A Dynamic Approach to the Recognition of 3D Facial Expressions and Their Temporal Models

Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic and Daniel Rueckert

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 modeled 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 a Hidden Markov Model in order to model the full temporal dynamics of the expression. The proposed fully automatic system was tested in a subset of the BU-4DFE database for the recognition of happiness, anger and surprise. Comparisons with a similar 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.**

I. 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.

Expression dynamics are of great importance for the interpretation of human facial behavior [14]. They convey cues for behavior interpretation [1], and are useful for distinguishing between spontaneous and posed emotional expressions[25]. In addition, they are essential for the recognition of complex states such as pain and mood, [29], 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.

The majority of existing expression recognition systems are based on 2D static images (either a single image [4] or several static images [2], [32] taken from image sequences). A full review of such systems can be found in [14]. In addition, several methods have used the temporal information from facial expressions in various ways. More specifically, research has been conducted to model the temporal relationships of different expressions [33], to encode the full image sequence into a feature vector or tensor [9], [35] and to recognize particular temporal segments of the facial expression [12]. However, none of these aim to model the temporal behavior of the expression and use this information for recognition. More recently several systems have been developed that explicitly recognize and model the temporal segments of either full expressions [3], [11], [12], or components of expression such as facial action units (AUs) [10], [15], [23], [26]. These systems make use of complete 2D image sequences, and track the motion between frames, using either feature-based or appearance-based methods, in order to perform classification and modeling.

The methods and systems outlined so far are just a small subset of the many proposed for automatic facial expression and 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 [13].

The use of 3D facial geometry data and extracted 3D features for expression recognition is at its infant stage. 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 [4], [8], [19]. However these methods are susceptible to the problems of illumination and pose inherent to all 2D methods.

Recently, several methods have been proposed that use 3D facial geometry data for facial expression recognition [20], [18], [22], [24], [28], [31]. One of the first methods for 3D facial expression analysis was proposed in [31]. A deformable model was used for tracking the changes between

Georgia Sandbach, Stefanos Zafeiriou and Daniel Rueckert are with Imperial College London, 180 Queen's Gate, London SW7 2AZ, United Kingdom {georgia.sandbach09,s.zafeiriou, d.rueckert}@imperial.ac.uk

Maja Pantic is with Imperial London, 180 Queen's Gate, London SW7 2AZ, United Kingdom and with University of Twente, 7500 AE Enschede, NL. m.pantic@imperial.ac.uk

frames. A static 3D facial expression recognition method was proposed in [28]. This method used a 3D primitive surface feature labelling for the classification of the six basic expressions. In [20] researchers used six characteristic distances between feature points to train a multilayered perceptron for classification of the seven basic expressions. The method proposed in [22] is also based on distances between facial points, but this time AdaBoost classification was used for classification of the features. The work in [18] reduces the 3D facial geometry to 2D curvature images which could be used for feature extraction via Gabor wavelets, before classification using various techniques. The work in [24] tracks the movement in the face by using an active shape model to represent pairs of 2D and 3D images and identifying points which correspond to the salient facial features. Rule based approaches were used for classification, with a previous frame's information able to influence the classification of the current frame; in this way, temporal information was exploited during classification.

Although several of these methods exploited the motion between frames, and even some temporal information, none of them aimed to model the temporal dynamics of the expression for recognition purposes. The only method proposed so far that exploits 3D facial expression dynamics in this way is [21]. In this work a deformable model was applied and its changes were tracked to extract geometric features. Dimensionality reduction was applied via Linear Discriminant Analysis (LDA), followed by the use of 2-dimensional Hidden Markov Model (HMM) to model the spatial and temporal relationships between the features. The method in [21] requires manual detection and annotation of certain facial landmarks.

In this paper we propose a fully automatic method for facial expression recognition that exploits the dynamics of 3D facial motion. The system developed 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) [16]. We extract features by applying a quad-tree decomposition of the x-y, x-z, y-z, $x-t$, $y-t$ and $z-t$ motion fields. Features are then collected using a GentleBoost feature selection method for the onset and offset temporal segments of the expression and frame classification. Temporal modeling of the full expression is performed via neutral-onset-apex-offset HMM models. These models are finally 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 latter one.

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

- An extension of the method proposed in [10] to perform expression recognition using 3D facial geometry information.
- Modeling of the temporal segments of the full expression rather than those of action units.

To the best of our knowledge, this is the first fully automatic approach for dynamic 3D facial expression recognition.

II. METHODOLOGY

An overview of our system can be seen in Fig. I. In the preprocessing stage, the 3D meshes in each frame are aligned to a reference frame using an ICP method [34], and then cropped. The 3D motion is captured from each set of frames via FFDs [16], 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.

A. Motion Extraction

The motion between each frame in each image sequence was captured using FFDs. FFDs [17] is a method for non-rigid registration based on B-spline interpolation between a lattice of control points. Our aim is given two meshes with vertices $\mathbf{p} = (x, y, z)$ and $\acute{\mathbf{p}} = (\acute{x}, \acute{y}, \acute{z})$, respectively to find a vector field given by $\mathbf{T}(\mathbf{p})$ such that:

$$\acute{\mathbf{p}} = \mathbf{T}(\mathbf{p}) + \mathbf{p}. \tag{1}$$

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 $\phi(i, j, k)$ 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 (x, y, z), let the closest control point have coordinates (x_0, y_0, z_0) and displacement $\phi_\delta(i, j, k)$. The transformation of this point can be given as the B-spline interpolation of the 64 closest control points:

$$\mathbf{T}(x, y, z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_{a_1,a_2,a_3} \phi_\delta(i + l, j + m, k + n) \tag{2}$$

where $a_1 = x - x_0$, $a_2 = y - y_0$, $a_3 = z - z_0$, $B_{a_1,a_2,a_3} = B_l(a_1)B_m(a_2)B_n(a_3)$, and B_l is the l^{th} basis function of uniform cubic B-spline, defined as follows:

$$B_0(a) = \tfrac{1}{6}(-a^3 + 3a^2 - 3a + 1)$$

$$B_1(a) = \tfrac{1}{6}(3a^3 + 6a^2 + 4)$$

$$B_2(a) = \tfrac{1}{6}(-3a^3 + 3a^2 + 3a + 1)$$

$$B_3(a) = \tfrac{1}{6}a^3.$$

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

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 $0.5mm$ is used. Fig.

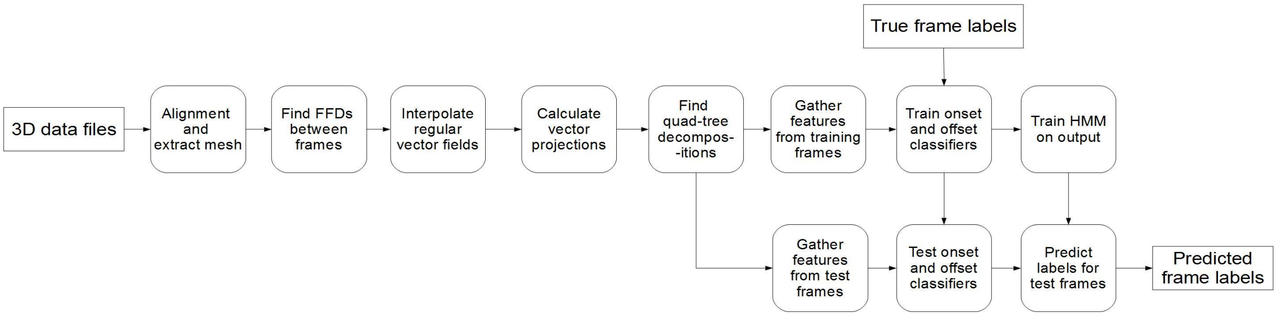
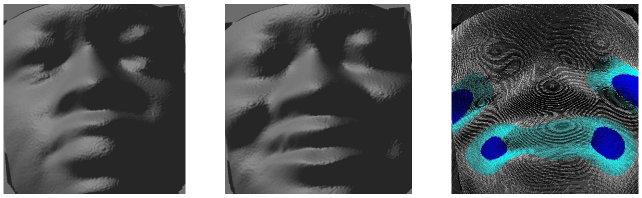

True frame labels
3D data files
Alignment and extract mesh
Find FFDs between frames
Interpolate regular vector fields
Calculate vector projections
Find quad-tree decompos- -itions
Gather features from training frames
Train onset and offset classifiers
Train HMM on output
Gather features from test frames
Test onset and offset classifiers
Predict labels for test frames
Predicted frame labels

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



(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

Fig. 2. 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.

2 shows a neutral and apex mesh for a happiness expression, and the motion tracked by FFDs between these frames. The most highly concentration areas of motion are around the corners of the mouth and the cheeks, as is expected for this expression.

B. 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, across the different axes (x, y, z and t), were computed. Furthermore, in order to focus only on the areas in which the greatest amount of 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.

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 i^{th} image sequence, M is the total number of image sequences of the current expression in the training set, and

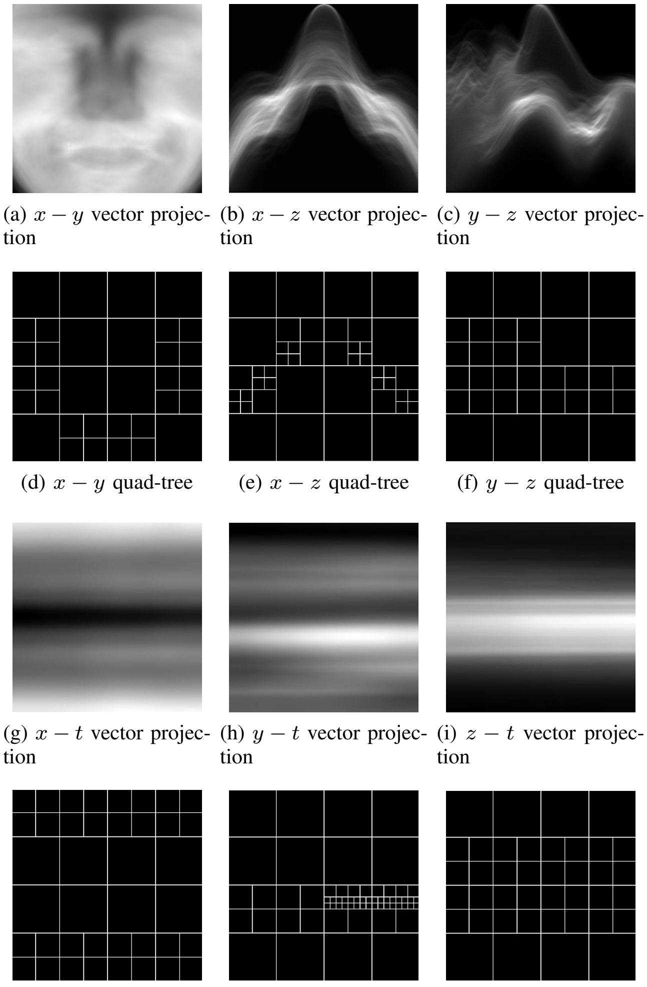
$$u_{i,x,y,z,t} = u^i(x,y,z,t),$$

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

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

are the vector components, in the x, y and z directions respectively, at coordinates (x,y,z) and time t in the i^{th} 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 $0 \leq t \leq 2\theta - 1$ as follows, using only the vector component in the spatial direction applicable:



(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

Fig. 3. 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

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

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

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

Examples of vector projections can be seen in Figs. 3a-3c and Figs. 3g-3i, 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.

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 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 used was 70% 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.

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 size gives 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. 3 can be seen in Figs. 3d-3f and Figs. 3j-3l. For example, Fig. 3e shows the decomposition created by dividing the vector projection in Fig. 3b 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) 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 2D 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 were 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,

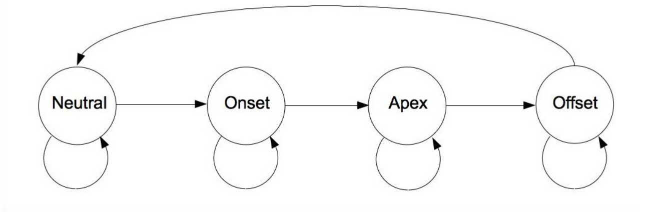


Fig. 4. Models a sequence consisting of neutral, onset, apex and offset states, each able to transition to the next.

from which features are collected.

C. Classification

At the next stage, once the features for a set of image sequences had been extracted, we used GentleBoost classifiers [7], 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 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. These were multiplied together to form a to distribution of values suitable which were suitable as an input for the HMMs.

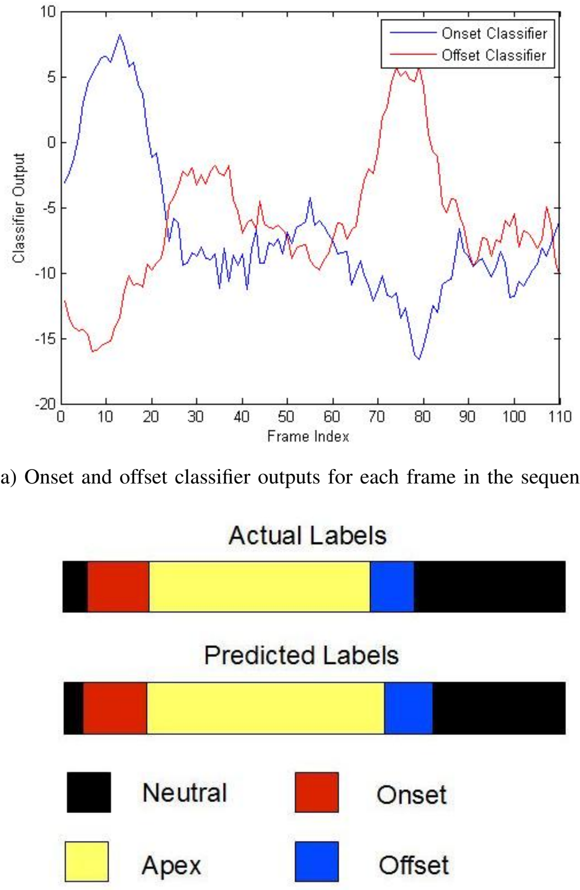
D. Temporal Modeling

We used HMMs in order to model the temporal dynamics of the entire expression. These were trained on the output from the GentleBoost classifiers.

We model a sequence which displays using four different temporal segments - neutral, onset, apex and offset. These form the basis for four possible states of the hidden variable. The general form of the model for one expression can be seen in Fig. 4.

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.



(a) Onset and offset classifier outputs for each frame in the sequence.

(b) Actual and predicted frame labels for this image sequence.

Fig. 5. The two classifier outputs for a Happy image sequence when using a window width of 8 with the actual and predicted frame labels from the HMM.

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.

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:

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

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

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. This is done by collecting features from the new image sequence using the same quadtrees 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. An example of the output from the two classifiers, and the resulting sequence chosen as most likely by the HMM can be seen in Fig. 5. Here the actual frame labels, and predicted frame labels are shown for comparison. The image sequence is then classified as a positive example if the apex state is present in the sequence.

III. EXPERIMENTAL RESULTS

We conducted experiments using the BU-4DFE database [30]. 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. In these preliminary tests, the system described above was tested on three expressions: happiness, anger and surprise. The happiness and anger expressions were chosen for testing purposes because they are at either ends of the valence expression spectrum, and surprise was also chosen as it is at one extreme of the arousal expression spectrum.

For the purpose of these experiments, datasets were created for each expression that was to be tested. A subset of subjects who were decided to be accurately acting out the expression was chosen from the database. A balanced dataset was then constructed by taking equal numbers of postive and negative sequences for the expression to be tested from these subjects, where the expressions used for the negative examples were selected randomly.

TABLE I

RESULTS FROM 2D AND 3D METHODS. CR = CLASSIFICATION RATE, RR = RECALL RATE, PR = PRECISION RATE

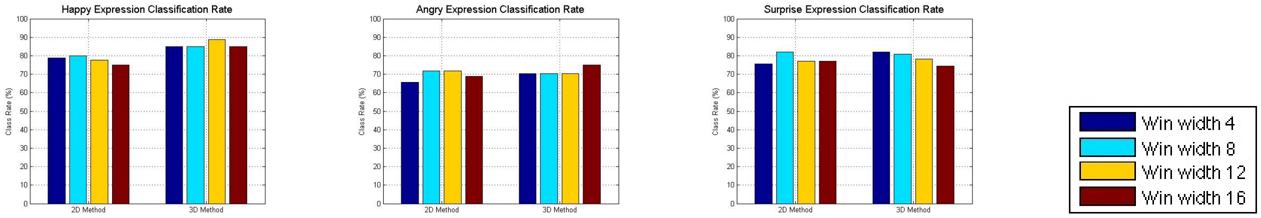
2D Method					
Expression	Win Size	Frame Results (%)			
		CR	RR	PR	F_1
Happy	4	76.27	61.38	84.32	**71.05**
Angry	8	67.07	45.67	72.40	**56.01**
Surprise	8	76.77	60.68	81.00	**69.38**
Average		**73.37**	**55.91**	**79.24**	**65.48**
Expression	Win Size	Expression Results (%)			
		CR	RR	PR	F_1
Happy	8	80.00	80.00	84.19	**82.04**
Angry	8	71.88	71.88	79.17	**75.34**
Surprise	8	82.05	82.05	86.79	**84.36**
Average		**77.98**	**77.98**	**83.38**	**80.58**
3D Method					
Expression	Win Size	Frame Results (%)			
		CR	RR	PR	F_1
Happy	4	80.18	69.58	80.28	**74.55**
Angry	4	65.35	44.76	70.70	**54.82**
Surprise	4	75.29	57.79	79.29	**66.85**
Average		**73.61**	**57.38**	**76.76**	**65.41**
Expression	Win Size	Expression Results (%)			
		CR	RR	PR	F_1
Happy	12	88.75	88.75	89.37	**89.06**
Angry	16	75.00	75.00	77.71	**76.33**
Surprise	4	82.05	82.05	85.40	**83.69**
Average		**81.93**	**81.93**	**84.16**	**83.03**

Verification of the classification system was performed using a 10-fold cross-validation testing, where the dataset was divided by subject into training and test sets, leaving four subjects out in each fold. Four measures of performance were recorded: the frame classification rate and F_1-measure, the balanced F-measure [27], and the expression classification rate and F_1-measure. The expression was determined to be present if one or more frames in the sequence were labeled as apex.

A. Performance

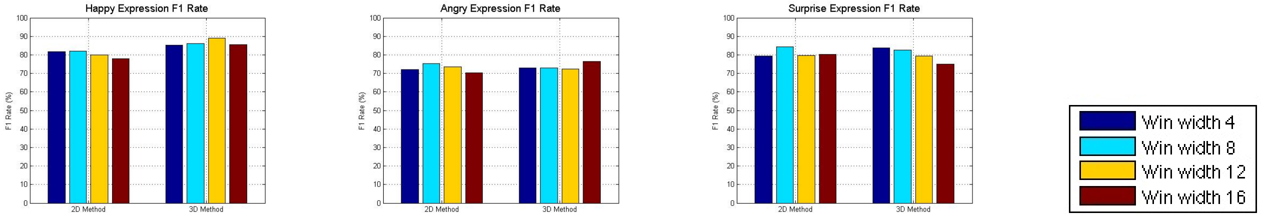
Table I shows the verification performance of the system for the three expressions happiness, anger and surprise. The performance is measured in two ways: by the F_1 measure when comparing the predicted frame labels to the actual frame labels, and the F_1 measure when the expression labeling is instead considered. Only the window width that achieved the best F_1 measure for each expression is included in this table. The average correct classification results achieved with our method are 73.61% for individual frames and 81.93% for expressions. The F_1 measures achieved are 65.41% and 83.03% for frames and expressions, respectively.

It is clear from all four measures that the happiness expression is the most accurately recognized expression, with a frame F_1 measure of 74.55%, and an expression F_1 measure of 89.06%. This was expected as happiness is the expression which consists of the largest motions, and the one which is acted in the most consistent manner in this database. Angry is the expression which resulted in the lowest F_1 measure for both frames and expressions, with



(a) Happy Expression Classfication Rate (b) Angry Expression Classfication Rate (c) Surprise Expression Classfication Rate (d) Legend

Fig. 6. Expression classification rates using both 2D and 3D methods with four different window widths.



(a) Happy Expression F_1 Measures (b) Angry Expression F_1 Measures (c) Surprise Expression F_1 Measures (d) Legend

Fig. 7. Expression F_1 measures using both 2D and 3D methods with four different window widths.

values of 54.92% and 76.33% respectively. This was again expected as the Angry expression consists of more subtle motions (which may have been removed by the smoothing of the meshes). Surprise achieves F_1 measures of 66.85% and 83.69% respectively for frames and expressions, rates that are between those of Happy and Angry. This was expected as this expression consists of significant movements of the face and is consistently acted in this database.

In all three expressions the frame classification rates and F_1 measures are considerably lower than those of the expression rates and F_1 measures. This was an expected outcome, and demonstrates that there are significant timing errors present in the predicted sequences produced by the classification system, but these often do not prevent the expression being correctly recognized. As the onset and offset segments of the expression can start and end gradually it is not surprising that the classifier will misclassify one or more frames at the beginning and end of these segments, but still correctly choose the path containing the apex segment.

The window width which gave the best performance differed between expressions, suggesting that the optimal window width varies based on the expression which is being recognised. However, the results here show some discrepancy between those favoured by the frame and expression rates, and also between the 2D and 3D experiments, which suggests that further testing would be required before a conclusive decision could be made on the optimal window width for each expression.

B. Comparison to 2D Method

In order to measure the benefit of using 3D facial geometries over 2D image sequences for facial expression recognition, the 2D facial intensities available from the BU-4DFE were used. The differences in these tests 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 lines as in [10] were followed. Hence a comparison between 2D and 3D facial motion is feasible. The results using the 2D method can also be seen in Table I. The average results for 2D were correct classification rates of 73.37% and 77.98% for frames and expressions, respectively. This corresponds to a very small frame rate increase from 2D to 3D (of $+0.24\%$), but to a much larger one in the expression classification rate (of $+3.95\%$). The F_1 measure is also a similar, with the frame classification rate being slightly decreased for the 3D method (-0.07%), but the expression rate noticeably increased ($+2.45\%$). Figs. 6 and 7 respectively show the full classification rates and F_1 measures achieved with both 2D and 3D methods for all four window widths tested.

IV. CONCLUSIONS

In this paper we capitalized on 3D facial motion from the BU-4DFE database in order to perform analysis of facial expression dynamics for the purpose of fully automatic expression recognition. We based the approach on 3D motion-based features, captured with FFDs, which were captured in each pair of dimensions. Best features were picked and classified by GentleBoost classifiers, and the output of these was used to build temporal models of each expression using an HMM. Three expressions were used to train and test the full system, and the results of these experiments were examined and compared with the same method performed on 2D facial motion data extracted from facial intensity image sequences from the same database (using manual image alignment). The averaged expression recognition rates indicate that there is a gain when using 3D facial motion data.

V. ACKNOWLEDGMENTS

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 through a Doctoral Training Account (DTA) Studentship. The work of Maja Pantic is also funded in part by the European Community's 7th Framework Programme [FP7/2007V2013] under the grant agreement no 231287 (SSPNet).

REFERENCES

[1] Z. Ambadar, J.W. Schooler, and J.F. Cohn. Deciphering the enigmatic face. *Psychological Science*, 16(5):403–410, 2005.

[2] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE, 2005.

[3] Y. Chang, C. Hu, and M. Turk. Probabilistic expression analysis on manifolds. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2. IEEE, 2004.

[4] I. Cohen, N. Sebe, A. Garg, L.S. Chen, and T.S. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *Computer Vision and Image Understanding*, 91(1-2):160–187, 2003.

[5] M. Costa, W. Dinsbach, A.S.R. Manstead, and P.E.R. Bitti. Social presence, embarrassment, and nonverbal behavior. *Journal of Nonverbal Behavior*, 25(4):225–240, 2001.

[6] P. Ekman and E.L. 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] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[8] S.B. Gokturk, J.Y. Bouguet, C. Tomasi, and B. Girod. Model-based face tracking for view-independent facial expression recognition. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 287–293. IEEE, 2002.

[9] L. Gralewski, N. Campbell, and I. Penton-Voak. Using a tensor framework for the analysis of facial dynamics. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 217–222. IEEE, 2006.

[10] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture based approach to recognition of facial actions and their temporal models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010. in press.

[11] I. Kotsia and I. Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing*, 16(1):172–187, 2007.

[12] G. Littlewort, M.S. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. *Image and Vision Computing*, 24(6):615–625, 2006.

[13] Dimensional Imaging Ltd. www.di3d.com.

[14] M. Pantic. Machine analysis of facial behaviour: Naturalistic and dynamic behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3505, 2009.

[15] M. Pantic and 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 CyberneticsPart B: Cybernetics*, 36(2):433, 2006.

[16] D. Rueckert, A.F. Frangi, and J.A. Schnabel. Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration. *Medical Imaging, IEEE Transactions on*, 22(8):1014–1025, 2003.

[17] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on medical imaging*, 18(8):712–721, 1999.

[18] A. Savran and B. Sankur. Automatic detection of facial actions from 3d data. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1993–2000. IEEE, 2009.

[19] N. Sebe, M.S. Lew, Y. Sun, I. Cohen, T. Gevers, and T.S. Huang. Authentic facial expression analysis. *Image and Vision Computing*, 25(12):1856–1863, 2007.

[20] H. Soyel and H. Demirel. 3d facial expression recognition with geometrically localized facial features. In *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, pages 1–4. IEEE, 2008.

[21] Y. Sun and L. Yin. Facial expression recognition based on 3D dynamic range model sequences. *Computer Vision–ECCV 2008*, pages 58–71, 2008.

[22] H. Tang and T.S. Huang. 3D facial expression recognition based on automatically selected features. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[23] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1683–1699, 2007.

[24] F. Tsalakanidou and S. Malassiotis. Real-time 2d+ 3d facial action and expression recognition. *Pattern Recognition*, 43(5):1763–1775, 2010.

[25] M.F. Valstar, H. Gunes, and M. Pantic. How to distinguish posed from spontaneous smiles using geometric features. In *Proceedings of the 9th international conference on Multimodal interfaces*, pages 38–45. ACM, 2007.

[26] M.F. Valstar and M. Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In *HCI'07: Proceedings of the 2007 IEEE international conference on Human-computer interaction*, pages 118–127, Berlin, Heidelberg, 2007. Springer-Verlag.

[27] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2 edition, 1979.

[28] J. Wang, L. Yin, X. Wei, and Y. Sun. 3D facial expression recognition based on primitive surface feature distribution. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1399–1406. IEEE, 2006.

[29] A.C.C. Williams. Facial expression of pain: An evolutionary account. *Behavioral and brain sciences*, 25(04):439–455, 2002.

[30] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3D dynamic facial expression database. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2009.

[31] L. Yin, X. Wei, P. Longo, and A. Bhuvanesh. Analyzing facial expressions using intensity-variant 3D data for human computer interaction. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 1248–1251. IEEE, 2006.

[32] S. Zafeiriou and I. Pitas. Discriminant graph structures for facial expression recognition. *Multimedia, IEEE Transactions on*, 10(8):1528–1540, 2008.

[33] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 699–714, 2005.

[34] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[35] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 915–928, 2007.