2011 UKSim 5th European Symposium on Computer Modeling and Simulation

Facial Expression Recognition In Image Sequences Using Active Shape Model and SVM

Rajesh A Patil
MNIT Jaipur
Email:rapatil_rtg@yahoo.co.in

Vineet Sahula
MNIT Jaipur
Email:sahula@ieee.org

A. S. Mandal
CEERI Pilani
Email:atanu@ceeri.ernet.in

***Abstract*—This paper introduces a method for automatic facial expression recognition in image sequences, which make use of Candide wire frame model and active appearance algorithm for tracking, and support vector machine for classification. Candide wire frame model is adapted properly on the first frame of face image sequence. Facial features in subsequent frames of image sequence are tracked using active appearance algorithm. The algorithm adapts Candide wire frame model to the face in each of the frames and tracks the grid in consecutive video frames over time. Last frame of image sequence corresponds to greatest facial expression intensity. The geometrical displacement of Candide wire frame 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 the support vector machine, which classifies facial expression into one of the class such as happy, surprise, sad, anger, disgust and fear.**

***Index Terms*—Candide wire frame model, active appearance model, SVM**

I. Introduction

The automatic acquisition and analysis of images to obtain desired data for interpreting a scene or controlling an activity is called machine vision. Machine vision is a difficult task - a task that seems relatively trivial to humans is infinitely complex for computers to perform. Current machine vision research concerns, not only understanding the process of vision, but also designing effective vision systems for various real world applications. All practical machine vision systems in use today exist for their own specific purposes. Facial expression recognition system is an example of machine vision system.

For a human being facial expression is one of the most powerful, natural and immediate means to communicate their emotions and intentions. Facial expressions can contain a great deal of information. Hence the demand of automatically extracting this information has been continuously increasing. Automatic facial expression analysis is an interesting and challenging problem, and impacts important applications in many areas such as human computer interaction, and data driven animation. Due to its wide range of applications, automatic facial expression recognition has attracted much attention in recent years. Various applications using automatic facial expression analysis can be envisaged in the near future, fostering further interest in doing research in different areas, including image understanding, psychological studies, facial nerve grading in medicine, face image compression and synthetic face animation, video indexing, robotics as well as virtual reality [1]. Though much progress has been made, recognizing facial expression with a high accuracy remains difficult due to subtlety, complexity and variability of facial expressions. An effective automatic expression recognition system could take human computer interaction to the next level. Although facial expression recognition looks simple. it is very difficult because of high variability that can be found in images containing a face. We can see an extremely large variety in lighting conditions, resolution, pose and orientation.

II. Related work

Good surveys on the research made regarding facial expression recognition in image sequences can be found in [1] and [2]. Facial expression recognition problem in image sequences can be divided into three sub problems. Face detection- before a facial expression can be analyzed, the face must be detected in a scene. Different methods used for face detection are eigenface, Canny edge detector, brightness distribution, skin color detection etc [2]. Feature extraction and tracking- to develop a mechanism for extraction of the facial expression information from the observed facial image sequence and then track these features in subsequent frames. Prominent physical features of the face constitute “facial feature” viz, eyebrows, eyes, nose, mouth, and chin. For feature extraction, researchers have explored many techniques like labeled graph, point distribution model, brightness distribution, optical flow computation, potential net fitting, Gabor wavelets. While for tracking they have used Kalman filters, active appearance algorithm and optical flow computation methods. Classification- to develop a mechanism to classify facial expressions into one of the basic facial expression. Different methods used by researchers for classifying facial expressions are hidden markov model, neural networks, principal component analysis and linear discriminant analysis, and support vector machines.

Essa and Pentland [3] are utilizing the eigenspace method to automatically track the face in the scene and extract the positions of the eyes, nose, and mouth. They applied principal component analysis (PCA) on a sample of 128 images. They used optical flow computation method to obtain motion estimates and error-covariance information. They generated the spatio-temporal templates for six different expressions two facial actions (smile and raised eyebrows) and four emotional expressions (surprise, sadness, anger, and disgust). The Euclidean norm of the difference between the motion energy

978-0-7695-4619-3/11 $26.00 © 2011 IEEE
DOI 10.1109/EMS.2011.25

template and the observed image motion energy is used as a metric for measuring similarity/dissimilarity [3].

Kimura and Yachida [4] utilize a potential net for face representation. They fit a potential net to each frame of the facial image sequence under consideration. The pattern of the deformed net is compared to the pattern extracted from an expressionless face (usually the first frame of the sequence), and the variation in the position of the net nodes is used for further processing. Kimura and Yachida built an emotion space by applying PCA on six image sequences carrying three expressions anger, happiness, and surprise shown by a single person gradually, from expressionless to a maximum intensity of expression. The eigenspace spanned by the first three principal components has been used as the emotion space, onto which an input image is projected for a quantified emotional classification [4].

Cohn et al. [5] have used a model of facial landmark points localized around the facial features, hand-marked with a mouse device in the first frame of an examined image sequence. In the rest of the frames, a hierarchical optical flow method is used to track the optical flow of 13×13 windows surrounding the landmark points. The displacement vectors, calculated between the initial and the peak frame, represent the facial information used for recognition of the displayed facial expression [5]. Seyed Mehdi Lajevardi and Margaret Lech [6] have proposed a method which is fully automatic. They have used Viola Jones method along with Adaboost algorithm for face detection. They have made use of Log Gabor filters for feature extraction, and Naive Bayesian (NB) classifier for feature classification.

Wang et al. [7] utilize 19 facial feature points (FFPs) - seven FFPs to preserve the local topology and 12 FFPs for facial expression recognition. The FFPs are treated as nodes of a labeled graph that are interconnected with links representing the Euclidean distance between the nodes. The initial location of the FFPs in the first frame of an input image sequence is assumed to be known. The FFPs are tracked in the rest of the frames. The correspondence between the FFPs tracked in two consecutive frames is treated as a labeled graph matching problem. The degree of expression change is determined based on the displacement of the FFPs in the consecutive frames [7].

M. Valstar et al. [8] proposed a system that performs action unit (AU) recognition using temporal templates as input data. Temporal templates are introduced by Bobick and Davis. These templates are 2D images constructed from image sequences, effectively reducing a 3D spatio-temporal space to a 2D representation. They have used Neural Network as a classifier. Black and Yacoob [9] are using local parametrized models of image motion for facial expression analysis. The motion parameters (e.g., translation and divergence) are used to derive the mid level predicates that describe the motion of the facial features. For each of the six basic emotional expressions, they have developed a model represented by a set of rules for detecting the beginning and ending of an expression. The rules are applied to the predicates of the mid level representation.

In [10] Irene Kotsia and Pitas have proposed a method, which 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 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 the facial action coding system (FACS). A well known Kanade Lucas Tomasi (KLT) tracker is used for tracking facial features in subsequent frames. The geometrical displacement of these nodes, defined as the difference of coordinates of each node at the first and the last frame of the facial image sequence, is used as an input to a support vector machine classifier.

In our proposed work we make use of Viola Jones algorithm [11] for face detection in an image sequence upon which Candide wire frame model is fitted . Facial feature tracking is done using active face model proposed by J. Ahlberg [12]. Expression classification is performed by multi class SVM. Let us consider an image sequence containing face. The face is detected using Viola Jones algorithm. First frame of the sequence gives neutral facial expression and last frame corresponds to fully expressed state. Our approach automatically fits Candide wire frame model on the first frame of the image sequence. The tracking system allows the grid to follow the evolution of the facial expression over time till it reaches its highest intensity, producing the deformed Candide grid at each video frame. We use active face model for tracking. The geometrical displacement of Candide wire frame nodes, defined as the difference of coordinates of each node at the first and the last frame of the facial image sequence, is used as an input to a multi class SVM classifier, which is used to classify facial expression into one of the classes such as happy, anger, sad, surprise, disgust, and fear. Our framework differs from the one proposed by Irene Kotsia and Pitas [10], who have used pyramidal Kanade Lucas Tomasi tracker [13], which is based on optical flow computation. We make use of Active face model proposed by J. Ahlberg [12] for tracking. The rest of the paper is organized as follows. The system used for facial expression recognition is described in Section III. Results are presented in Section IV. We conclude, summarize and discuss limitations in Section V.

III. FACIAL EXPRESSION RECOGNITION SYSTEM

The proposed framework is composed of three subsystems. First is used for face detection. Second is for Candide grid node coordinate displacement extraction and third is used for grid node displacement classification. Face detection is performed by Viola Jones algorithm. Candide wire frame model is fitted automatically on first frame of image sequence. In the subsequent frames Candide grid nodes are tracked using Active Appearance Algorithm. The grid node information extraction is performed by Active face model tracking system, while the grid node information classification is performed by a multi class SVM system. The flow diagram of the proposed framework is shown in Figure 1.

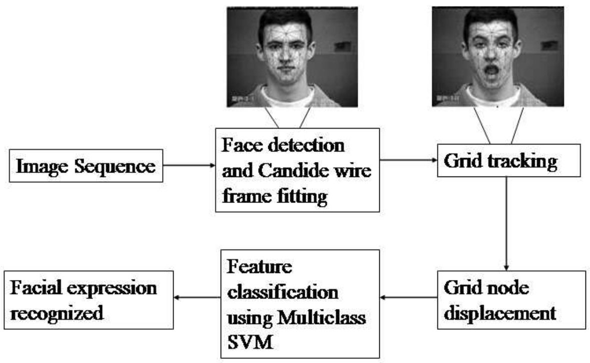

Figure 1. Flow diagram for facial expression recognition system

A. Face Detection

In present work, as we propose to make the system to be fully automatic, we start by detecting the user's face inside the scene. Although, we seemed it an easy problem at first, we immediately realized that the high variability in the types of faces encountered would make the automatic detection of the face a tricky problem. Many different techniques have been reported in the literature for face detection. We have selected Viola Jones algorithm [11] for face detection. The result of face detection algorithm is illustrated in Figure 2.

Figure 2. Face detection

B. Extraction of Candide grid node coordinates

1) Candide wire frame model: The facial wire frame model, we have used in the tracking procedure is the Candide wire frame model [14]. Candide wire frame is a parametrized face mask specifically developed for model-based coding of human faces. A frontal view of the model is shown in Figure 3. It has 113 vertices and 184 triangles. The small number of its triangles, allows fast face animation with moderate computing power. Candide model can be completely described by a set of vertices, a set of surfaces, and a set of deformation units. A vertex is described by a set of three coordinates telling the position of the vertex in a local 3-D coordinate system, and a surface is described by three integers telling which vertices to connect in the triangle that constitutes the surface. A deformation unit is either an action unit or a shape unit.

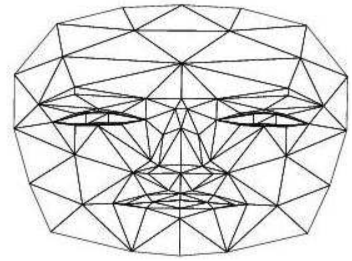

Figure 3. Candide wire frame model [14]

The geometry of the model as discussed in [14] can be expressed as in (1).

$$V(\sigma,\alpha) = \overline{V} + \sum_{i=1}^{14} S_i\sigma_i + \sum_{i=1}^{65} A_i\alpha_i \tag{1}$$

Here the resulting vector V contains (x,y,z) coordinates of vertices of the model. $\overline{V}$ is vector containing vertex coordinates of standard model. S_i represents a shape unit. There are 14 shape units, such as head height, mouth width, eyebrows vertical position, eyes width etc. The parameter σ_i is shape parameter. A_i represents animation unit. There are 65 animation units such as lip stretched, nose wrinkle, inner brow raiser, outer brow raiser etc. Whereas α_i is animation parameter. The difference between shape and animation modes is that the shape modes define deformations that differentiate individuals from each other, while the animation modes define deformations that occur due to facial expression. To perform global motion of the model, six more parameters are added to formula in (1), three for rotation, one for scaling, and two for translation

$$V(R,s,\sigma,\alpha,t) = Rs(\overline{V} + S\sigma + A\alpha) + t \tag{2}$$

Here $R = (\theta_x, \theta_y, \theta_z)$ is rotation matrix, s is scale, and $t = (t_x, t_y)$ is a 2D translation vector. The geometry of the model is thus parametrized by (3)

$$p = [\theta_x, \theta_y.\theta_z, s, t_x, t_y, \sigma, \alpha]^T \tag{3}$$

Once the model is adapted properly on the first frame, for the subsequent frames only α will change. Our goal is to find the optimal adaptation of the model to the input image i.e. to find p that minimizes the distance between the model and the image. The Candide model is adapted to a set of images using different parameters: 3D -rotation, 2D -translation, scale, and action units. We collect those parameters in a vector p, which thus parametrizes the geometry of the model. For each image in the training set, the image under the wire frame model is mapped to the model, and the model is then normalized to a standard shape, size, and position, in order to collect a geometrically normalized set of textures. On this set of textures, a Principal component analysis (PCA) has been performed and the eigentextures (geometrically normalized eigenfaces) have been computed [12].

$$x = \bar{x} + X\xi, \tag{4}$$

where $\bar{x}$ is mean texture, X is eigen texture and ξ is texture parameter. We can now describe the complete appearance of the model by the geometry parameters p and an N dimensional texture parameter vector, where N is the number of eigentextures we want to use for synthesizing the model texture. Given an input image and a p, the texture parameters are given by projecting the normalized input image on the eigentextures, and thus p is the only necessary parameter in our case. Geometrical normalization of the face used to obtain its normalized texture removes texture variations caused by its global and local motion and geometrical differences between individuals. We chose to work with 33 × 40 pixels images which are conveniently small and effective for image warping.

2) Texture Synthesis: We perform PCA on the training set (stored as 33 × 40 texture vector) so that we obtain the principal modes of variation, i.e., the eigenfaces. In this case, we collect 32 eigenfaces in a matrix X which could represent 90% of variance. A face vector j can be parametrized as in (5) where $\bar{x}$ is the mean face, and we could synthesize a face image according to (6) .

$$\xi = X^T(j - \bar{x}), \tag{5}$$

$$x = \bar{x} + XX^T(j - \bar{x}) \tag{6}$$

3) Tracking: Facial tracking means to find optimal adaptation of model to frames in image sequence. It can be obtained by finding the parameter vector p that minimizes the distance between normalized and synthesized faces. The initial value of p we use is the optimal adaptation to the previous frame. Assuming that the motion from one frame to another is small enough, we reshape the model to $V(p)$ and map the image i (the new frame) onto the model. Then we geometrically normalize the shape and get the resulting image as a vector.

Initially, we map the input Image *(i)* on the model. Then the model is geometrically normalized to the standard shape and we get the resulting normalized image as a vector, given by (7). We then are able to compute texture parameters from normalized image, as in (8).

$$j(i,p) = j(i,V(p)) \tag{7}$$

$$\xi(i,p) = X^T(j(i,p) - \bar{x}) \tag{8}$$

Synthesized texture would be given as in (9); residual image is calculated as in (10); Summed square error (SSE) is selected as error measure and is given by (11).

$$x(i,p) = \bar{x} + XX^T(j(i,p) - \bar{x}) \tag{9}$$

$$r(i,p) = j(i,p) - x(i,p) \tag{10}$$

$$e = \|r(i,p)\|^2 \tag{11}$$

For good model adaptation residual image and error e is much smaller. We then find the update vector $\triangle p$ by using update matrix U as in (12). The new error measure for updated parameter is given by (13).

$$\triangle p = Ur(p) \tag{12}$$

$$e_0 = \|r(i, p + \triangle p)\|^2 \tag{13}$$

If $e_0 < e$ we update $e_0 \to e$ and $p + \triangle p \to p$; else, we try smaller steps. Then e is recomputed as in (14); e.g. for $k = 1,2,3,\ldots\ldots\ldots$ if $e_k < e$ we update $e_k \to e$ and $p + \frac{1}{2^k}\triangle p \to p$. This scheme is iterated and the convergence is declared when $e_k > e$.

$$e_k = \left\| r(i, p + \frac{1}{2^k}\triangle p) \right\|^2 \tag{14}$$

4) Creating Update Matrix: Assuming that $r(i,p)$ is linear in p that is

$$\frac{\partial}{\partial p} r(i,p) = G \tag{15}$$

Taylor expanding $r(i,p)$ around $p + \triangle p$ would be given by (16). We want to find $\triangle p$ that minimizes error as in (17). This minimization is a least square problem with the solution given in (18) and update matrix U as the negative pseudo inverse of the gradient matrix G given in (19).

$$r(i, p + \triangle p) = r(i,p) + G\triangle p + O(\triangle p^2) \tag{16}$$

$$e(i, p + \triangle p) = \|r(i,p) + G\triangle p\|^2 \tag{17}$$

$$\triangle p = -(G^T G)^{-1} G^T r(i,p) \tag{18}$$

$$U = -(G^T G)^{-1} G^T \tag{19}$$

Gradient matrix G is calculated from training images in advance. j^{th} column in G is given by (20). The approximation could be taken as in (21), where h is the step size for perturbation and q_j is a vector with one in j^{th} column and zero in the rest elements.

$$G_j = \frac{\partial}{\partial p_j} r(i,p) \tag{20}$$

$$G_j \approx \frac{r(i, p + hq_j) - r(i, p - hq_j)}{2h} \tag{21}$$

The Candide wire frame model was adapted to every training image in the training set to compute the shape and texture modes. So, a set of corresponding parameter vectors p_n is obtained for a suitable step size to estimate G_j by averaging as in (22), where N is the number of training images and K is the number of steps to perturb the parameter [15].

$$G_j \approx \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{r(i_n, p_n + khq_j) - r(i_n, p_n - khq_j)}{2h} \tag{22}$$

C. Classification using SVM

The classification is performed only on the basis of geometrical information, not taking into consideration any luminance or color information. The geometrical information used is the displacement of one point, defined as the difference between the last and the first frame's coordinates. For every image sequence to be examined, a feature vector is constructed, containing the geometrical displacement of every point taken into consideration. The feature vector is used as an input to a multi class Support Vector Machine system, with six classes, that classifies each set of Candide grid node's geometrical displacements to one of the six basic facial expressions happy, surprise, sad, anger, fear, disgust. SVM classifier is a well suited for classifying facial expressions, as it is robust to the number of features, and known to model data in a highly optimized way. Basically, SVMs maximize the hyper plane margin between different classes. They map input space into a high dimension linearly separable feature space. This mapping does not affect the training time because of the implicit dot product and the application of the kernel function. In principle the SVM technique finds the hyper plane from the number of candidate-hyper planes, which has the maximum margin. The margin is enhanced by support vectors, which are lying on the boundary of a class. See the appendix.

Let μ be the video database that contains the facial image sequences. It is clustered into four different classes μ_k, $k = 1,........,6$, each one representing one of the six basic facial expressions. The geometrical information used for facial expression recognition is the displacement of one node d_{ij}, defined as the difference of the i^{th} grid node coordinates at the first and fully formed expression facial video frame

$$d_{ij} = \quad [\triangle x_{ij} \quad \Delta y_{ij}]^T \tag{23}$$

where $i = 1,........,E$ and $j = 1,...........,N$, and $\triangle x_{ij}$, $\triangle y_{ij}$ are the x, y coordinate displacement of the i^{th} node in the j^{th}image respectively. E is the total number of nodes and N is the number of the facial image sequences. This way, for every facial image sequence in the training set, a feature vector g_j is created given by (24). The vector g_j is called grid deformation feature vector, which contains the geometrical displacement of every grid node.

$$g_j = \left[d_{1,j} d_{2,j}.................d_{E,j}\right]^T \tag{24}$$

Here $j = 1,..........,N$. The dimension of vector g_j is $D = 113 \times 2 = 226$ dimensions. Each grid deformation feature vector g_j belongs to one of the six facial expression classes.

The multiclass SVMs problem solves only one optimization problem. It constructs six facial expression rules, where k^{th}function $W_k^T \phi(g_j) + b_k$ separates training vectors of class K from the rest of the vector by minimizing the objective function as given in (25),

$$\min_{w,b,\xi} \sum_{k=1}^{6} W_k^T W_k + C \sum_{j=1}^{N} \sum_{k \neq l_j} \xi_j^k \tag{25}$$

with the constraints $W_{l_j}^T \phi(g_j) + b_{l_j} \geq W_k^T \phi(g_j) + b_k + 2 - \xi_j^k$ $\xi_j^k \geq 0,\ j = 1,..........,N,\ k\{1,.........,6\}$. Where ϕ is the function that maps deformation vectors to high dimensional space. C is the term that penalizes error, and g_j is grid deformation vector. $b = [b_1,..........,b_6]$ is bias vector, and $\xi = \left[\xi_1^1,..........,\xi_i^k,............,\xi_N^6\right]$ is the slack variable vector [10]. The decision function is given in (26).

$$h(g) = \underset{k=1,...,6}{\arg\max}(W_k^T \phi(g) + b_k) \tag{26}$$

Using this decision function, a test grid deformation feature vector is classified to one of the six facial expressions.

IV. Experimental results

The Cohn-Kanade data base [16] has been used for constructing the update matrix as well as for SVM training. Viola Jones algorithm is successfully used for face detection in a scene. Candide wire frame model is fitted on the first frame and tracked in subsequent frame using active appearance algorithm. For texture synthesis, 40 images of different persons with different expressions are considered as training images. Candide model is manually adapted to these images, these images are then geometrically normalized (33 × 40 pixels) to standard shape, and then PCA is performed on them. For constructing update matrix we have selected 7 animation units. Theses are upper lip raiser, jaw drop, lip stretcher, eyebrow lowerer, eyebrow raiser, lip corner depressor, and nose wrinkler. Candide model is manually adapted to 40 training images, and then all the parameters have been perturbed one by one in steps of 0.01 in the range [-0.2, 0.2] to collect residual images. Number of steps K selected is 20, and number of images, N=40. Then the update matrix U is computed. On each frame update matrix is multiplied with residual image to get updated parameter vector. Difference between vertex coordinates of the first and last frame is used as input to SVM. SVM is trained using image sequences from Cohn-Kanade database. From the database 25 image sequences of each class are selected for training. When only four expressions (happy, sad, anger, surprise) are considered overall accuracy is 85%. Confusion matrices and accuracy for four classes is shown in Table I. But when two expressions (fear and disgust) are added, accuracy decreases to 69.88%. Fear expression creates confusion with disgust and anger, while disgust creates confusion with sad and anger. Confusion matrices and accuracy for six classes is shown in Table II. Work is going on to improve classification accuracy for more number of expressions.

Table I
CONFUSION MATRIX AND ACCURACY FOR FOUR CLASSES

Expression	*Happy*	*Surprise*	*Sad*	*Anger*
Happy	84%	6%	9%	8%
Surprise	16%	94%	0	0
Sad	0	0	66%	0
Anger	0	0	25%	92%

Table II
CONFUSION MATRIX AND ACCURACY FOR SIX CLASSES

Expression	*Happy*	*Surprise*	*Sad*	*Anger*	*Disgust*	*Fear*
Happy	69%	0	9%	0	0	0
Surprise	8%	88%	0	0	0	0
Sad	0	0	66%	0	24%	0
Anger	8%	0	25%	77%	30%	26%
Disgust	0	6%	0	15%	46%	8%
Fear	15%	6%	0	8%	0	66%

V. Conclusions

The successful use of active shape model alongwith SVM classifier has been illustrated for facial expression recognition. The tracking system is based on active shape model, which uses active appearance algorithm. Face is detected from a scene in a first frame of image sequence and Candide wire frame model is automatically fitted on it. As the facial expression changes in subsequent frames, model deform its shape. When the last frame is reached, model is fully deformed. Difference between Candide grid node coordinates of first and last frame is given as input to a six class SVM system. Only geometrical information is given to SVM, no texture information is given to SVM. The system is fully automatic i the sense that manual fitting of Candide wire frame model on first frame of image sequence under test, is not required. The method could still be used for color images. It recognizes only six basic facial expressions. Nevertheless, this is unrealistic since it is not at all certain that all facial expressions able to be displayed on the face can be classified under these basic emotion categories. It should be modified for more number of expressions. We are working on modifying this frame work for more number of expressions Work is still going on to adopt this frame work to track features in faces with head rotations. Major limitation of this method is requirement of rigorous training for constructing the update matrix, which is time intensive process.

APPENDIX

Support Vector Machine : An n-dimensional pattern x has n coordinates, $x = (x_1, x_2, ..., x_n)$, where each $x_i \varepsilon R$ for $i = 1, 2, ..., n$. Each pattern x_j belongs to a class $y_i \varepsilon \{-1, +1\}$. Consider a training set T of m patterns together with their classes, $T = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$. Consider a dot product space S in which the patterns x are embedded, $x_1, x_2, ...x_m \varepsilon S$. Any hyperplane in the space S can be written as in (27), and The dot product $W \cdot x$ is defined by (28).

$$\{x \varepsilon S \mid W \cdot x + b = 0\}, \; W \varepsilon S, \; b \varepsilon R \tag{27}$$

$$W \cdot x = \sum_{i=1}^{n} W_i x_i \tag{28}$$

A hyperplane $W \cdot x + b = 0$ can be denoted as a pair (W, b). A training set of patterns is linearly separable if at least one linear classifier exists defined by the pair (W, b), which correctly classifies all training patterns. All patterns from class +1 are located in the space region defined by $W \cdot x + b > 0$, and all patterns from class -1 are located in the space region defined by $W \cdot x + b < 0$. Using the linear classifier defined by the pair (W, b), the class of a pattern x_k is determined using (29). The distance from a point x to the hyperplane defined by (W, b) is given by (30), where $\| W \|$ is the norm of vector W. Of all the points on the hyperplane, one has the minimum distance d_{min} to the origin given by (31).

$$\text{class}(x_k) = \left\{ \begin{array}{ll} +1 \text{ if } & W \cdot x + b > 0 \\ -1 \text{ if } & W \cdot x + b < 0 \end{array} \right\} \tag{29}$$

$$d(x \; ; \; W, b) = \frac{\mid W \cdot x + b \mid}{\| W \|} \tag{30}$$

$$d_{min} = \frac{\mid b \mid}{\| W \|} \tag{31}$$

We consider the patterns from the class -1 that satisfy the equality $W \cdot x + b = -1$ and that determine the hyperplane H_1, the distance between the origin and the hyperplane H_1 is equal to $\mid -1 - b \mid / \| W \|$. Similarly, the patterns from the class +1 satisfy the equality $W \cdot x + b = -1$ and that determine the hyperplane H_2. The distance between the origin and the hyperplane H_2 is $\mid +1 - b \mid / \| W \|$. Hyperplanes H, H_1, and H_2 are parallel and no training patterns are located between hyperplanes H_1 and H_2. So the margin of linear classifier H is $2/\|W\|$. The optimum separation hyperplane conditions can be formulated into the following expression that represents a linear SVM, minimize $f(x) = \frac{\|W\|^2}{2}$ with the constraints $g_i(x) = y_i(W \cdot x_i + b - 1 \geq 0, i = 1, ..., m$.

REFERENCES

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

[2] M. Pantic and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1424–1445, Dec. 2000.

[3] I. A. Essa and A. P. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 757–763, July 1997.

[4] S. Kimura and M. Yachida, "Facial expression recognition and its degree estimation," *Proc. Computer Vision and Pattern Recognition*, pp. 295–300, 1997.

[5] J. F. Cohn, A. J. Ziochower, J. J. lien, and T. kanade, "Feature point tracking by optical flow discriminates subtle differences in facial expression," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, pp. 396–401, 1998.

[6] S. Lajevardi and M. Lech, "Facial expression recognition from image sequences using optimized feature selection," *23rd International Conference on Image and Vision Computing New Zealand*, pp. 1–6, 2008.

[7] M. Wang, Y. Iwai, and M. Yachindai, "Expression recognition from time-sequential facial images by use of expression change model," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, pp. 324–329, 1998.

[8] M. Valstar, I. Patras, and M. Pantic, "Facial action unit recognition using temporal templates," *13th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 253–258, 2004.

[9] M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *lnt'l J. Computer Vision*, vol. 25, no. 1, pp. 23–48, 1997.

[10] I. Kotsia and I. Pitas, "Facial expression recognition in image sequences using geometric deformation features and support vector machines," *IEEE Transactions on Image Processing*, vol. 16, pp. 172–187, Jan. 2007.

[11] P. Viola and M. Jones, "Robust real-time object detection," tech. rep., Cambridge Research Laboratory Technical report series, February 2001.

[12] J. Ahlberg, "Fast image warping for active models," Tech. Rep. Report No. LiTH-ISY-R-2355, Dept. of EE, Linkoping University, 2001.

[13] J. Y. Bouguet, "Pyramidal implementation of the Lucas-Kanade feature tracker," tech. rep., Intel Corporation, Microprocessor Research Labs, 1999.

[14] J. Ahlberg, "Candide-3 an updated parameterized face," Tech. Rep. Report No. LiTH-ISY-R-2326, Dept. of EE, Linkoping University, 2001.

[15] J. Ahlberg, "An active model for facial feature tracking," *EURASIP Journal on Applied Signal processing*, vol. 2002, pp. 566–571, 2001.

[16] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," *Proc. IEEE Int. Conf. Face and Gesture Recognition*, pp. 46–53, March 2000.

[17] C. Kyrkou and T. Theocharides, "Scope: Towards a systolic array for SVM object detection," *Embedded Systems Letters, IEEE*, vol. 1, no. 2, pp. 46 –49, 2009.

[18] A. Mathur and G. Foody, "Multiclass and binary SVM classification: Implications for training and classification users," *Geoscience and Remote Sensing Letters, IEEE*, vol. 5, no. 2, pp. 241 –245, 2008.

[19] H. Liu, Y. wei Huang, and D. Liu, "Multi-class surface EMG classification using support vector machines and wavelet transform," in *8th World Congress on Intelligent Control and Automation (WCICA)*, pp. 2963 – 2967, 2010.

[20] T. F. Cootes, G. Edwards, and C. J. Taylor, "Active appearance models," *Proc. 5th European Conference on Computer vision*, pp. 484–498, 1998.