Facial Expression Recognition in Image Sequences using Geometric Deformation Features and Support Vector Machines

Irene Kotsia† and Ioannis Pitas†,Senior Member IEEE

†Aristotle University of Thessaloniki

Department of Informatics

Box 451

54124 Thessaloniki, Greece

email: {ekotsia,pitas}@aiia.csd.auth.gr

Abstract

In this paper, two novel methods for facial expression recognition in facial image sequences are presented. The user has to manually place some of Candide grid nodes to face landmarks depicted at the first frame of the image sequence under examination. The grid tracking and deformation system used, based on deformable models, tracks the grid in consecutive video frames over time, as the facial expression evolves, until the frame that corresponds to the greatest facial expression intensity. The geometrical displacement of certain selected Candide nodes, defined as the difference of the node coordinates between the first and the greatest facial expression intensity frame, is used as an input to a novel multi-class Support Vector Machine (SVM) system of classifiers, that are used to recognize either the six basic facial expressions or a set of chosen Facial Action Units (FAUs). The results on the Cohn-Kanade database show a recognition accuracy of 99.7% for facial expression recognition using the proposed multi-class SVMs and 95.1% for facial expression recognition based on FAU detection.

Index Terms

Facial expression recognition, Facial Action Unit, Facial Action Coding System, Support Vector Machines, Candide grid.

I. Introduction

During the past two decades, facial expression recognition has attracted a significant interest in the scientific community, as it plays a vital role in human centered interfaces. Many applications such as virtual reality, video-conferencing, user profiling and customer satisfaction studies for broadcast and web services, require efficient facial expression recognition in order to achieve the desired results. Therefore, the impact of facial expression recognition on the above mentioned application areas, is constantly growing.

Several research efforts have been done regarding facial expression recognition. The facial expressions under examination were defined by psychologists as a set of six basic facial expressions (anger, disgust, fear, happiness, sadness and surprise) [1]. In order to make the recognition procedure more standardized, a set of muscle movements known as $Facial\ Action\ Units\ (FAUs)$ that produce each facial expression, was created, thus forming the so called $Facial\ Action\ Coding\ System\ (FACS)$ [2]. These FAUs are combined in order to create the rules responsible for the formation of facial expressions as proposed in [3].

A. Facial expression recognition

A survey on the research made regarding facial expression recognition can be found at [4], [5]. The approaches reported regarding facial expression recognition can be distinguished in two main directions, the feature-based ones and the template based ones, according to the method they use for facial information extraction. The feature-based methods use texture or geometrical information as features for expression information extraction. The template-based methods use 3-D or 2-D head and facial models as templates for expression information extraction.

Feature based Approaches

Facial feature detection and tracking is based on active InfraRed illumination in [6], in order to provide visual information under variable lighting and head motion. The classification is performed using a Dynamic Bayesian Network (DBN).

A method for static and dynamic segmentation and classification of facial expressions is proposed in [7]. For the static case, a DBN is used, organized in a tree structure. For the dynamic approach, multi level Hidden Markov Models (HMMs) classifiers are employed.

The system proposed in [8] automatically detects frontal faces in the video stream and classifies them in seven classes in real time: neutral, anger, disgust, fear, joy, sadness and surprise. An expression recognizer receives image regions produced by a face detector and then a Gabor representation of the facial image region is formed to be later processed by a bank of SVMs classifiers.

Gabor filters are also used in [9] for facial expression recognition. Facial expression images are coded using a multi-orientation, multi-resolution set of Gabor filters which are topographically ordered and aligned approximately with the face. The similarity space derived from this facial image representation is compared with one derived from semantic ratings of the images by human observers. The classification is performed by comparing the produced similarity spaces.

The images are first transformed using a multiscale, multiorientation set of Gabor filters in [10]. The grid is then registered with the facial image region either automatically, using elastic graph matching [11] or by manual clicking on fiducial face points. The amplitude of the complex valued Gabor transform coefficients are sampled on the grid and combined into a single vector, called Labelled Graph Vector (LGV). The classification is performed using the distance of the LGV from each facial expression cluster center. Gabor features are used for facial feature extraction given a set of fiducial points in [12]. The classification is performed using Bayes, SVMs, Adaboost and Linear Programming classifiers.

A Neural Network (NN) is employed to perform facial expression recognition in [13]. The features used can be either the geometric positions of a set of fiducial points on a face or a set of multi-scale and multi-orientation Gabor wavelet coefficients extracted from the facial image at the fiducial points. The recognition is performed by a two layer perceptron NN. A Convolutional NN was used in [14]. The system developed is robust to face location changes and scale variations. Feature extraction and facial expression classification were performed using neuron groups, having as input a feature map and properly adjusting the weights of the neurons for correct classification. A method that performs facial expression recognition is presented in [15]. Face detection is performed using a Convolutional

NN, while the classification is performed using a rule-based algorithm. Optical flow is used for facial region tracking and facial feature extraction in [16]. The facial features are inserted in a Radial Basis Function (RBF) NN architecture that performs classification. Discrete Cosine Transform (DCT) is used in [17], over the entire face image as a feature detector. The classification is performed using a one-hidden layer feedforward NN.

A feature selection process that is based on Principal Component Analysis (PCA) is proposed in [18]. A decision tree-based classifier that uses successive projections onto more precise representation subspaces, is employed. The image pixels are used in [19] as input to PCA and Linear Discriminant Analysis (LDA) to reduce the original feature space dimensionality. The resulted features are lexicographically ordered and concatenated to a feature vector, which is used for classification according to the nearest neighbor rule.

The approach followed in [20] uses structured and geometrical features of a user sketched expression model. The classification is performed using Linear Edge Mapping (LEM). Expressive face modelling, using an Active Appearance Model (AAM) is employed in [21]. The facial model is constructed based on either three or one PCA. The classification is performed in the space of AAM.

Model-template based Approaches

Two methods for facial expression recognition are proposed in [22], based on a 3-D model enriched with muscles and skin. The first method estimates facial muscle actuations from optical flow data. The classification is performed according to its similarity to the classical patterns of muscle actuation. The second method uses the classical patterns of muscle actuation to generate the classical pattern of motion energy associated with each facial expression, thus resulting in a set of simple facial expression "detectors", each of which looks for the particular space-time pattern of motion energy associated with each facial expression.

A face model, defined as a point-based model composed of two 2-D facial views (frontal and profile views) is used in [3]. The deformation of facial features is extracted from both the frontal and profile views and its correspondence with the FAUs is established. The facial expression recognition is performed based on a set of decision rules.

A 3-D facial model is proposed in [23]. Anatomically-based muscles are added to it. A Kalman filter in correspondence with optical flow computation are used to extract muscle action in order to form a new model

of facial action, the so-called $FACS+$.

A 3-D facial model used for facial expression recognition is also proposed in [24]. First, the head pose is estimated in a facial video sequences. Subsequently, face images are warped onto a face model with canonical face geometry, then they are rotated to frontal ones and are projected back onto the image plane. Pixels brightness is linearly rescaled and resulting images are convolved with a bank of Gabor kernels. The Gabor representations are then channelled to a bank of SVMs to perform facial expression recognition.

B. FAU based facial expression recognition

For FAUs detection, the approaches followed were also feature based. Many techniques for FAUs recognition are proposed in [25]. PCA, Independent Component Analysis (ICA), Local Features Analysis (LFA), LDA, Gabor wavelet representations and Local Principal Components (LPC) are investigated more thoroughly.

A group of FAUs is detected in [26]. The facial feature contours are adjusted and both permanent and transient facial features changes are automatically detected and tracked in the image sequence. The facial parameters are then fed into two NN classifiers, one for the upper face and one for the lower face.

FAUs detection is also investigated in [27]. Facial expression information extraction is performed either by using optical flow, or by facial feature point tracking. The extracted information is used as an input in a HMMs system that has as an output upper face expressions at the forehead and brow regions.

HMMs are also used in [28]. Dense optical flow extraction is used to track flow across the entire face image, after the input image sequence is aligned. Facial feature tracking of a small set of pre-selected features is performed and high-gradient component detection uses a combination of horizontal, vertical, and diagonal line and edge feature detectors to detect and track changes in standard and transient facial lines and furrows. The results from the above system are fed to a HMMs system to perform facial expression recognition.

A NN is employed for FAUs detection in [29]. The geometric facial features (including mouth, eyes, brows and cheeks) are extracted using multi-state facial component models. After extraction, these features are represented parametrically. The regional facial appearance patterns are captured using a set of multi-scale and multiorientation Gabor wavelet filters at specific locations. The classification is performed using a back-propagation NN.

In the current paper, two novel fast feature-based methods are proposed, that use SVMs classifiers for recognizing dynamic facial expressions either directly or by firstly detecting the FAUs. SVMs were chosen due to their good performance in various practical pattern recognition applications [30], [31]-[33], and their solid theoretical foundations. A novel class of SVMs, which incorporates statistic information about the classes under examination, is also proposed. The classification on both cases (facial expression recognition using multi-class SVMs or based on FAU detection) is performed using only geometrical information, without taking into consideration any facial texture information.

Let us consider an image sequence containing a face, whose facial expression evolves from a neutral state (first frame) to a fully expressed state (last frame). The proposed method is based on mapping and tracking the facial model Candide onto the video frames. The proposed facial expression recognition system is semi-automatic, in the sense that the user has to manually place some of the Candide grid nodes [34] on face landmarks depicted at the first frame of the image sequence under examination. The tracking system allows the grid to follow the evolution of the facial expression over time till it reaches its highest intensity, producing at the same time the deformed Candide grid at each video frame. A subset of the Candide grid nodes is chosen, that predominantly contribute to the formation of the facial deformations described by FACS. The geometrical displacement of these nodes, defined as the difference of each node coordinates at the first and the last frame of the facial image sequence, is used as an input to a SVMs classifier (either the classical or the proposed one). When facial expression recognition using multi-class SVMs is performed, the SVMs system consists of a six-class SVMs classifier, each class representing one of the six basic facial expressions (anger, disgust, fear, happiness, sadness and surprise). When FAU based facial expression recognition is performed, 8 or 17 FAUs are chosen that corresponds to the new empirically derived facial expressions rules and to the rules proposed in [3]. Thus, the recognition system used is composed of a bank of two-class SVMs, each one detecting the presence or absence of a particular FAU that corresponds to a specific facial expression. The experiments were performed using the Cohn-Kanade database and the results show that the proposed novel facial expression recognition system can achieve a recognition accuracy of 99.7% or 95.1%, when recognizing six basic facial expressions on the Cohn-Kanade database by the multi-class SVMs approach or by the FAU detection based approach, respectively.

Summarizing, the contributions of this paper are:

- The presentation of a real-time system able to correctly classify facial expressions and FAUs, taking under consideration only geometrical displacement information based on the standard and well known Candide grid [35], in contrary to other approaches that use their own models without having explicitly defined them [7], [9], [23], [36], [37].
- The introduction of a new class for multi-class SVMs classification, based on the extension of the approach described in [30].
- The presentation of a new set of empirical rules for facial expression recognition using FAUs, as well as a simplified Candide model whose nodes correspond to the above mentioned FAUs.

Our system is different from the method proposed in [38] as it:

- uses a general and well known model (Candide facial grid) for tracking and information extraction, and not an arbitrary grid that the author chose not having been properly defined for public use
- a method for FAU recognition and facial expression recognition through the FAUs appearing in a facial grid is also presented
- a novel modified SVMs system is proposed and used to solve the facial expression recognition problem, proving at the same time that its performance greatly outperforms the maximum margin SVMs approach [39].

This paper is organized as follows: The system used for facial expression classification is described in Section II. The facial expression rules used for the synthesis of the six basic facial expressions as proposed in [3] are presented in Section II-C. The modified SVMs for a two-class and multi-class problem are presented in Sections III and IV respectively. The database used for the experiments and their description for both approaches are presented in Section V-A. The newly proposed rules for the simplified Candide grid and the facial expressions are described in Section V-B. The accuracy rates achieved when the chosen subset of FAUs was used as well as when facial expression recognition was attempted using multi-class SVMs are shown in Sections V-C and V-E respectively. Conclusions are drawn in Section VI.

II. System description

The facial expression recognition system is composed of two subsystems: one for Candide grid node information extraction and one for grid node information classification. The grid node information extraction is performed by a tracking system, while the grid node information classification is performed by a SVMs system. The flow diagram of the proposed system is shown in Figure 1.

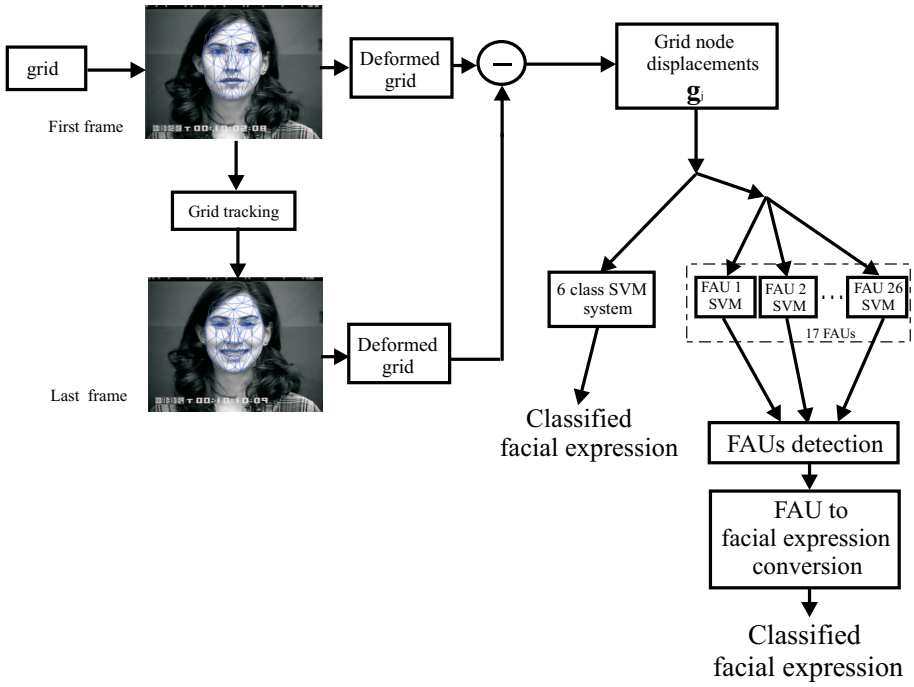

Fig. 1. System architecture for facial expression recognition in facial videos

A. Tracking system initialization

The initialization procedure is performed in a semi-automatic way in order to attain reliability and robustness of the initial grid displacement. The facial wireframe model used in the tracking procedure is the well-known Candide wireframe model [34], in the contrary to the other approaches that use their own models without having explicitly defined them. Candide is a parameterized face mask specifically developed for model-based coding of human faces. A frontal and a profile view of the model can be seen in Figure 7. The low number of its triangles allows fast face animation with moderate computing power.

In the beginning, the Candide wireframe grid is randomly placed on the facial image depicted at the first frame. The grid is in its neutral state. The user has to manually select a number of point correspondences that are matched

against the facial features of the actual face image. Future research involves the automatical placement of the grid on the face, using elastic graph matching algorithms. The most significant nodes (around the eyes, eyebrows and mouth) should be chosen, since they are responsible for the formation of facial deformations modelled by FACS. It has been empirically determined that 5 to 8 node correspondences are enough for a good model fitting. These correspondences are used as the driving power which deforms the rest of the model and matches its nodes against face image points. The result of the initialization procedure, when 7 nodes (4 for the inner and outer corner of the eyes and 3 for the upper lip) are placed by the user, can be seen in Figure 2.

Fig. 2. Result of initialization procedure when 7 Candide nodes are placed by the user on a facial image.

B. Model based tracking

Wireframe node tracking is performed by a pyramidal variant of the well-known Kanade-Lucas-Tomasi (KLT) tracker [40]. The loss of tracked features is handled through a model deformation procedure that increases the robustness of the tracking algorithm.

The algorithm, initially fits and subsequently tracks the Candide facial wireframe model in video sequences containing the formation of a dynamic human facial expression from the neutral state to the fully expressive one. The facial features are tracked in the video sequence using a variant of KLT tracker [40]. If needed, model deformations are performed by mesh fitting at the intermediate steps of the tracking algorithm. Such deformations provide robustness and tracking accuracy.

The facial model is assumed to be a deformable 2-D mesh model. The facial model elements (springs) are assumed to have a certain stiffness. The driving forces that are needed, i.e., the forces that deform the model, are determined from the point correspondences between the facial model nodes and the face image features. Each force

is defined to be proportional to the difference between the model nodes and their corresponding matched feature points on the face image. If a node correspondence is lost, the new node position is the result of the grid deformation. This solves a major problem of feature-based tracking algorithms, the gradual elimination of features points with respect to time. In the modified tracking algorithm, the incorporation of the deformation step enables the tracking of features that would have been lost otherwise. The tracking algorithm provides a $dynamic\ facial\ expression$ model for each video sequence, which is defined as a series of frame facial expression models, one for each video frame.

An example of the deformed frame facial expression models produced for each one of the 6 basic facial expressions can be seen in Figure 3.

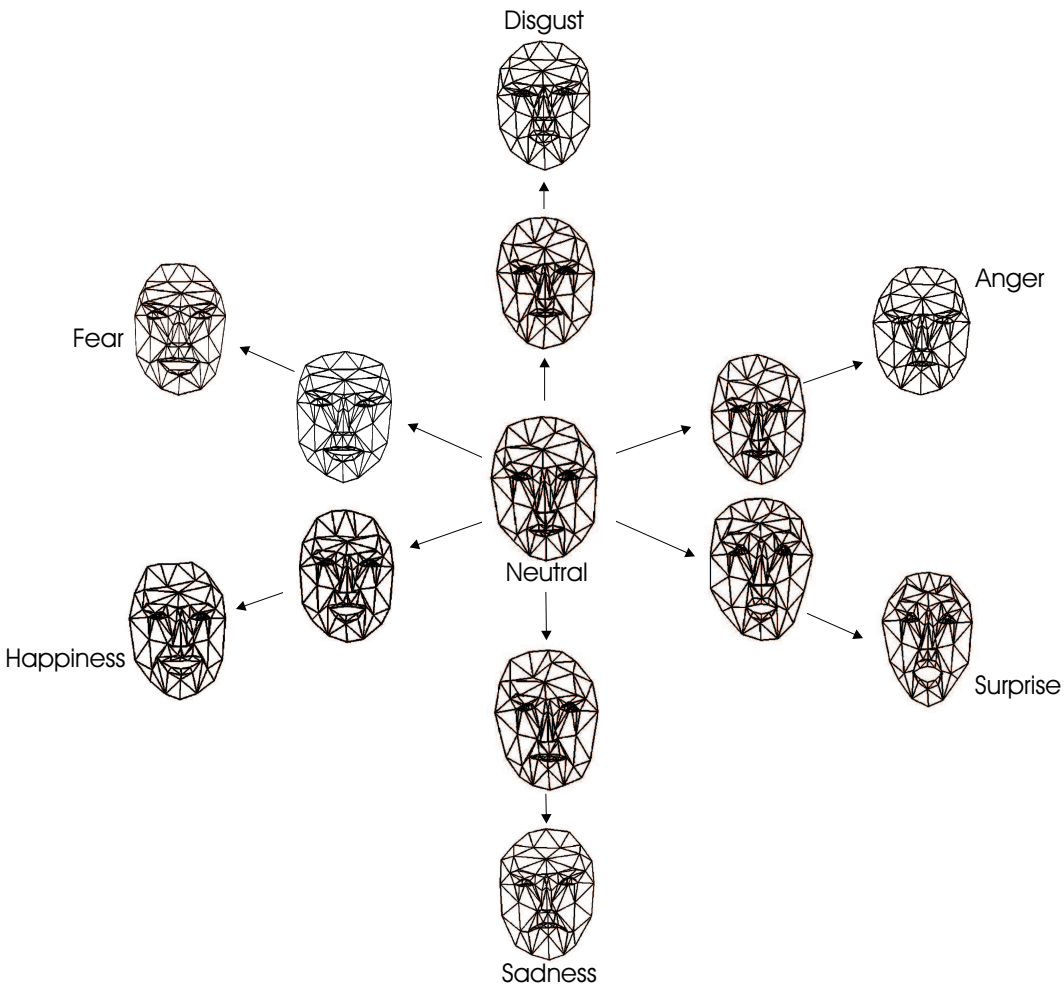

Fig. 3. An example of the deformed Candide grids for each one of the 6 facial expressions.

C. Grid node displacement extraction

In the proposed approach, the facial expression classification is performed based only on geometrical information, without taking directly into consideration any facial texture information. The geometrical displacement information of the grid node coordinates is used either for facial expression recognition using multi-class SVMs or for FAU-based facial expression recognition. In the FAU-based recognition, the activated FAUs should be detected in the grid, before employing them to produce one of the six basic facial expressions using a set of rules that maps them to facial expressions.

Let $\mathcal{U}$ be the video database that contains the facial image sequences. In the case of facial expression recognition using multi-class SVMs, it is clustered into 6 different classes $\mathcal{U}_k,\ k = 1, \ldots, 6$, each one representing one of 6 basic facial expressions (anger, disgust, fear, happiness, sadness and surprise). In the case of FAU-based facial expression recognition, for every FAU, the database is clustered into 2 different classes $\mathcal{U}_k^i,\ i = 1, 2$ for the k-th FAU, $k = \{1, \ldots, 17\}$. The first class, $\mathcal{U}_k^1$, represents the presence of the FAU under examination at the grid being processed, while the second one, $\mathcal{U}_k^2$, represents its absence.

The geometrical information used for facial expression recognition is the displacement of one node $\mathbf{d}_{i,j}$, defined as the difference of the i-th grid node coordinates at the first and the fully formed expression facial video frame:

$$\mathbf{d}_{i,j} = [\Delta x_{i,j} \quad \Delta y_{i,j}]^T \quad i = 1, \ldots, E \quad \text{and} \quad j = 1, \ldots, N \tag{1}$$

where $\Delta x_{i,j},\ \Delta y_{i,j}$ are the $x,\ y$ coordinate displacement of the i-th node in the j-th image respectively. E is the total number of nodes ($E = 104$ for the Candide model) and N is the number of the facial image sequences. This way, for every facial image sequence in the training set, a feature vector $\mathbf{g}_j$ is created, called $grid\ deformation\ feature\ vector$ containing the geometrical displacement of every grid node:

$$\mathbf{g}_j = [\mathbf{d}_{1,j} \quad \mathbf{d}_{2,j} \ \ldots \ \mathbf{d}_{E,j}]^T,\ \ j = 1, \ldots, N \tag{2}$$

having $L = 104 \cdot 2 = 208$ dimensions. We assume that each grid deformation feature vector $\mathbf{g}_j\ \ j = 1, \ldots, N$ belongs to one of the six facial expression classes $\mathcal{U}_k,\ k = 1, \ldots, 6$ (for facial expression recognition using multi-class SVMs) and activates a number of FAUs (for FAUs detection based facial expression recognition).

Facial expressions can be described as combinations of FAUs, as proposed in [3]. As can be seen in the original

TABLE I

THE FACIAL EXPRESSION SYNTHESIS RULES AS PROPOSED IN [3].

Expression	FAUs coded description [3]
Anger	$4+7+(((23 \text{ or } 24) \text{ with or not } 17) \text{or } (16+(25 \text{ or } 26)) \text{ or } (10+16+(25 \text{ or } 26)))$with or not 2
Disgust	$((10 \text{ with or not } 17) \text{ or } (9 \text{ with or not } 17))+(25 \text{ or } 26)$
Fear	$(1+4)+(5+7)+20+(25 \text{ or } 26)$
Happiness	$6+12+16+(25 \text{ or } 26)$
Sadness	$1+4+(6 \text{ or } 7)+15+17+(25 \text{ or } 26)$
Surprise	$(1+2)+(5 \text{ without } 7)+26$

rules (Table I), the FAUs that are necessary for fully describing all facial expressions are FAUs 1, 2, 4, 5, 6, 7, 9, 10, 12, 15, 16, 17, 20, 23, 24, 25 and 26. Therefore, these 17 FAUs are responsible for describing face deformations according to FACS. The operators $+$, *or* refer to the logical AND, OR operations respectively. Therefore, FAUs can be easily used for facial expression recognition. In the following, we will formulate the SVMs-based classification problems used for FAUs detection in the grid and for facial expression recognition.

III. FAU DETECTION USING SVMS

Here we shall describe two such classification algorithms. One is based on the well-known SVMs [39] and the other one is a modified version of SVMs as proposed in [30].

A. Two Class SVMs FAU detection

In our approach in order to detect the activated FAUs, the grid deformation feature vector $\mathbf{g}_j \in \Re^L \;\; j=1,\ldots,N$ is used as an input to 17 two class SVMs systems, each one detecting a specific FAU. Each SVMs system, uses the grid nodes geometrical displacements to decide whether a specific FAU is activated at the grid under examination or not. The k-th SVM $k=1,\ldots,17$ is trained with the examples in $\mathcal{U}_k^1 = \{\ (\mathbf{g}_j, y_j^k),\ \ j=1,\ldots,N,\ \ y_j^k=1\ \}$ as positive ones and all other examples $\mathcal{U}_k^2 = \{\ (\mathbf{g}_j, y_j^k),\quad j=1,\ldots,N,\ \ y_j^k=-1\ \}$ as negative ones.

In order to train the k-th SVMs network, the following minimization problem has to be solved [41]:

$$\min_{\mathbf{w}_k, b_k, \boldsymbol{\xi}^k} \qquad \frac{1}{2}\mathbf{w}_k^T\mathbf{w}_k + C_k\sum_{j=1}^{N}\xi_j^k \tag{3}$$

subject to the separability constraints:

$$y_i^k(\mathbf{w}_k^T\phi(\mathbf{g}_j)+b_k)\geq 1-\xi_j^k,\xi_j^k\geq 0,\quad j=1,\ldots,N$$

where b_k is the bias for the k-th SVM, $\boldsymbol{\xi}^k=[\xi_1^k,\ldots,\xi_N^k]$ is the slack variable vector and C_k is the term that penalizes the training errors.

After solving the optimization problem (3) subject to the separability constraints (4) ([39], [42]), the function that decides whether the k-th FAU is activated by a test displacement feature vector $\mathbf{g}$ is:

$$f_k(\mathbf{g})=\text{sign}(\mathbf{w}_k^T\phi(\mathbf{g})+b_k) \tag{4}$$

where $\mathcal{H}$ is an arbitrary dimensional Hilbert space [43] and $\phi:\Re^L\rightarrow\mathcal{H}$. In this formulation, a nonlinear mapping ϕ has been used for a high dimensional feature mapping for obtaining a linear SVMs system in which it should be $\phi(\mathbf{g})=\mathbf{g}$. This mapping is defined by a positive kernel function, $h(\mathbf{g}_i,\mathbf{g}_j)$, specifying an inner product in the feature space and satisfying the Mercer condition [39], [42]:

$$h(\mathbf{g}_i,\mathbf{g}_j)=\phi(\mathbf{g}_i)^T\phi(\mathbf{g}_j). \tag{5}$$

The functions used as SVMs kernels were the d degree polynomial function:

$$h(\mathbf{g}_i,\mathbf{g}_j)=({\mathbf{g}_i}^T\mathbf{g}_j+1)^d \tag{6}$$

and the Radial Basis Function (RBF) kernel:

$$h(\mathbf{g}_i,\mathbf{g}_j)=\exp(-\gamma\parallel\mathbf{g}_i-\mathbf{g}_j\parallel^2). \tag{7}$$

where γ is the spread of the Gaussian function.

B. A Modified Two Class SVMs

The other classifier tested for FAU detection is based on a modified two class SVMs formulation proposed in [30]. The approach in [30], was motivated by the fact that the Fisher's discriminant optimization problem for two classes is a constraint least-squares optimization problem [30], [44], [45]. The problem of minimizing the within-class variance has been reformulated so that it can be solved by constructing the optimal separating hyperplane for both separable and nonseparable cases. The modified SVMs class [30] has been applied successfully in order

to weight the elastic graph nodes local similarity value according to their corresponding discriminant power for frontal face verification. It has been shown that it outperforms the classical[1] SVMs approach. More details about the motivations of this modified SVMs can be found in [30].

1) The Linear Case: In order to form the optimization problem of the SVMs proposed in [30] we should define the within class scatter matrix of the training set:

$$\mathbf{S}_w^k = \sum_{\mathbf{g}_i \in \mathcal{U}_k^1} (\mathbf{g}_i - \boldsymbol{\mu}_k^1)(\mathbf{g}_i - \boldsymbol{\mu}_k^1)^T + \sum_{\mathbf{g}_i \in \mathcal{U}_k^2} (\mathbf{g}_i - \boldsymbol{\mu}_k^2)(\mathbf{g}_i - \boldsymbol{\mu}_k^2)^T \tag{8}$$

where $\boldsymbol{\mu}_k^1$ and $\boldsymbol{\mu}_k^2$ are the mean vectors of the classes $\mathcal{U}_k^1$ and $\mathcal{U}_k^2$, respectively. In this approach we assume that the within scatter matrix $\mathbf{S}_w^k$ is invertible (which is true in our case, since the dimensionality of the vector $\mathbf{g}_i$ is classically smaller than the number of available training examples). The optimization problem of the modified SVMs is [30]:

$$\min_{\mathbf{w}_k, b_k, \boldsymbol{\xi}^k} \qquad \mathbf{w}_k^T \mathbf{S}_w^k \mathbf{w}_k + C_k \sum_{j=1}^{N} \xi_j^k \tag{9}$$

subject to the separability constraints (4) (here we refer to the linear case where $\phi(\mathbf{g}) = \mathbf{g}$). The solution of the optimization problem (9) subject to the constraints (4) is given by the saddle point of the Lagrangian:

$$L(\mathbf{w}_k, b_k, \boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \boldsymbol{\xi}^k) = \mathbf{w}_k^T \mathbf{S}_w^k \mathbf{w}_k + C_k \sum_{i=1}^{N} \xi_i^k - \sum_{i=1}^{N} a_i^k [y_i^k (\mathbf{w}_k^T \mathbf{g}_i - b_k) - 1 + \xi_i^k] - \sum_{i=1}^{N} \beta_i^k \xi_i^k \tag{10}$$

where $\boldsymbol{\alpha}^k = [\alpha_1^k, \ldots, \alpha_N^k]$ and $\boldsymbol{\beta}^k = [\beta_1^k, \ldots, \beta_N^k]$ are the vectors of Lagrangian multipliers for the constraints (4). The vector $\mathbf{w}_k$ can be derived from the Kuhn-Tucker (KT) conditions [30]:

$$\mathbf{w}_k = \frac{1}{2} {\mathbf{S}_w^k}^{-1} \sum_{i=1}^{N} a_i^k y_i^k \mathbf{g}_i. \tag{11}$$

Instead of finding the saddle point of the Lagrangian (10), we find the maximization point of the Wolf dual problem [30]:

$$W(\boldsymbol{\alpha}^k) = \sum_{i=1}^{N} \alpha_i^k - \frac{1}{4} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^k \alpha_j^k y_i^k y_j^k \mathbf{g}_i^T {\mathbf{S}_w^k}^{-1} \mathbf{g}_j \tag{12}$$

subject to:

$$\begin{aligned} &0 \leq \alpha_i^k \leq C_k, i = 1, \ldots, N \\ &\textstyle\sum_i^N \alpha_i^k y_i^k = 0. \end{aligned} \tag{13}$$

[1]The term classical SVMs refers to the maximal margin SVMs proposed in [39]

The above optimization problem can be solved using optimization packages like [46] or using the "$quadprog$" function of MATLAB [47].

The linear decision function that decides whether the k-th FAU is activated in the geometrical displacement vector $\mathbf{g}$, or not, is:

$$f_k(\mathbf{g}) = \text{sign}(\mathbf{w}_k^T \mathbf{g} + b_k) = \text{sign}(\frac{1}{2}\sum_{j=1}^{N} y_i^k a_i^k \mathbf{g}_j^T \mathbf{S}_w^{k\,-1} \mathbf{g} + b_k). \tag{14}$$

2) The Non-Linear Case: The nonlinear multi-class decision surfaces can be created in the same manner as the two class non-linear decision surfaces that have been proposed in [30]. That is, in the dual Wolf problem the term $\mathbf{g}_i^T \mathbf{S}_w^{k\,-1} \mathbf{g}_j$ is employed. Assuming that the within scatter matrix is invertible, this term can be written as $(\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_i)^T (\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_j)$. Applying the nonlinear function ϕ to the vectors $\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_i$, we have $h(\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_i, \mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_j) = \phi(\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_i)^T \phi(\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_j)$ [30]. Then, we can apply kernel functions in (15) as:

$$W(\boldsymbol{\alpha}_k) = \sum_{i}^{N} \alpha_i^k - \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i^k \alpha_j^k y_i^k y_j^k h(\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_i, \mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_j). \tag{15}$$

The corresponding non-linear decision function that detects the k-th FAU in the geometrical displacement vector $\mathbf{g}$ is given by:

$$f_k(\mathbf{g}) = \frac{1}{2}\sum_{j=1}^{N} y_i^k \alpha_i^k h(\mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}_j, \mathbf{S}_w^{k\,-\frac{1}{2}} \mathbf{g}) + b_k. \tag{16}$$

IV. Facial Expression Recognition using multi-class SVMs

For facial expression recognition using multi-class SVMs, the grid deformation feature vector $\mathbf{g}_j \in \Re^L$ is used as an input to a multi class SVMs system [48]. Six classes were considered for the experiments, each one representing one of the basic facial expressions (anger, disgust, fear, happiness, sadness and surprise). The SVMs system, classifies the set of the grid geometrical displacements to one of the six basic facial expressions. More specifically, the grid deformation vectors $\mathbf{g}_j, \;\; j = 1, \ldots, N$, are used as an input to the SVMs system. The output of the SVMs system is a label that classifies the grid deformation under examination to one of the six basic facial expressions.

In this Section we will also show how the two class SVMs described in Section III-B.2 can be extended to multi-class classifications problems using the multi-class SVMs formulation presented in [39], [49], [50]. In the experimental results section we will show that the modified multi-class SVMs outperforms the ones proposed in [39], [49], [50].

A. Multi-class SVMs

A brief conversation about the optimization problem of the multi-class SVMs will be given below. The interested reader can refer to [39], [41], [49], [50] and the references therein for formulating and solving multi-class SVMs optimization problems.

The training data are $(\mathbf{g}_1, l_1), \ldots, (\mathbf{g}_N, l_N)$ where $\mathbf{g}_j \in \Re^L$ are the grid deformation vectors and $l_j \in \{1, \ldots, 6\}$ are the facial expression labels of the feature vector. The multi-class SVMs problem solves only one optimization problem [49]. It constructs 6 facial expressions rules, where the $k-$th function $\mathbf{w}_k^T \phi(\mathbf{g}_j) + b_k$ separates training vectors of the class k from the rest of the vectors, by minimizing the objective function:

$$\min_{\mathbf{w},\mathbf{b},\boldsymbol{\xi}} \qquad \frac{1}{2}\sum_{k=1}^{6} \mathbf{w}_k^T \mathbf{w}_k + C \sum_{j=1}^{N} \sum_{k \neq l_j} \xi_j^k \tag{17}$$

subject to the constraints:

$$\mathbf{w}_{l_j}^T \phi(\mathbf{g}_j) + b_{l_j} \geq \mathbf{w}_k^T \phi(\mathbf{g}_j) + b_k + 2 - \xi_j^k \tag{18}$$

$$\xi_j^k \geq 0, \quad j = 1, \ldots, N, \quad k \in \{1, \ldots, 6\} \backslash l_j.$$

ϕ is the function that maps the deformation vectors to a higher dimensional space, where the data are supposed to be linearly or near linearly separable. C is the term that penalizes the training errors. The vector $\mathbf{b} = [b_1 \ldots b_6]^T$ is the bias vector and $\boldsymbol{\xi} = [\xi_1^1, \ldots, \xi_i^k, \ldots, \xi_N^6]^T$ is the slack variable vector. Then, the decision function is:

$$h(\mathbf{g}) = \underset{k=1,\ldots,6}{\operatorname{argmax}} (\mathbf{w}_k^T \phi(\mathbf{g}) + b_k). \tag{19}$$

Using this procedure, a test grid deformation feature vector is classified to one of the six facial expressions using (19). Once the six-class SVMs system is trained, it can be used for testing, i.e., for recognizing facial expressions on new facial image sequences. For the solution of the optimization problem (17) subject to the constraints (18) someone can refer to [39], [49], [50].

B. A Modified Class of Multi-class SVMs

In this section, a novel multi-class SVMs method extending the constraint optimization problem in (9), is proposed. This novel multi-class SVMs method is the generalization of the two class modified SVMs described in Section III-B.

1) The Linear Case: Let that the within class scatter matrix of our grid deformation feature vectors $\mathbf{g}_i$ is defined as:

$$\mathbf{S}_w = \sum_{k=1}^{6} \sum_{\mathbf{g}_i \in \mathcal{U}_k} (\mathbf{g}_i - \boldsymbol{\mu}_k)(\mathbf{g}_i - \boldsymbol{\mu}_k)^T \tag{20}$$

where six is the number of facial expression classes and $\boldsymbol{\mu}_k$ is the geometrical displacement vector for the class k. In this Section we assume that the within class scatter matrix $\mathbf{S}_w$ is invertible, which holds in our case, since classically for our deformation the feature vector dimension is smaller than the available training examples.

The modified constraint optimization problem is:

$$\min_{\mathbf{w}_k, \mathbf{b}, \boldsymbol{\xi}} \qquad \sum_{k=1}^{6} \mathbf{w}_k^T \mathbf{S}_w \mathbf{w}_k + C \sum_{j=1}^{N} \sum_{k \neq l_j} \xi_j^k \tag{21}$$

subject to the separability constraints in (18) (in the linear case $\phi(\mathbf{g}) = \mathbf{g}$). The solution of the above constraint optimization problem can be given by finding the saddle point of the Lagrangian:

$$\begin{aligned} L(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad &= \textstyle\sum_{k=1}^{6} \mathbf{w}_k^T \mathbf{S}_w \mathbf{w}_k + C \sum_{i=1}^{N} \sum_{k=1}^{6} \xi_i^k - \sum_{i=1}^{N} \sum_{k=1}^{6} \alpha_i^k [(\mathbf{w}_{l_i} - \mathbf{w}_k)^T \mathbf{g}_i + b_{l_i} - b_k - 2 + \xi_i^k] \\ &\quad - \textstyle\sum_{i=1}^{N} \sum_{k=1}^{6} \beta_i^k \xi_i^k \end{aligned} \tag{22}$$

where $\boldsymbol{\alpha} = [\alpha_1^1, \ldots, \alpha_i^k, \ldots, \alpha_N^6]$ and $\boldsymbol{\beta} = [\beta_1^1, \ldots, \beta_i^k, \ldots, \beta_N^6]$ are the Lagrangian multipliers for the constraints (18) with :

$$\alpha_i^{l_i} = 0, \ \xi_i^{l_i} = 2, \ \beta_i^{l_i} = 0, \ i = 1, \ldots, N \tag{23}$$

and constraints:

$$\alpha_i^k \geq 0, \ \beta_i^k \geq 0, \ i = 1, \ldots, l, \ k \in \{1, \ldots, 6\} \backslash l_i. \tag{24}$$

The Lagrangian (22) has to be maximized with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and minimized with respect to $\mathbf{w}$ and $\boldsymbol{\xi}$. In order to produce a more compact equation form let us define the following variables:

$$A_i = \sum_{k=1}^{6} \alpha_i^k \ \text{ and } \ c_i^k = \begin{cases} 1, & \text{if } l_i = k \\ 0, & \text{if } l_i \neq k. \end{cases} \tag{25}$$

After a series of manipulations shown in Appendix I, the search of the saddle point of the Lagrangian (22) is reformulated to the maximization of the Wolf dual problem:

$$W(\boldsymbol{\alpha}) = 2 \sum_{i=1}^{N} \sum_{k=1}^{6} \alpha_i^k + \frac{1}{4} \sum_{i,j,k} (-\frac{1}{2} c_j^{l_j} A_i A_j + \alpha_i^k \alpha_i^{l_i} - \frac{1}{2} \alpha_i^k \alpha_j^k) \mathbf{g}_i \mathbf{S}_w^{-1} \mathbf{g}_j \tag{26}$$

which is a quadratic function in terms of $\boldsymbol{\alpha}$ with the linear constraints:

$$\sum_{i=1}^{N} a_i^k = \sum_{i=1}^{N} c_i^k A_i,\ k = 1, \ldots, 6. \tag{27}$$

The above optimization problem can be solved using optimization packages like [49]. The corresponding decision hyperplane is:

$$f(\mathbf{g}) = \underset{k=1,\ldots,6}{\operatorname{argmax}}(\mathbf{w}_k^T \mathbf{g} + b_k) = \underset{k=1,\ldots,6}{\operatorname{argmax}}[\frac{1}{2}\sum_{i=1}^{N}(c_i^k A_i - \alpha_i^k)\mathbf{g}_i^T \mathbf{S}_w^{-1}\mathbf{g} + b_k]. \tag{28}$$

as is detailed in Appendix I.

2) The Non-Linear Case: The nonlinear multi-class decision surfaces can be created in the same manner as the two class non-linear decision surfaces that have been proposed in [30] and are described in Section III-B.2. That is, we exploit the fact that the term $\mathbf{g}_i^T \mathbf{S}_w^{-1} \mathbf{g}_j$ can be written in terms of dot products as $(\mathbf{S}_w^{-\frac{1}{2}} \mathbf{g}_i)^T (\mathbf{S}_w^{-\frac{1}{2}} \mathbf{g}_j)$. Then, we can apply kernels in (26) as:

$$W(\boldsymbol{\alpha}) = 2\sum_{i=1}^{N}\sum_{k=1}^{6}\alpha_i^k + \frac{1}{4}\sum_{i,j,k}(-\frac{1}{2}c_j^{l_j} A_i A_j + \alpha_i^k \alpha_i^{l_i} - \frac{1}{2}\alpha_i^k \alpha_j^k) h(\mathbf{S}_w^{-\frac{1}{2}} \mathbf{g}_i, \mathbf{S}_w^{-\frac{1}{2}} \mathbf{g}_j), \tag{29}$$

and the corresponding decision surface is:

$$f(\mathbf{g}) = \underset{k=1,\ldots,6}{\operatorname{argmax}}\ \frac{1}{2}[\sum_{i=1}^{N}(c_i^k A_i - \alpha_i^n) h(\mathbf{S}_w^{-\frac{1}{2}} \mathbf{g}_i, \mathbf{S}_w^{-\frac{1}{2}} \mathbf{g}) + b_k]. \tag{30}$$

V. EXPERIMENTAL RESULTS

A. Database description

The Cohn-Kanade database [2] was used for the facial expression recognition in 6 basic facial expressions classes (anger, disgust, fear, happiness, sadness and surprise). This database is annotated with FAUs. These combinations of FAUs were translated into facial expressions according to [3], in order to define the corresponding ground truth for the facial expressions. All the subjects were taken under consideration to form the database for the experiments.

In Figure 4, a sample of an image for every facial expression for one poser from this database, is shown.

The most usual approach for testing the generalization performance of a SVMs classifier, is the leave-one cross-validation approach. The leave-one out method [7] was used in order to make maximal use of the available data and produce averaged classification accuracy results. The term leave-one out cross-validation, does not correspond to the classic leave-one-out definition here, as a variant of leave-one-out is used (i.e., leave 20% out) for the formation

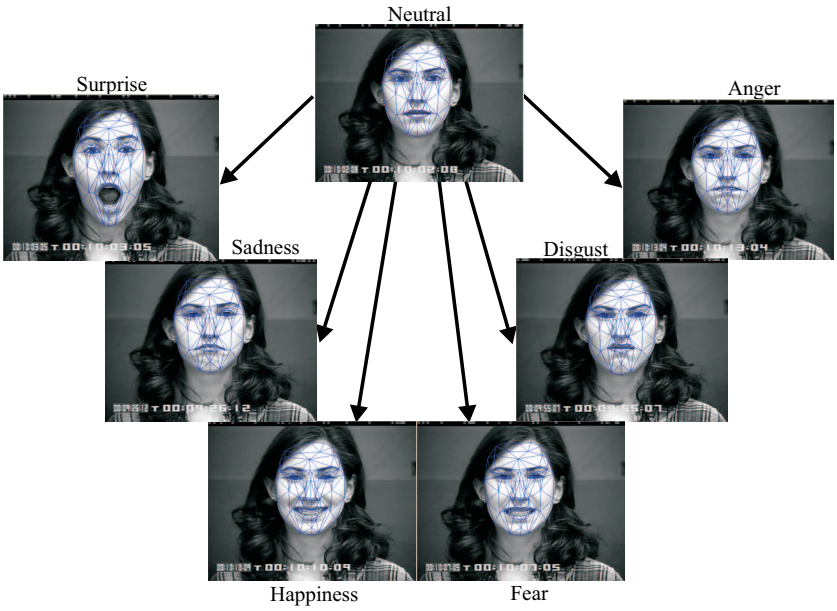

Fig. 4. An example of each facial expression for a poser from the Cohn-Kanade database.

of the test dataset. However the procedure followed will be called leave-one-out from now on. More specifically, all image sequences contained in the database are divided into 6 classes, each one corresponding to one of the 6 basic facial expressions to be recognized. Neutral state is not considered as a class, as the system tries to recognize the fully expressed facial expression starting from the neutral state. Five sets containing 20% of the data for each class, chosen randomly, were created. One set containing 20% of the samples for each class is used for the test set, while the remaining sets form the training set. After the classification procedure is performed, the samples forming the testing set are incorporated into the current training set, and a new set of samples (20% of the samples for each class) is extracted to form the new test set. The remaining samples create the new training set. This procedure is repeated five times. A diagram of the leave-one-out cross-validation method can be seen in Figure 5. The average classification accuracy is the mean value of the percentages of the correctly classified facial expressions.

The accuracy achieved for each facial expression is averaged over all facial expressions and does not provide any information with respect to a particular expression. The confusion matrices[10] have been computed to handle this problem. The confusion matrix is a $n \times n$ matrix containing the information about the actual class label lab_{ac} (in its columns) and the label obtained through classification lab_{cl} (in its rows). The diagonal entries of the confusion matrix are the rates of facial expressions that are correctly classified, while the off-diagonal entries are

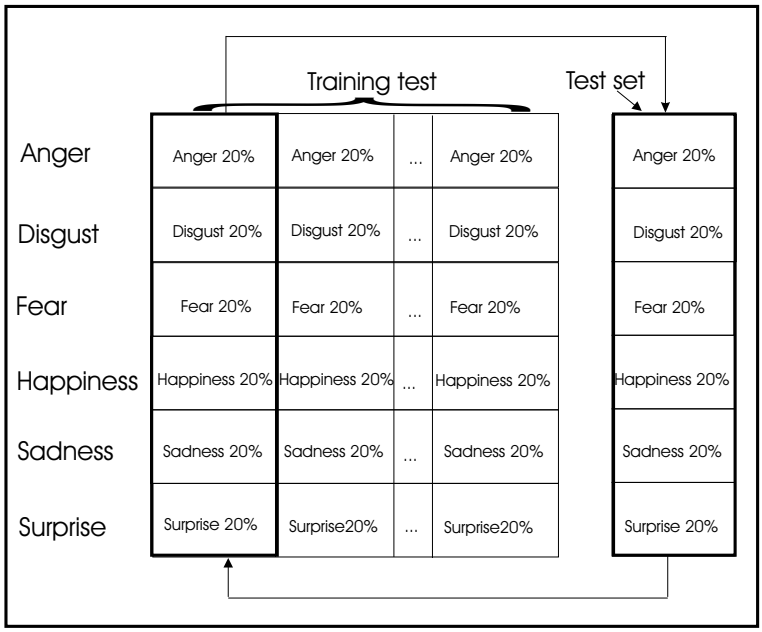

Fig. 5. Diagram of leave-one-out method.

the percentages corresponding to misclassification rates. The abbreviations an, di, fe, ha, sa and su represent anger, disgust, fear, happiness, sadness and surprise respectively.

B. Representative FAU and grid node selection

The rules proposed in [3] require the detection of 17 FAUs. The use of so many FAUs makes the rules sensitive to false FAU detection or rejection. In order to simplify the rules, a small set of rules are proposed for facial expression classification that yield better performance in the experiments performed.

From all the FAUs appearing in the facial expression description rules, many describe two or more facial expressions. Those that appear once in all facial expression rules are chosen to describe uniquely the facial expressions under examination. For example, FAU 26 appears in every facial expression. Thus, its presence is irrelevant when defining a facial expression, as no facial expression could be specified. Therefore, a FAU that exists in only one facial expression rule should be specified for each facial expression. Where it is not possible, a unique combination of FAUs should be defined instead. Therefore:

- For facial expression anger, the FAUs that appear once are the FAUs 23 and 24. The rest FAUs that participate in the facial expression rule are observed 2 or more times in all facial expression rules. FAUs 23 and 24 do not

participate in the rest of facial expressions rules for the other 5 facial expressions. Anger should be therefore defined by the appearance of those two FAUs.

- For disgust facial expression, the FAU that appears only once is the FAU 9. It is also uniquely observed in disgust rules, thus making it appropriate for disgust classification.
- For fear facial expression, the FAU that appears only once is the FAU 20. Since it appears only in fear facial expression rule, it will be the only one taken under consideration when recognizing fear.
- For happiness facial expression, the FAU that appears only once is the FAU 12. However, in Figure 6 it can be seen that FAUs 12 and 16 appear the same. Therefore, facial expression happiness will be recognized if FAUs 12 and 16 exist (both of them).
- For sadness facial expression, the FAU that appears only once is the FAU 15. Since it appears only in sadness facial expression rule, it will be the only one taken under consideration when recognizing sadness.
- For surprise facial expression, the FAUs that appear only once are FAUs 2, 5. Since they appear only in surprise facial expression rule, they will be the ones taken under consideration when recognizing surprise.

The deformed Candide grid produced by the grid tracking algorithm [51] that corresponds to the greatest intensity of the facial expression shown, contains 104 nodes. Only some of these nodes are important for facial expression recognition. For example, nodes on the outer face contour do not contribute much to facial expression recognition. Thus, a subset of 62 nodes is chosen that controls the facial deformations. The grid that is composed of these nodes can be seen in Figure 7. From this time onwards, this grid will be called $Primary\ Facial\ Expression\ Grid\ (PFEG)$. Figure 6 presents the 8 FAUs chosen as the most representative for each facial expression. The FAU definition image, as provided by Ekman and Friesen [52], is depicted, as well as its application to a poser from the Cohn-Kanade database used in our experiments.

C. FAUs detection

In this Section, only FAUs detection is described. The method followed was the application of either the classical two class SVMs (described in Section III-A) or the modified two class SVMs (described in Section III-B). The accuracy rates obtained for FAUs detection using RBF and polynomial functions as kernels, for both the classical two class SVMs as well as the modified two class SVMs and the original set of FAUs (FAUs 1, 2, 4, 5, 6, 7, 9, 10,

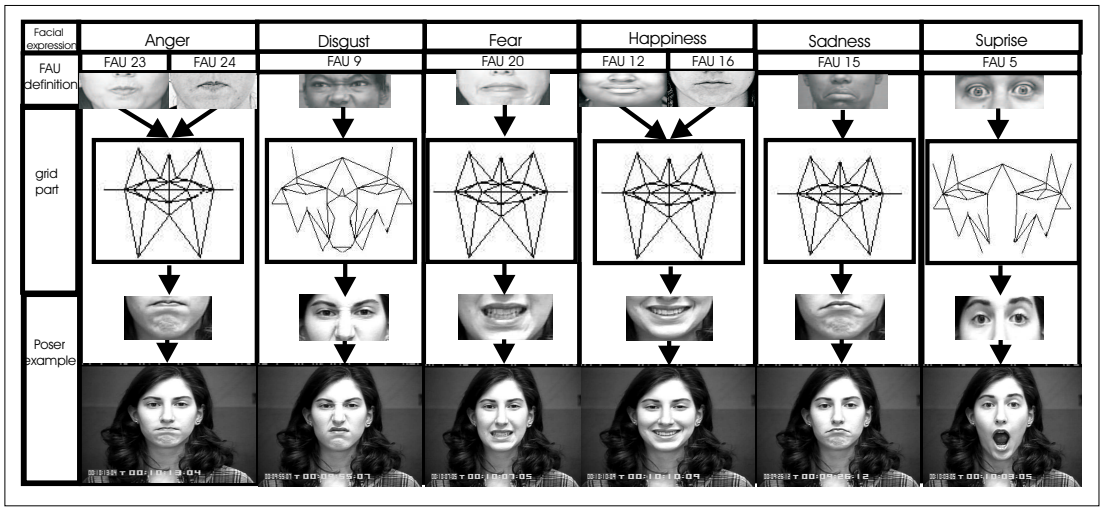

Fig. 6. The 8 most representative FAUs used for facial expression recognition.

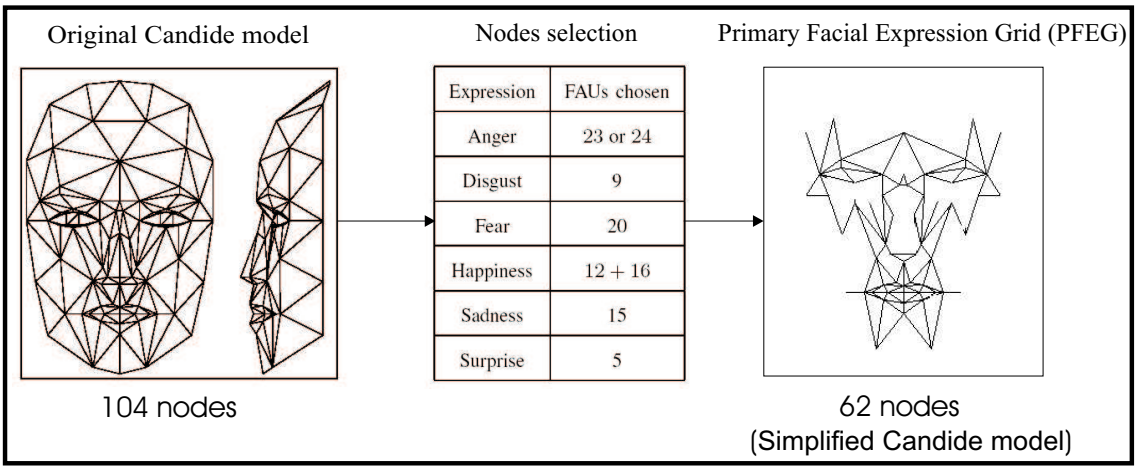

Fig. 7. The Primary Facial Expression Grid (PFEG) according to FACS, used for the experiments.

12, 15, 16, 17, 20, 23, 24, 25 and 26, 17 FAUs in total) proposed in [3], are presented in Figure 8. The equivalent FAUs detection accuracies obtained when using our proposed set of rules (corresponding to FAUs 5, 9, 12, 15, 16, 20, 23 and 24, 8 FAUs in total), are presented in Figure 9.

1) FAUs detection using Candide grid: The FAUs detection accuracy was measured as the percentage of the correctly recognized FAUs. The ground truth for FAUs was provided by the Cohn-Kanade database annotation. The achieved FAUs detection accuracy when the set of 17 FAUs was under examination and the modified two class SVMs and Candide grid were used, was equal to 82.7%. The equivalent FAUs detection accuracy rate, when our set of FAUs (subset of the 17 set of FAUs) was taken under consideration was equal to 93.5%.

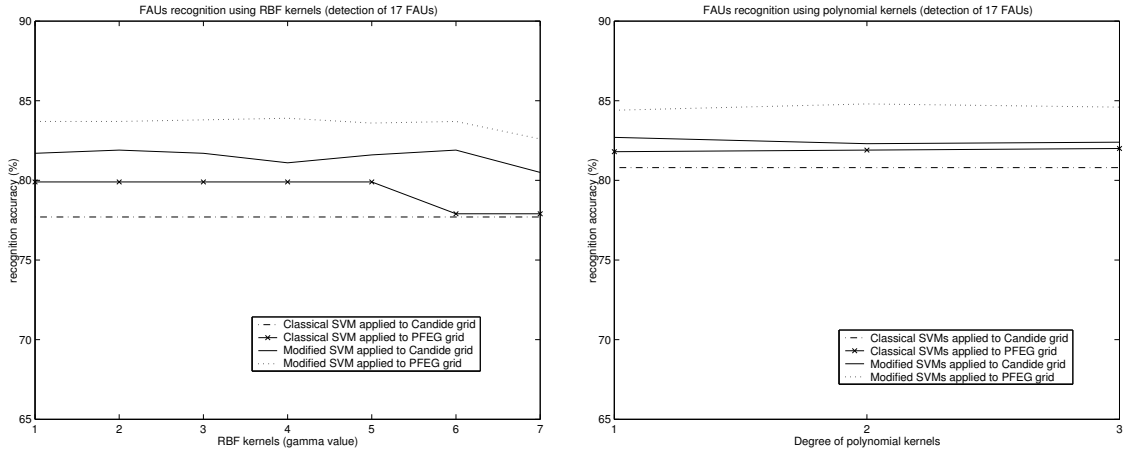

Fig. 8. Accuracy rates obtained for the set of 17 FAUs detection.

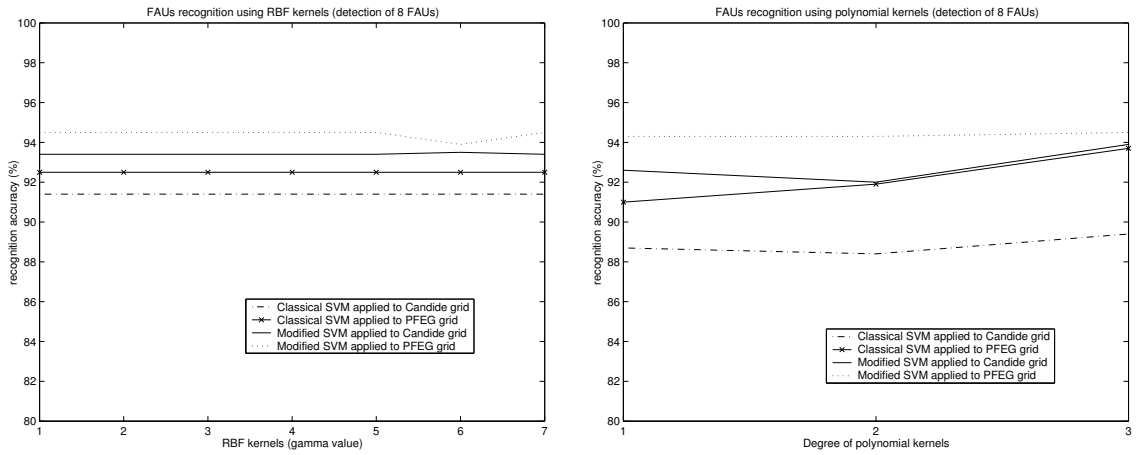

Fig. 9. Accuracy rates obtained for the set of 8 FAUs detection.

2) FAUs detection using PFEG grid: The achieved FAUs detection accuracy when the original set of FAUs was under examination and the modified two class SVMs and PFEG grid were used, was equal to 84.7%. The equivalent FAUs detection accuracy when our set of FAUs was taken under consideration was equal to 94.5%. The detection accuracy achieved by the proposed method for FAUs detection is quite satisfactory, when compared with the state of the art facial expression recognition performance for the Cohn-Kanade database [26]. More specifically, in [26] the recognition accuracy achieved was equal to 95,6%, when using the Cohn-Kanade database. The detected FAUs can be separated in 2 groups, those of upper and those of lower face. The accuracies achieved were 95.4% and 95.6% respectively although the classification method followed was not the leave-one-out procedure, as used in this paper.

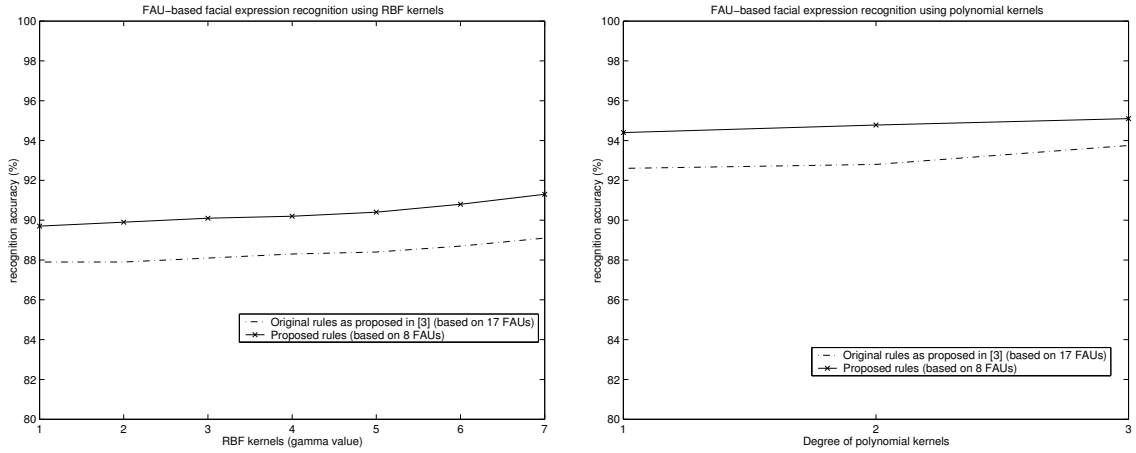

Fig. 10. Accuracy rates obtained for facial expression recognition from the detected FAUs using the PFEG grid.

D. Facial expression recognition using the detected FAUs

In this Section, facial expression recognition from the already detected FAUs is performed, either using the original set of rules as proposed in [3], or the newly proposed one. When the original set of FAUs (17 FAUs in total) was used to describe the six basic facial expressions, and the Candide grid, taking under consideration 104 nodes was applied, the facial expression recognition accuracy achieved was equal to 87.9%. When the chosen FAUs subset (FAUs 5, 9, 12, 15, 16, 20, 23 and 24, 8 FAUs in total) was used to describe the six basic facial expressions, and the Candide grid was applied, the equivalent recognition accuracy achieved was equal to 92.5%.

Regarding the application of PFEG grid (taking under consideration 62 grid nodes), the recognition accuracy obtained for the six basic facial expressions when the original set of 17 FAUs was used, was equal to 93.75%. The equivalent recognition accuracy achieved when the PFEG grid and the proposed set of 8 FAUs were used, was equal to 95.1%. Thus, when the FAUs annotation is available, the equivalent depicted facial expression can be recognized with an recognition accuracy of 95.1%, if only the FAUs 5, 9, 12, 15, 16, 20, 23 and 24, are taken into consideration.

The accuracy rates obtained for facial expression recognition from the detected FAUs using the proposed set of rules (8 FAUs) and applying the PFEG grid are presented in Figure 10. The first confusion matrix shown in Table III, presents the results obtained while using the Candide grid and the new set of rules proposed. As can be seen, the most ambiguous facial expression was disgust, since it was misclassified the most times (as anger and then sadness). The facial expressions that follow, are anger and sadness, with a similar misclassification rate. The second

TABLE II

CONFUSION MATRICES FOR FAU DETECTION BASED FACIAL EXPRESSION RECOGNITION WHEN USING THE CANDIDE AND PFEG GRID AND THE PROPOSED SET OF FAU RULES

$lab_{cl} \backslash lab_{ac}$	an	di	fe	ha	sa	su
an	80%	9.6%	0%	0.3%	5.5%	2.2%
di	6.7%	77%	0.3%	0.8%	1.5%	1.5%
fe	0%	3%	98.8%	3.2%	0%	1.9%
ha	0%	0%	0.9%	95.4%	0%	0%
sa	13.3%	8.9%	0%	0.3%	93%	0%
su	0%	1.5%	0%	0%	0%	94.4%

$lab_{cl} \backslash lab_{ac}$	an	di	fe	ha	sa	su
an	91.3%	9.5%	0%	0.3%	4.4%	1.9%
di	2.7%	80%	0.3%	0.5%	0.7%	1.1%
fe	0%	3%	99.1%	0.8%	0%	1.5%
ha	0%	0%	0.6%	98.1%	0%	0%
sa	6%	6%	0%	0.3%	94.9%	0%
su	0%	1.5%	0%	0%	0%	95.5%

confusion matrix shown in Table II, presents the results obtained while using the PFEG grid and the new set of rules proposed. As can be seen, the most ambiguous facial expression remained disgust, since it was misclassified most times, followed by anger and then sadness.

$lab_{cl} \backslash lab_{ac}$	an	di	fe	ha	sa	su
an	80%	9.6%	0%	0.3%	5.6%	2.2%
di	6.7%	77%	0.3%	0.8%	1.5%	1.5%
fe	0%	3%	98.9%	3.2%	0%	1.9%
ha	0%	0%	0.9%	95.5%	0%	0%
sa	13.3%	8.9%	0%	0.3%	93%	0%
su	0%	1.5%	0%	0%	0%	94.4%

E. Facial expression recognition using multi-class SVMs

In this Section, facial expression recognition directly from the grid nodes displacements is described. The method followed was the application of either the classical six class SVMs (described in Section IV-A) or the modified six class SVMs (described in Section IV-B).

1) Facial expression recognition using Candide grid: When the classical six class SVMs were applied to the classic Candide grid, taking under consideration 104 nodes, the facial expression recognition accuracy achieved

TABLE III

CONFUSION MATRICES FOR FACIAL EXPRESSION RECOGNITION USING MULTI-CLASS SVMS AND THE MODIFIED SVMS TO THE CANDIDE AND PFEG GRID

$lab_{ac} \backslash lab_{cl}$	an	di	fe	ha	sa	su
an	81.3%	0%	0%	0%	0%	0%
di	4.7%	100%	0%	0%	0%	0%
fe	0%	0%	100%	0%	0%	0%
ha	0%	0%	0%	100%	0%	0%
sa	14%	0%	0%	0%	100%	0%
su	0%	0%	0%	0%	0%	100%

$lab_{ac} \backslash lab_{cl}$	an	di	fe	ha	sa	su
an	96.7%	0%	0%	0%	0%	0%
di	0%	100%	0%	0%	0%	0%
fe	0%	0%	100%	0%	0%	0%
ha	0%	0%	0%	100%	0%	0%
sa	3.3%	0%	0%	0%	100%	0%
su	0%	0%	0%	0%	0%	100%

was equal to 91.4%. The equivalent facial expression recognition accuracy, when the modified six class SVMs were used, was equal to 98.2%. Therefore, the introduction of the modified six class SVMs increases the recognition accuracy by 6.8%. The first confusion matrix shown in Table III, presents the results obtained while applying the modified six class SVMs to the Candide grid. As can be seen, the most ambiguous facial expression was anger, being misclassified as sadness or disgust.

2) Facial expression recognition using PFEG grid: When the classical six class SVMs were applied to the PFEG grid, taking under consideration 62 nodes, the facial expression recognition accuracy achieved was equal to 95.75%. The equivalent facial expression recognition accuracy when the modified six class SVMs were used, was equal to 99.7%. Therefore, the introduction of the modified six class SVMs increases the recognition accuracy by 3.95%. The second confusion matrix shown in Table III, presents the results obtained while applying the modified six class SVMs to the PFEG grid. As can be seen, the most ambiguous facial expression remains anger, since it was the only one being misclassified as another facial expression (sadness). The recognition accuracies obtained for facial expression recognition using six class SVMs, RBF and polynomial functions as kernels for both the classical as well as the modified six class SVMs, are presented in Figure 11. Polynomial kernels offer better recognition performance.

The recognition accuracy achieved by the proposed method for facial expression recognition is better than any

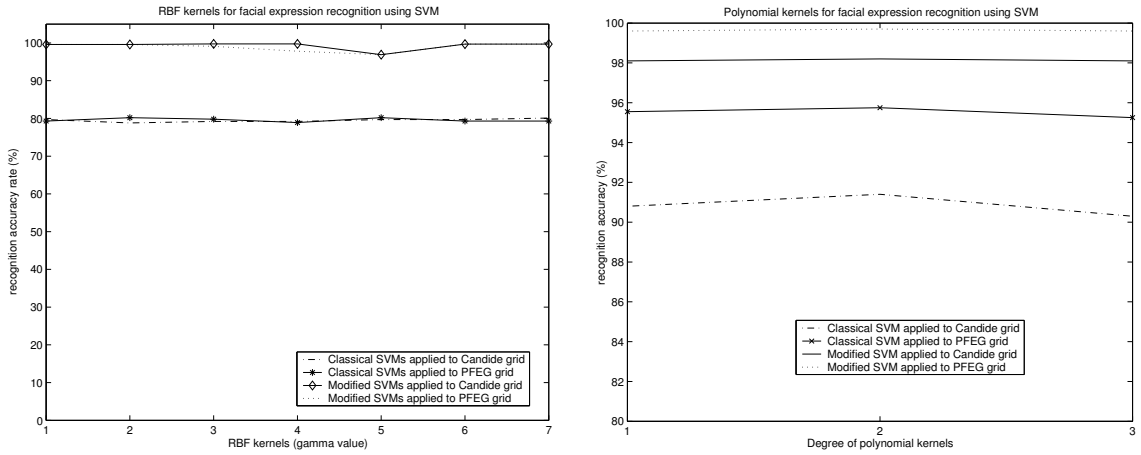

Fig. 11. Accuracy rates obtained for facial expression recognition using multi-class SVMs.

other reported in the literature so far for the Cohn-Kanade database, at least according to the authors knowledge. More specifically, in [7] the recognition accuracy achieved was equal to 74.5% for the Cohn-Kanade database and leave-one-out approach, while in [8], it was 90.7%. Generally speaking, the best facial expression recognition accuracy reported so far, is equal to 96,1% [53].

In order to understand if the proposed modified class of SVMs approach is statistically significant better than the classical SVMs approach, the McNemar's test [54] has been used. McNemar's test is a null hypothesis statistical test based on a Bernoulli model. If the resulting p-value is below a desired significance level (for example, 0.02), the null hypothesis is rejected and the performance difference between two algorithms is considered to be statistically significant. Using this test it has been verified that the modified class of SVMs outperforms the other tested classifiers in the demonstrated experiments at a significant level less that $p = 10^{-5}$.

The experiments indicated that for both approaches, the whole system is fast enough to perform almost real-time facial expression recognition, on a PC having an Intel Centrino (1,5 GHz) processor with 1GB RAM memory, since it is able to classify expressions at a rate of 20 frames per second during testing.

VI. Conclusions

Two novel methods for facial expression recognition using SVMs for facial expression recognition are proposed in this paper. A novel class for SVMs classifiers that incorporates statistical information of the classes under examination, is also proposed. The user initializes some of the Candide grid nodes on the facial image depicted at the first frame of the image sequence. The tracking system used, based on deformable models, tracks the facial

expression as it evolves over time, by deforming the Candide grid, eventually producing the grid that corresponds to the facial expression's greatest intensity, classically depicted at the last facial video frame. Only Candide nodes that influence the formation of FAUs are used in our system.Their geometrical displacement, defined as their coordinate difference between the last and the first frame, is used as an input to the SVMs system (either the classical one or the modified one). In the case of facial expression recognition, this system is composed of one six-class SVMs, one for each one of the 6 basic facial expressions (anger, disgust, fear, happiness, sadness and surprise) to be recognized. When FAUs detection based facial expression recognition is attempted, the SVMs system consists of 8 two-class SVMs, one for each one of the 8 chosen FAUs used. The proposed methods, achieve a facial expression recognition accuracy of 99.7% and 95.1% respectively. The achieved accuracy for facial expression recognition using multi-class SVMs (99.7%) is better than any other reported in the literature so far for the Cohn-Kanade database, at least according to the authors knowledge.

VII. Acknowledgments

This work was supported by the research project 01ED312 "Use of Virtual Reality for training pupils to deal with earthquakes" financed by the Greek Secretariat of Research and Technology and the "SIMILAR" European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union (www.similar.cc).

Appendix I

Wolf Dual problem for the Modified Multi-class SVMs

In order to find the optimum separating hyperplanes for the optimization problem (21) subject to the constraints (18), we have to define the saddle point of the Langragian (22).

In the saddle point, the solution should satisfy the K-T conditions, for $k = 1, \ldots, 6$:

$$\frac{\partial L(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{w}_k} = 0 \Longrightarrow \mathbf{w}_k = \frac{1}{2}\mathbf{S}_w^{-1} \sum_{i=1}^{N} (c_i^k A_i - a_i^k)\mathbf{g}_i \tag{31}$$

$$\frac{\partial L(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b_k} = 0 \Longrightarrow \sum_{i=1}^{N} \alpha_i^k = \sum_{i=1}^{N} c_i^k A_i \tag{32}$$

$$\frac{\partial L(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_k} = 0 \Longrightarrow \beta_j^k + \alpha_j^k = C \quad \text{and} \quad 0 \leq \alpha_j^k \leq C \tag{33}$$

Substituting (31) back into (22) we obtain:

$$\begin{aligned} L(\mathbf{w},\mathbf{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) \quad &= \textstyle\sum_{k=1}^{6}\sum_{i=1}^{N}\sum_{j=1}^{N}(c_i^k A_i - \alpha_i^k) \\ &(c_j^k A_j - \alpha_j^k)(\mathbf{g}_i^T \mathbf{S}_w^{-1}\mathbf{g}_j) - \\ &- \textstyle\sum_{k=1}^{6}\sum_{i=1}^{N}\alpha_i^k[\sum_{j=1}^{N}(c_j^{l_i}A_j - \alpha_j^{l_i}) \\ &(\mathbf{g}_i^T \mathbf{S}_w^{-1}\mathbf{g}_j) - \textstyle\sum_{j=1}^{N}(c_j^k A_j - \alpha_j^k) \\ &(\mathbf{g}_i^T \mathbf{S}_w^{-1}\mathbf{g}_j) + b_{l_i} - b_k - 2] - \\ &- \textstyle\sum_{k=1}^{6}\sum_{i=1}^{N}\alpha_i^k\xi_i^k + \\ &+ C\textstyle\sum_{k=1}^{6}\sum_{i=1}^{N}\xi_i^k - \sum_{i=1}^{N}\sum_{k=1}^{6}\beta_i^k\xi_i^m. \end{aligned} \tag{34}$$

Adding the constraint (33) the terms in $\boldsymbol{\xi}$ disappear. Considering the two terms in $\boldsymbol{\beta}$ only:

$$\begin{aligned} B_1 &= \textstyle\sum_{i,k}\alpha_i^k b_{l_i} = \sum_k b_k(\sum_i c_i^k A_i) \text{ and} \\ B_2 &= -\textstyle\sum_{i,k}\alpha_i^k b_k = -\sum_k b_k(\sum_i \alpha_i^k). \end{aligned} \tag{35}$$

But, from (32) we have

$$\sum_{i=1}^{N}\alpha_i^k = \sum_{i=1}^{N} c_i^k A_i \tag{36}$$

so $B_1 = B_2$ and the two terms cancel, giving:

$$\begin{aligned} L(\mathbf{w},\mathbf{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) \quad &= W(\boldsymbol{\alpha}) = 2\textstyle\sum_{i,k}\alpha_i^k + \\ &+ \tfrac{1}{4}\textstyle\sum_{i,j,k}(\tfrac{1}{2}c_i^k c_j^k A_i A_j - \tfrac{1}{2}c_i^k A_i \alpha_j^k - \\ &- \tfrac{1}{2}c_j^k A_i \alpha_i^k + \tfrac{1}{2}\alpha_i^k \alpha_j^k - c_j^{l_i} A_j \alpha_i^k \\ &+ \alpha_i^k \alpha_j^{l_i} + c_j^k A_j \alpha_i^k - \alpha_i^k \alpha_j^k)(\mathbf{g}_i^T \mathbf{S}_w^{-1}\mathbf{g}_j) \end{aligned} \tag{37}$$

Since $\sum_k c_i^k A_i \alpha_j^k = \sum_k c_j^k A_j \alpha_i^k$ we have:

$$\begin{aligned} W(\boldsymbol{\alpha}) = \quad & 2\textstyle\sum_{i,k}\alpha_i^k + \tfrac{1}{4}\sum_{i,j,k}(\tfrac{1}{2}c_i^k c_j^k A_i A_j - c_j^{l_i} A_i A_j + \\ & + \alpha_i^k \alpha_j^{l_i} - \tfrac{1}{2}\alpha_i^k \alpha_j^k)(\mathbf{g}_i^T \mathbf{S}_w^{-1}\mathbf{g}_j) \end{aligned}$$

but $\sum_k c_i^k c_j^k = c_i^{l_i} = c_j^{l_j}$ so:

$$\begin{aligned} W(\boldsymbol{\alpha}) = \quad & 2\textstyle\sum_{i,k}\alpha_i^k + \tfrac{1}{4}\sum_{i,j,k}[-\tfrac{1}{2}c_j^{y_i} A_i A_j + \alpha_i^k \alpha_j^{y_i} - \\ & - \tfrac{1}{2}\alpha_i^k \alpha_j^k](\mathbf{g}_i^T \mathbf{S}_w^{-1}\mathbf{g}_j) \end{aligned} \tag{39}$$

which is a quadratic function in terms of alpha with linear constraints:

$$\sum_{i=1}^{N}\alpha_i^k = \sum_{i=1}^{N} c_i^k A_i, \quad k = 1,\ldots,6 \tag{40}$$

and

$$0 \leq \alpha_i^k \leq C, \quad a_i^{l_i} = 0 \tag{41}$$

$$i = 1, \ldots, N \quad k \in \{1, \ldots, 6\} \backslash l_i.$$

This gives the decision function:

$$f(\mathbf{g}, \mathrm{x}) = \underset{k=1,\ldots,6}{\operatorname{argmax}} [\sum_{i=1}^{N} \frac{1}{2} (c_i^k A_i - \alpha_i^k)(\mathbf{g}_i^T \mathbf{S}_w^{-1} \mathbf{g}) + b_k] \tag{42}$$

or equivalently:

$$f(\mathbf{g}, \mathrm{x}) = \quad \operatorname{argmax}_{k=1,\ldots,6} [\tfrac{1}{2} \textstyle\sum_{i:y_i=k} A_i \mathbf{g}_i^T \mathbf{S}_w^{-1} \mathbf{g} -$$

$$-\tfrac{1}{2} \textstyle\sum_{i:y_i \neq k} \alpha_i^k \mathbf{g}_i^T \mathbf{S}_w^{-1} \mathbf{g} + b_k].$$

REFERENCES

[1] P. Ekman and W. V. Friesen, *Emotion in the Human Face*. New Jersey: Prentice Hall, 1975.

[2] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proceedings of IEEE International Conference on Face and Gesture Recognition*, March 2000, pp. 46–53.

[3] M. Pantic and L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions," *Image and Vision Computing*, vol. 18, no. 11, pp. 881–905, August 2000.

[4] ——, "Automatic analysis of facial expressions: The state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, December 2000.

[5] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[6] 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*, vol. 27, no. 5, pp. 699–714, May 2005.

[7] I. Cohen, N. Sebe, S. Garg, L. S. Chen, and T. S. Huanga, "Facial expression recognition from video sequences: temporal and static modelling," *Computer Vision and Image Understanding*, vol. 91, pp. 160–187, 2003.

[8] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction," in *Proceedings of Conference on Computer Vision and Pattern Recognition Workshop*, vol. 5, Madison, Wisconsin, 16-22 June 2003, pp. 53–58.

[9] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 200–205.

[10] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357–1362, 1999.

[11] L. Wiskott, J. Fellous, N. Krüger, and C. v. d. Malsburg, "Face recognition by elastic bunch graph matching." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, July 1997.

[12] G.Guo and C.R.Dyer, "Learning from examples in the small sample case: Face expression recognition," *IEEE Transactions on Systems, Man, And Cybernetics-Part B: Cybernetics*, vol. 35, no. 3, pp. 477–488, June 2005.

[13] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron," in *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, Nara Japan, 14-16 April 1998, pp. 454–459.

[14] B. Fasel, "Multiscale facial expression recognition using convolutional neural networks," IDIAP, Tech. Rep., 2002.

[15] M. Matsugu, K. Mori, Y.Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5-6, pp. 555–559, June-July 2003.

[16] M. Rosenblum, Y. Yacoob, and L. S. Davis, "Human expression recognition from motion using a radial basis function network architecture," *IEEE Transactions on Neural Networks*, vol. 7, no. 5, pp. 1121–1138, September 1996.

[17] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," *IEEE Transactions on Systems, Man, And Cybernetics-Part B: Cybernetics*, vol. 34, no. 3, pp. 1588–1595, June 2004.

[18] S. Dubuisson, F. Davoine, and M. Masson, "A solution for facial expression representation and recognition," *Signal Processing: Image Communication*, vol. 17, no. 9, pp. 657–673, October 2002.

[19] X.-W. Chen and T. Huang, "Facial expression recognition: A clustering-based approach," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1295–1302, June 2003.

[20] Y. Gao, M. Leung, S. Hui, and M. Tananda, "Facial expression recognition from line-based caricatures," *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, vol. 33, no. 3, pp. 407–412, May 2003.

[21] B. Abboud, F. Davoine, and M. Dang, "Facial expression recognition and synthesis based on an appearance model," *Signal Processing: Image Communication*, vol. 19, no. 8, pp. 723–740, 2004.

[22] I. A. Essa and A. P. Pentland, "Facial expression recognition using a dynamic model and motion energy," in *Proceedings of the International Conference on Computer Vision (ICCV 95)*, Cambrdige, Massachusetts, 20-23 June 1995.

[23] ——, "Coding, analysis, interpretation, and recognition of facial expressions," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 757–763, July 1997.

[24] M. S. Bartlett, G. Littlewort, B. Braathen, T. J. Sejnowski, and J. R. Movellan, "An approach to automatic analysis of spontaneous facial expressions," in *Proceedings of 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*, Washington, D.C, 2002.

[25] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying Facial Actions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999.

[26] Y. L. Tian, T. Kanade, and J. F. Cohn, "Recognizing Action Units for facial expression analysis," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, February 2001.

[27] J. J. Lien, T. Kanade, J. Cohn, and C. C. Li, "Automated facial expression recognition based on FACS Action Units," in *Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition*, April 1998, pp. 390–395.

[28] J. J. Lien, T. Kanade, J. F. Cohn, and C. Li, "Detection, tracking, and classification of Action Units in facial expression," *Journal of Robotics and Autonomous Systems*, July 1999.

[29] Y. L. Tian, T. Kanade, and J. Cohn, "Evaluation of Gabor wavelet-based Facial Action Unit recognition in image sequences of increasing complexity," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 229–234.

[30] A. Tefas, C. Kotropoulos, and I. Pitas, "Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 735–746, 2001.

[31] H. Drucker, W. Donghui, and V. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048 – 1054, September 1999.

[32] A. Ganapathiraju, J. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2348 – 2355, August 2004.

[33] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637–646, 1998.

[34] M. Rydfalk, "CANDIDE: A parameterized face," Linkoping University, Tech. Rep., 1978.

[35] F. Dornaika and F. Davoine, "Simultaneous facial action tracking and expression recognition using a particle filter," in *Proceedings of IEEE International Conference on Computer Vision*, Beijing, China, 17-20 Oct 2005.

[36] S. B. Gokturk, C. Tomasi, B. Girod, and J.-Y. Bouguet, "Model-based face tracking for view-independent facial expression recognition," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, Cambridge, U.K., May 2002, pp. 287–293.

[37] M. Malciu and F. Preteux, "Tracking facial features in video sequences using a deformable model-based approach," in *Proceedings of the SPIE*, vol. 4121, 2000, pp. 51–62.

[38] P. Michel and R. Kaliouby, "Real time facial expression recognition in video using support vector machines," in *Proceedings of 5th international conference on Multimodal interfaces*, Vancouver, British Columbia, Canada, 2003, pp. 258–264.

[39] V. Vapnik, *Statistical learning theory*. New York: Wiley, 1998.

[40] J. Y. Bouguet, "Pyramidal implementation of the Lucas-Kanade feature tracker," Intel Corporation, Microprocessor Research Labs, Tech. Rep., 1999.

[41] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, March 2002.

[42] C. J. C. Burges, "A tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge discovery*, vol. 2, no. 2, 1998.

[43] B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A. Smola, "Input space vs. feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, September 1999.

[44] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, March 2001.

[45] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[46] S.Gunn, "Support vector machines for classification and regression," Image Speech and Intelligent Systems Group, Univ. of Southampton," MP-TR-98-05, 1998.

[47] MATLAB, *Users Guide*. The MathWorks, Inc., http:// www.mathworks.com, 1994-2001.

[48] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of classifier methods: A case study in handwritting digit recognition," in *Proceedings of International Conference on Pattern Recognition*, 1994, pp. 77–87.

[49] J. Weston and C. Watkins, "Multi-class Support Vector Machines, Tech. Rep. Techical report CSD-TR-98-04, 2004.

[50] ——, "Multi-class Support Vector Machines," in *Proceedings of ESANN99*, Brussels, Belgium, 1999.

[51] S. Krinidis and I. Pitas, "2-D physics-based deformable shape models: Explicit governing equations," in *Proceedings of First International Workshop on "Interactive Rich Media Content Production: Architectures, Technologies, Applications, Tools"*, Lausanne, Switzerland, 16-17 October 2003, pp. 43–55.

[52] P. Ekman and W. V. Friesen, "FACS - Facial Action Coding System," *availiable at http://www-2.cs.cmu.edu/afs/cs/project/face/www/facs.htm*, 1978.

[53] F. Dornaika and F. Davoine, "View- and texture-independent facial expression recognition in videos using dynamic programming," in *Proceedings of IEEE International Conference on Image Processing*, Genova, Italy, 11-14 September 2005.

[54] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *ICASSP*, 1989, pp. 532–535.