieves a higher average F_1-measure than 2D: 61.03% compared to 59.70%, and 64.46% compared to 60.28%. In the manual case, for five of the expressions, Happy, Sad, Disgust, Surprise and Fear, the 3D method outperforms the 2D method, achieving a significant rise in F_1-measure. This is particularly striking for the Sad expression, which achieves a far higher F_1-measure with 3D features than with 2D, 62.5% compared to 44.83%. This improvement seen with the 3D data is in contrast to that seen purely from the validation results in Fig. 7. This suggests that the 3D system is not generally superior to 2D when distinguishing positive and negative examples for most of the expressions, with the notable exception of Sad, and to a lesser extent Fear. However, 3D information is beneficial when it comes to discrimination between the six expressions, which is demonstrated by the improvement in the F_1-measure seen in five out of six of the expressions.

The only expression for which the 2D method significantly outperforms the 3D is Angry, achieving an F_1-measure of 58.72% compared to 50.00%. Comparing the performance in the validation results, the 2D system does demonstrate a slightly better ability to discriminate between positive Angry examples and the other negative examples. This may be because the FFDs used in the 3D case are too coarse to pick up on the subtle motions in the forehead and around the eyes that are present in many of the Angry examples. However, the significant difference in the Angry results in the six-way decision are not accounted for solely by this. In addition, most of the misclassification of this expression seems to be due to the

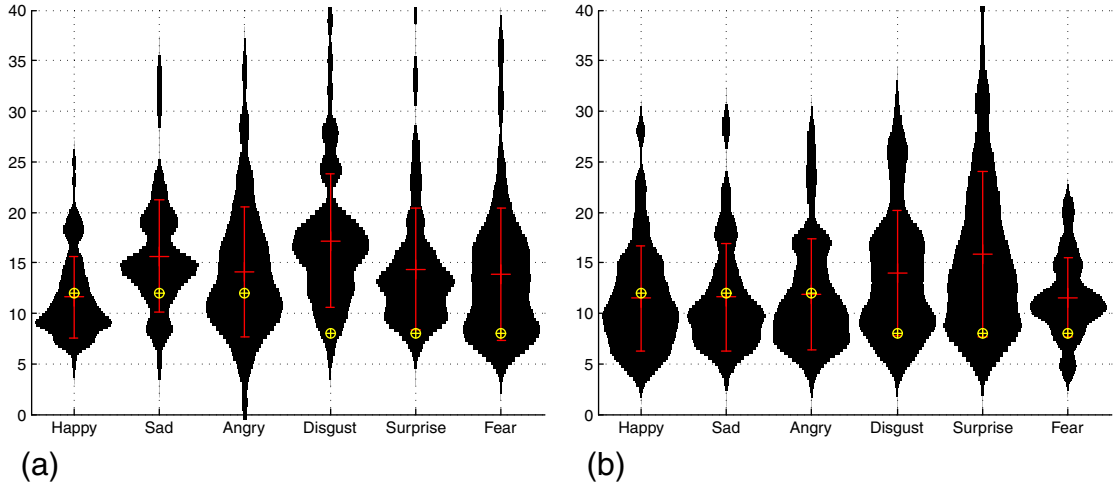

Fig. 8. Comparison between the chosen window widths and the onset and offset length distributions for each expression. Violin plots are used to show the number of sequences with each segment length, the mean is shown as a red cross at the centre of the red bar showing the standard deviation, and the window width chosen is shown as a yellow circled cross. (a) The onset length distributions for each expression. (b) The offset length distributions for each expression.

performance of Sad. Though the 2D system misclassifies a significant number of Angry sequences as Disgust, as happens in the 3D case, the difference here is that there is much less confusion between Angry and Sad. The reason for this may be due to the fact that in this case Sad performs very poorly for 2D. This suggests that the 3D process is able to distinguish features that are useful for recognition of the Sad expression that are not possible using 2D. This increases the recognition rate for Sad, but it could mean that these features are confused with those present in the Angry sequences, and so add to the confusion between these two expressions which results in a much lower classification rate for Angry.

4.3. Temporal comparison

The proposed method employs HMMs in order to temporally model the full expression, and to find the most likely sequence from the individual frame outputs from the GB classifiers. The aims of this process is to smooth errors in the GB classification when predicting the frame sequence, to ensure that the full sequence is present in order for any part of the sequence to be labelled, and then to require that the sequence is likely enough to be chosen from the classifier outputs. In addition, the likelihood of each of the sequences being predicted is also used to determine the best expression for each sequence.

The smoothing benefits of this method are demonstrated in Fig. 9. This figure shows positive and negative examples from the Sad classifier results. The graphs in Fig. 9a and b show the emission values from the two GB classifiers for a positive example, Sad, and a negative example, Fear, respectively. As has been previously stated, these emission values are formed from the labels for the frames, 1 or −1, being multiplied by the confidence value for these labels. Hence, whenever the plot becomes positive, this is due to the label changing from negative to positive classification. Alongside these, Fig. 9c and 9d show the true frame segment labelling for these sequences for the Sad classifier, Fig. 9e and 9f show the frame labels taken directly from each of the classifiers, with the apex frames inferred as filling in gaps between the onset and offset frames, and Fig. 9g and 9h show the labels predicted by the most likely HMM sequence when using the emission values from the classifiers.

The positive example demonstrates that it can be possible for the GB classifiers to repeatedly classify onset and offset frames throughout the sequence, as is shown in Fig. 9e, but the HMM takes only the main sections of the sequence where these classifications occur, which are where the emission values are at their highest, as part of the most likely sequence, Fig. 9g. Hence the effect is to smooth the labelling to something that is much closer to the true frame labels as shown in Fig. 9c. The negative example demonstrates the other benefit of the HMM. This time the GB classifiers classify some frames as onset and offset, as seen in Fig. 9f, even though none are present for the Sad expression in this sequence. This results in apex frames being inferred, so this expression would be considered to possibly be Sad if using the classifiers outputs alone. However, the HMM is able to smooth over these frames, choosing the most likely sequence as one that contains only neutral frames, as seen in Fig. 9h, due to the low emission values, and small number, of these frames. This means that this sequence would be rejected immediately as not being Sad.

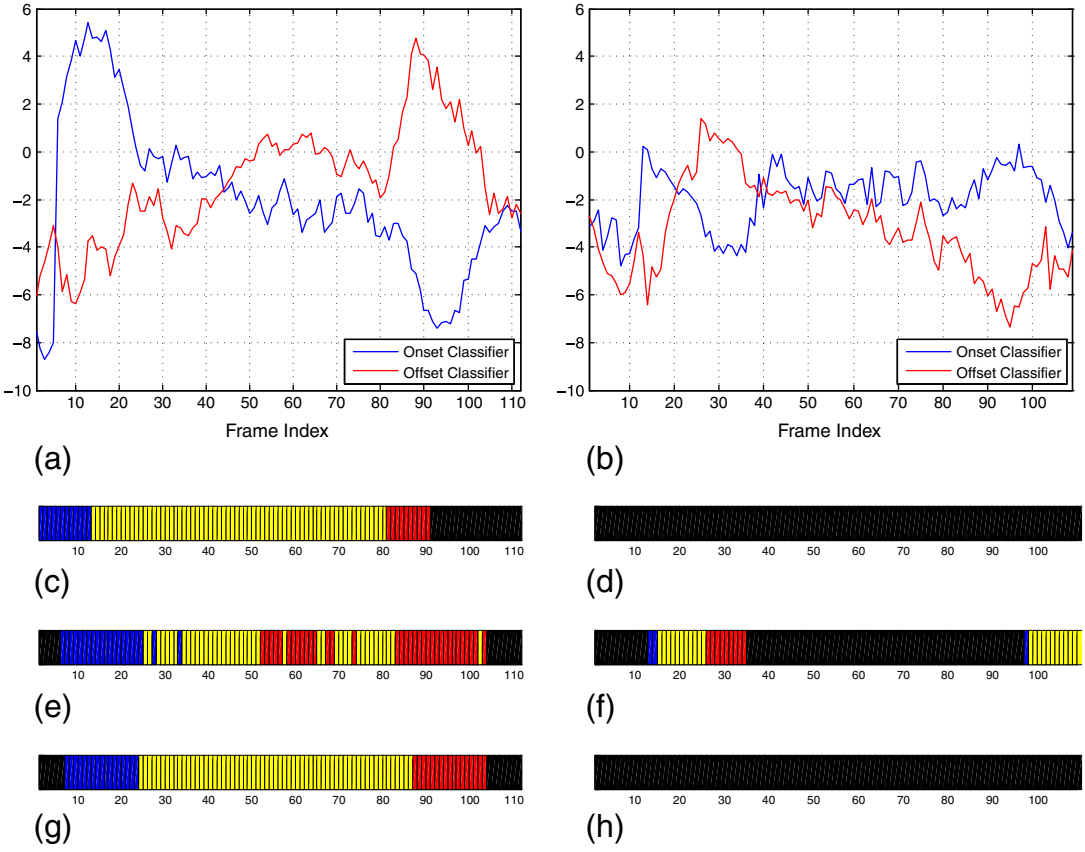

Fig. 9. The Sad GB classifier outputs and labels for positive and negative examples Frame labels: black — neutral, blue — onset, yellow — apex, red — offset. (a) Sad onset and offset classifier outputs for a Sad sequence. (b) Sad onset and offset classifier outputs for a Fear sequence. (c) True Sad classifier segment labels for Sad sequence. (d) True Sad classifier segment labels for Fear sequence. (e) Sad GB classifiers predicted labels for Sad sequence. (f) Sad GB classifiers predicted labels for Fear sequence. (g) Sad HMM classifier predicted labels for Sad sequence. (h) Sad HMM classifier predicted labels for Fear sequence.

In addition to looking at particular examples of the benefit of using the HMMs for temporal modelling, it is also possible to do an analysis of the expression classification differences between using the GB classifier outputs directly, from the HMM classifier outputs. Due to the way the six-way decision is made from the sequence likelihoods, and the fact there is no equivalent probability measure to use for the GB classifiers, it is not possible to perform a fair comparison between six-way expression classifications directly. However, one way to compare the performance of each method is to assess the percentage of sequences that are predicted to contain the apex state in each case. This would be the first stage in each classification process, where those expressions for which the sequence does not display the apex state are rejected. Here we look at the desired result for positive and negative examples for each expression: positive examples should contain the apex, and negative examples should not contain the apex state. The results of this analysis for a window width of 12 can be seen in Table 3. These results show that the GB classifiers generally show the apex for a higher percentage of positive examples than the HMM classifier — 92.4% compared to 80.4%. However, this is at a cost, as the negative rates show. The GB classifiers also give the apex state in a large number of negative sequences, with only 36.0% of sequences not containing the apex state. This is compared to 72.6% when using the smoothed HMM output. This results in the HMM process giving a much higher average percentage of 76.5% compared to the average GB result of 64.2%. This demonstrates that the full HMM classifier allows better discrimination between positive and negative examples purely on the basis of whether the apex is present or not, and that is before the likelihoods of the different expressions are taken into account.

5. Discussion and future work

This method has been demonstrated to exploit the extra information available in the 3D facial geometries to improve on the results found with the 2D, and to use temporal modelling to smooth over incorrect classifications from the GB classifiers in order to correctly classify image sequences. The approach proposed in this paper has been shown to achieve an improvement over the equivalent 2D method when tested on the BU-4DFE database. However, in order to prove that these improvements are repeatable it would have been desirable to test the method on other databases. But, as the BU-4DFE database is the only dynamic 3D database currently publicly available, this was not possible. In future, it is expected that more databases will become available, allowing validation testing to be done on several datasets.

The results show that there are a number of image sequences will are still incorrectly classified by this method. The main way in which the method fails is when an expression is classified as positive by two or more expressions. This can occur due to the onset/offset being incorrectly detected by some of the GB classifiers, and the expression which the highest likelihood is then one of the incorrect expressions. This may happen due to variability in the motion during the onset/offsets of some of the expressions, and the GB classifiers not capturing this variability adequately. Another failure mode is when the onset or offset for an expression is not detected by the appropriate classifier, and so another expression is deemed most likely. These problems occur both in the 2D and 3D methods, and demonstrate the GB classifiers are not capable of fully capturing the variability of the onset and offsets for all the expressions. Throughout the database, the same expression can be acted in very different ways by different subjects, hence giving a wide range of features that are particular to the onset or offset of an expression, and this causes these problems. Even within the onset of an expression by one subject, there can be different stages (e.g. the upper face and lower face moving separately). In order to deal with this problem a more complex model would be required that is able to capture the different aspects of the onset or offset of an expression. However, the complexity required would quickly make this method unwieldy, especially as more expressions were added to the recognition process. For this reason in future work we would prefer to move towards AU detection rather than full expression analysis.

Table 3
Percentage of positive/negative sequences containing/not containing the apex state using the GB classifier outputs directly compared to the HMM outputs for a window width of 12. Values in bold show the best performing method for each category of sequences.

Method	Pos	Neg	Average
GBs	**92.4**	36.0	64.2
GBs + HMM	80.4	**72.6**	**76.5**

AUs are well defined due to their anatomical basis, and so vary far less between subjects. They are far better suited to the neutral–onset–apex–offset model, and so would be better captured by this method. In addition, due to the number of AUs being finite, it would be possible to adapt this system to cover all of the possible AUs, and then use higher level methods to perform expression recognition from these actions. A subset of the AUs was focussed on in [21] in this way, and this work would be able to build on this to detect more AUs, including those that are particular hard to detect in 2D, such as AU18 (Lip Pucker), AU29 (Jaw Thrust) and AU31 (Jaw Clencher). This approach would also have the added advantage that AUs are much more consistent as regards the length of time taken for onset and offset, and so would be expected to have window widths which are clearly best suited for each AU. This would eradicate the issue of automatic versus manual selection of window width that was seen in this paper, due to the wide variation seen in the onset and offset lengths across the expressions (as seen in Fig. 8). So far there is no publicly available data that contains dynamic examples of the AUs, and hence experiments of this kind have not yet been possible.

There are other extensions to this method that would be desirable in future work. Posed expressions, as employed in these experiments, have been shown to differ greatly, in content and dynamics, from spontaneous natural expression data. Spontaneous expressions are also rarely seen on their own; subjects often display a mixture of emotions, such as amusement and embarrassment, or sadness and anger. In addition, one expression can transition to another without a neutral expression in between, and expressions can be mixed with periods of speech. In order to create a system that will be useful in real-life situations, it is therefore highly desirable to train and test this system on this kind of data, and to adapt the models used in order to do so. However, the 3D facial expression databases currently available contain only posed examples, hence these experiments are currently not possible. The collection of more 4D databases, including spontaneous expressions of emotion, is an open area of research in this field. Finally, an additional area of research in 3D dynamic facial expression analysis is continuous expression modelling, as opposed to discrete recognition as is carried out in this paper. However, currently this is also not possible due to no 3D databases being currently available that contain continuous annotation.

6. Conclusions

In this paper we exploit the facial geometry data in the BU-4DFE database in order to perform dynamic analysis of the six universal expressions for the purpose of fully automatic expression recognition. The methodology used employed 3D motion-based features, captured with FFDs, which were captured in each pair of dimensions, spatial and time. Best features were chosen and classified by GB classifiers, and the output of these was used to build temporal models of each expression using an HMM. Six-way classification was conducted using all sequences in the database deemed to accurately reflect the expression, with a classifier being trained and tested for each expression in each fold. Window widths were chosen based on the validation results for each expression.

The results were compared with the same method (using manual alignment) conducted on 2D facial motion data extracted from the facial intensity image sequences in the same database. The expression recognition rates achieved indicate that there is a gain when using 3D facial geometry data, and that the 3D data is particularly important for correct classification of the Sad expression. In addition, temporal analysis indicates that modelling the full dynamics of the expression with the HMMs leads to a higher recognition rate of the expressions than using the GB classifier outputs alone.

Acknowledgements

This work has been funded by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB). The work of Georgia Sandbach is funded by the Engineering and Physical Sciences Research Council (EPSRC) through a Doctoral Training Account Studentship. The work of S. Zafeiriou was funded in part by the Junior Research Fellowship of Imperial College London.

References

[1] M. Pantic, Machine analysis of facial behaviour: naturalistic and dynamic behaviour, Philosophical Transactions of the Royal Society B: Biological Sciences 364 (1535) (2009) 3505.
[2] Z. Ambadar, J. Schooler, J. Cohn, Deciphering the enigmatic face, Psychological Science 16 (5) (2005) 403–410.
[3] M. Valstar, H. Gunes, M. Pantic, How to distinguish posed from spontaneous smiles using geometric features, Proceedings of the 9th international conference on Multimodal interfaces, ACM, 2007, pp. 38–45.
[4] A. Williams, Facial expression of pain: an evolutionary account, Behavioral and brain sciences 25 (04) (2002) 439–455.
[5] M. Costa, W. Dinsbach, A. Manstead, P. Bitti, Social presence, embarrassment, and nonverbal behavior, Journal of Nonverbal Behavior 25 (4) (2001) 225–240.
[6] P. Ekman, E. Rosenberg, What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS), Oxford University Press, USA, 2005.
[7] D. I. Ltd, www.di3d.com.
[8] I. Cohen, N. Sebe, A. Garg, L. Chen, T. Huang, Facial expression recognition from video sequences: temporal and static modeling, Computer Vision and Image Understanding 91 (1–2) (2003) 160–187.
[9] S. Gokturk, J. Bouguet, C. Tomasi, B. Girod, Model-based face tracking for view-independent facial expression recognition, Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, IEEE, 2002, pp. 287–293.
[10] N. Sebe, M. Lew, Y. Sun, I. Cohen, T. Gevers, T. Huang, Authentic facial expression analysis, Image and Vision Computing 25 (12) (2007) 1856–1863.
[11] S. Mpiperis, S. Malassiotis, M.G. Strintzis, Bilinear decomposition of 3D face images: an application to facial expression recognition, 10th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2009), 2009.
[12] H. Soyel, H. Demirel, 3d facial expression recognition with geometrically localized facial features, Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on, IEEE, 2008, pp. 1–4.
[13] A. Savran, B. Sankur, Automatic detection of facial actions from 3d data, Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1993–2000.
[14] H. Tang, T. Huang, 3D facial expression recognition based on automatically selected features, Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, IEEE, 2008, pp. 1–8.
[15] J. Wang, L. Yin, X. Wei, Y. Sun, 3D facial expression recognition based on primitive surface feature distribution, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, IEEE, 2006, pp. 1399–1406.
[16] L. Yin, X. Wei, P. Longo, A. Bhuvanesh, Analyzing facial expressions using intensity-variant 3D data for human computer interaction, Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, Vol. 1, IEEE, 2006, pp. 1248–1251.
[17] F. Tsalakanidou, S. Malassiotis, Real-time 2D + 3D facial action and expression recognition, Pattern Recognition 43 (5) (2010) 1763–1775.
[18] Y. Chang, M. Vieira, M. Turk, L. Velho, Automatic 3D facial expression analysis in videos, Analysis and Modelling of Faces and Gestures (2005) 293–307.
[19] Y. Sun, L. Yin, Facial expression recognition based on 3D dynamic range model sequences, Computer Vision—ECCV 2008, 2008, pp. 58–71.
[20] D. Rueckert, A. Frangi, J. Schnabel, Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration, IEEE Transactions on Medical Imaging 22 (8) (2003) 1014–1025.
[21] S. Koelstra, M. Pantic, I. Patras, A dynamic texture-based approach to recognition of facial actions and their temporal models, IEEE transactions on pattern analysis and machine intelligence (2010) 1940–1954.
[22] I. Kotsia, I. Pitas, Facial expression recognition in image sequences using geometric deformation features and support vector machines, IEEE Transactions on Image Processing 16 (1) (2007) 172–187.
[23] M. Pantic, I. Patras, Detecting facial actions and their temporal segments in nearly frontal-view face image sequences, Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics, 2005, pp. 3358–3363.
[24] M. Pantic, I. Patras, Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences, IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 36 (2) (2006) 433.
[25] M. Valstar, M. Pantic, Combined support vector machines and hidden markov models for modeling facial action temporal dynamics, HCI'07: Proceedings of the 2007 IEEE International Conference on Human–Computer Interaction, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 118–127.
[26] Y. Tong, W. Liao, Q. Ji, Facial action unit recognition by exploiting their dynamic and semantic relationships, IEEE Transactions on Pattern Analysis and Machine Intelligence (2007) 1683–1699.
[27] Y. Zhang, Q. Ji, Active and dynamic information fusion for facial expression understanding from image sequences, IEEE Transactions on Pattern Analysis and Machine Intelligence (2005) 699–714.
[28] S. Zafeiriou, I. Pitas, Discriminant graph structures for facial expression recognition, IEEE Transactions on Multimedia 10 (8) (2008) 1528–1540.
[29] C. Shan, S. Gong, P. McOwan, Facial expression recognition based on Local Binary Patterns: a comprehensive study, Image and Vision Computing 27 (6) (2009) 803–816.
[30] Y. Chang, C. Hu, M. Turk, Probabilistic expression analysis on manifolds, Computer Vision and Pattern Recognition, 2004. CVPR 2004, Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 2, IEEE, 2004.
[31] S. Koelstra, M. Pantic, Non-rigid registration using free-form deformations for recognition of facial actions and their temporal dynamics, International Conference on Automatic Face and Gesture Recognition, Vol. 1, Citeseer, 2008.
[32] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, D. Hawkes, Nonrigid registration using free-form deformations: application to breast MR images, IEEE Transactions on medical imaging 18 (8) (1999) 712–721.
[33] M. Yeasin, B. Bullot, R. Sharma, From facial expression to level of interest: a spatio-temporal approach, Computer Vision and Pattern Recognition, 2004. CVPR 2004, Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 2, 2004, pp. II-922–II-927http://dx.doi.org/10.1109/CVPR.2004.1315264, doi:10.1109/CVPR.2004.1315264.
[34] M. Yeasin, B. Bullot, R. Sharma, Recognition of facial expressions and measurement of levels of interest from video, IEEE Transactions on Multimedia 8 (3) (2006) 500–508.
[35] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, J. Movellan, Dynamics of facial expression extracted automatically from video, Image and Vision Computing 24 (6) (2006) 615–625.
[36] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Transactions on Pattern Analysis and Machine Intelligence (2007) 915–928.
[37] L. Gralewski, N. Campbell, I. Penton-Voak, Using a tensor framework for the analysis of facial dynamics, Automatic Face and Gesture Recognition, 2006. FGR 2006, 7th International Conference, IEEE, 2006, pp. 217–222.
[38] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, J. Movellan, Recognizing facial expression: machine learning and application to spontaneous behavior, Computer Vision and Pattern Recognition, 2005. CVPR 2005, IEEE Computer Society Conference, Vol. 2, IEEE, 2005, pp. 568–573.
[39] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), The Annals of Statistics 28 (2) (2000) 337–407.
[40] H. Soyel, H. Demirel, Facial expression recognition using 3D facial feature distances, Image Analysis and Recognition (2007) 831–838.
[41] P. Wang, C. Kohler, F. Barrett, R. Gur, R. Verma, Quantifying facial expression abnormality in schizophrenia by combining 2D and 3D features, Computer Vision and Pattern Recognition, 2007. CVPR'07, IEEE Conference, IEEE, 2007, pp. 1–8.
[42] I. Mpiperis, S. Malassiotis, M. Strintzis, Bilinear elastically deformable models with application to 3D face and facial expression recognition, Automatic Face & Gesture Recognition, 2008. FG'08, 8th IEEE International Conference, IEEE, 2008, pp. 1–8.
[43] I. Mpiperis, S. Malassiotis, M. Strintzis, Bilinear models for 3-D face and facial expression recognition, Information Forensics and Security, Transactions on IEEE 3 (3) (2008) 498–511.
[44] S. Ramanathan, A. Kassim, Y. Venkatesh, W. Wah, Human facial expression recognition using a 3D morphable model, Image Processing, 2006 IEEE International Conference on, IEEE, 2007, pp. 661–664.
[45] A. Savran, B. Sankur, M. Bilge, Facial action unit detection: 3D versus 2D modality, Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, IEEE, 2010, pp. 71–78.
[46] A. Savran, B. Sankur, Non-rigid registration of 3D surfaces by deformable 2D triangular meshes, Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, IEEE, 2008, pp. 1–6.
[47] Z. Wen, T. Huang, Capturing subtle facial motions in 3D face tracking, Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, IEEE, 2008, pp. 1343–1350.
[48] L. Yin, X. Chen, Y. Sun, T. Worm, M. Reale, A high-resolution 3D dynamic facial expression database, Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on, IEEE, 2009, pp. 1–6.
[49] F. Tsalakanidou, S. Malassiotis, Robust facial action recognition from real-time 3D streams, Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on, IEEE, 2009, pp. 4–11.
[50] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, International Journal of Computer Vision 13 (2) (1994) 119–152.
[51] C.J. van Rijsbergen, Information Retrieval, 2nd Edition Butterworths, London, 1979.