Features classification using geometrical deformation feature vector of support vector machine and active appearance algorithm for automatic facial expression recognition.

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 proposes a method for facial expression recognition in image sequences. Face is detected from the scene and then facial features are detected using image normalization, and thresholding techniques. Using an optimization algorithm the Candide wire frame model is adapted properly on the first frame of face image sequence. In the subsequent frames of image sequence facial features are tracked using active appearance algorithm. Once the model fits on the first frame, animation parameters of model are set to zero, to obtain the shape of model for the neutral facial expression of the same face. The last frame of the image sequence corresponds to greatest facial expression intensity. The geometrical displacement of the Candide wire frame nodes, between the neutral expression frame and the last frame, is used as an input to the multiclass support vector machine, which classifies facial expression into one of the class such as happy, surprise, sadness, anger, disgust, fear and neutral. This method is applicable for frontal as well as tilted faces with angle $\pm 30^0$, $\pm 45^0$, $\pm 60^0$ with respect to y axis.**

***Index Terms*—Normalization, Feature extraction, active appearance model, SVM**

I. Introduction

Facial expressions play an important role in human-to-human communications, since they carry much information about human's feelings, emotions and so on. Automatic facial expression recognition system with high accuracy and performance will help to create human-like robots and machines that are expected to enjoy truly intelligent and transparent communications with humans. Humans detect and interpret faces and facial expressions in a scene with little or no effort, but for a machine this task is rather difficult. There are several related problems, such as face detection, feature extraction, tracking and classification. A system that would perform these tasks in real time with high accuracy will form a big achievement in human machine interaction. Though much progress has been made, recognizing facial expression with a high accuracy remains difficult due to subtlety, complexity and variability of facial expressions. 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.

The best known facial expression model was given by Ekman [1], who has argued that there are a neutral facial expression and six basic facial expressions corresponding to happiness, sadness, surprise, anger, disgust and fear. These seven classes of facial expressions are considered in this work. Facial expression recognition should not be confused with human emotion recognition as is often done in the computer vision community. While facial expression recognition deals with the classification of facial motion and facial feature deformation into abstract classes that are purely based on visual information, human emotions are a result of many different factors and their state might or might not be revealed through a number of channels such as emotional voice, pose, gestures, gaze direction and facial expressions.

II. Related work

Facial expression recognition in image sequences survey can be found in [2] and [3]. 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 [3].
- Feature extraction and tracking- to develop a mechanism for the extraction of the facial expression information from the observed facial image sequence and then track these features in subsequent frames. Prominent facial features of the face constitute eyebrows, eyes, nose and mouth. 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 expressions. 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.

In [4] Lajevardi and Margaret have proposed a method which is fully automatic. They have used Viola Jones method and Adaboost algorithm [5] for face detection. For feature extraction they have made use of log Gabor filters. Five scales and eight orientations were used to extract features from face images. This leads to 40 filter transfer functions representing different scales and orientations. For each training image a set of 4 log-Gabor filters with the smallest value of the spectral difference was selected. This reduces feature dimensions from 40 to only 4 arrays of size 60×60 for each image. Further reduction of the feature data was achieved by down-sampling the feature vectors by the factor of 4 to vectors of length 3600 samples per image. As a classifier they have made use of Naive Bayesian (NB) Classifier.

Y. Yacoob and L. S. Davis [6] proposed an approach, which is based on a statistical characterization of the motion patterns in specified regions of the face. They developed a region tracker for rectangles enclosing the face features. Each rectangle encloses one feature of interest, so the flow computation within the region is not contaminated by the motions of other facial features. To simplify the modeling of the eyebrows, they define the rectangles to include the eyes, and then subtract the rectangle of the eye from the combined rectangle. The tracking algorithm integrates spatial and temporal information at each frame. In order to enhance the tracking the statistics of the motion directions within a rectangle are used to verify translation of rectangles upward and downward and verify scaling of the rectangles. With the help of universal expression descriptions proposed by Ekman and Friesen, and motion patterns of expression proposed by Bassili, they prepare dictionary of facial feature actions (motion based feature description of facial actions). The dictionary is divided into components, basic actions of these components, and motion cues. The components are defined qualitatively and relative to the rectangles surrounding the face regions. Using component's visible deformations, the basic actions are determined. Using optical flow within these regions, the basic actions are determined. They designed a rule based system that combines certain expression descriptions. They have prepared rules in identifying the onsets of the beginning and the ending of each facial expression e.g. for Anger, beginning is inward lowering brows and mouth compaction and ending is outward raising brows and mouth expansion. These rules are applied to the mid level representation to create a complete temporal map describing the evolving facial expression.

Essa and Pentland [7] made use of eigenspace method proposed by Pentland et al. [8] to detect faces in an image sequence. They have applied principal component analysis (PCA) on a sample of 128 facial images and created face space in order to detect facial features. The distance of the observed image from the face space is calculated to detect the presence of face. To detect the location of the facial feature in a given image, the distance of each feature image from the relevant feature space is computed using an FFT. The extracted position of facial feature is used to normalize the input image. A two-dimensional (2D) spatio-temporal motion energy representation of facial motion between two subsequent normalized frames is used as a dynamic face model. They employ optical flow computation method proposed by Simoncelli [9], which uses multiscale coarse to fine Kalman filter to compute motion estimates. The spatio-temporal templates are generated for six different expressions two facial actions (smile and raised eyebrows) and four emotional expressions (surprise, sadness, anger, and disgust) by learning ideal 2D motion views for each expression category. The Euclidean norm of the difference between the motion energy template and the observed image motion energy is used as a metric for measuring similarity (dissimilarity) [3]. The method is useful only for frontal view face image sequences.

Kimura and Yachida [10] make use of integral projection method proposed in [11] to detect facial features. The input image is normalized using the center of the eyes and the center of the mouth. A potential net is then fitted on the normalized image to model the face and its movement. To do that, they first compute edge image by applying differential filter. Then, in order to extract the external force, they apply Gaussian filter. The filtered image is called potential field and an elastic net model is placed over it. 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. They 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 classification [3].

Cohn et al. [12] proposed a method in which key feature points were manually marked with a computer-mouse around facial landmarks on the first frame of the image sequence. Each point is the center of a 13×13 flow window that includes horizontal and vertical flows. A hierarchical optical flow method proposed by Lucas and Kanade [13] is used to automatically track feature points in the image sequence. The displacement of each feature point is calculated by subtracting its normalized position in the first frame from its current normalized position. The resulting flow vectors (6 horizontal and vertical dimensions in the brow region, 8 horizontal and vertical dimensions in the eye region, 6 horizontal and vertical dimensions in the nose region, and 10 horizontal and vertical dimensions in the mouth region) are concatenated to produce a 12 dimensional displacement vector in the brow region, a 16-dimensional displacement vector in the eye region, a 12 dimensional displacement vector in the nose region, and a 20 dimensional vector in the mouth region [12]. Separate group variance-covariance matrices were used for classification. They used two discriminant functions for three facial actions of the eyebrow region, two discriminant functions for

three facial actions of the eye region, and five discriminant functions for nine facial actions of the nose and mouth region. [3].

Wang et al. [14] 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 proposed by Buhmann et al.[15]. For three emotion categories viz. anger, happy and surprise, they use 12 B-spline curves corresponding to facial feature points, one each for one FFP, in order to construct the expression model. Each curve gives the relationship between expression change and the displacement of the corresponding FFP. The expression is determined by the minimal distance between the actual FFPs and FFPs of model. The degree of expression change is determined based on the displacement of the FFPs in the consecutive frames [3].

M. Valstar et al. [16] have proposed a system that performs action units (AU) recognition using temporal templates as input data. Temporal templates have also been used by Bobick and Davis [17]. These templates are 2D images constructed from image sequences, effectively reducing a 3D spatio-temporal space to a 2D representation. To achieve this they first select 9 facial points from the first frame of the image sequence manually. These points are then tracked in all subsequent frames using a condensation based template tracking technique proposed by Isard and Blake [18]. They have used Neural Network as a classifier. They have used just a simple k-level neural network (kNN) based learning machine to classify an input image sequence into one of m facial expression classes, each of which corresponds either to an individual AU or to an AU-combination. The employed algorithm is straightforward for a test sample it uses a distance metric to compute which k-labeled training samples are nearest to the sample in question and then casts a majority vote on the labels of the nearest neighbors to decide the class of the test sample [16].

Cohen et al. [19] uses Piecewise B´ezier Volume Deformation (PBVD) tracker proposed by Tao and Huang [20]. This face tracker uses a model-based approach where an explicit 3D wire frame model of the face is constructed. In the first frame of the image sequence, landmark facial features such as the eye corners and mouth corners are selected interactively. The generic face model is then warped to fit the selected facial features. The face model consists of 16 surface patches embedded in B´ezier volumes. The surface patches defined this way are guaranteed to be continuous and smooth. The shape of the mesh can be changed by changing the locations of the control points in the B´ezier volume. Once the model is fitted head motion and local deformations of the facial features such as the eyebrows, eyelids, and mouth can be tracked. First the 2D image motions are measured using template matching between frames at different resolutions. Image templates from the previous frame and from the very first frame are both used for more robust tracking. The measured 2D image motions are modeled as projections of the true 3D motions onto the image plane. The recovered motions are represented in terms of magnitudes of some predefined motion of various facial features. Each feature motion corresponds to a simple deformation on the face. These motion vectors are referred as Motion-Units (MU's). They are similar but not equivalent to Ekman's AU's and are numeric in nature, representing not only the activation of a facial region, but also the direction and intensity of the motion. The MU's are used as the basic features for the classification. Bayesian classifier is used for classification.

Black and Yacoob [21] are using local parametrized models of image motion for facial expression analysis. The location of the face, eyes, eyebrows, and mouth are assumed to be known. They estimate the rigid motion of the face region between two frames using a planar motion model. This estimation is performed using a robust statistical approach to cope with violations of the rigid plane assumption. The motion of the face is used to register the images via warping and subsequently the relative motion of the feature regions is estimated in the coordinate frame of the face using exactly the same robust estimation procedure. The motion estimates of the face and features are used to predict their locations in the next frame and the process is repeated. The estimated motion parameters provide a simple abstraction of the underlying facial motions and can be used to classify the type of rigid head motion and the facial expression. 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 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 [3].

Irene Kotsia and Pitas [22] have proposed a method which is based on mapping and tracking the facial model Candide onto the video frames. They have used Candide wire frame model. 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 popular Kanade Lucas Tomasi tracker [23] 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.

Summary of the surveyed methods for automatic facial

feature extraction is presented in Table I. In all surveyed methods, face should be in frontal view, but in a method proposed by Black and Yacoob [21] head motions are allowed. In a method proposed by Wang et al. [14], and Cohn et al. [12] facial landmark points should be labeled by hand manually, while in a method proposed by Irene Kotsia and Pitas [22] the Candide wire frame model should be fitted manually on the first frame of the image sequence. All surveyed methods, except method proposed by Essa [7] assume that faces should be without hair and glasses.

Table I
COMPARING FEATURE EXTRACTION METHODS

Reference	Method	Remarks / Limitations
Seyed [4]	Log Gabor filters with gaussian transfer functions	Only front view faces without hair and glasses allowed
Yaser [6]	statistical characterization of motion pattern in specified regions of face	Only front view faces without hair and glasses allowed
Essa [7]	Optical flow method	Front views, Face with hair and glasses, light variation allowed
Kimura [10]	Potential net fitting to normalized face image by Gaussian filter	Only front view faces without hair and glasses allowed
Cohn [12]	Optical flow algorithm of Lucas Kanade	Front views, Face without hair and glasses, manual labeling on first frame
Wang [14]	labeled graph fitting	Front views, Face without hair and glasses, manual labeling on first frame
M. Valstar [16]	AU recognition using temporal templates	Front views, Face without hair and glasses
Cohen [19]	Piecewise B´ezier Volume Deformation (PBVD) tracker	Front views, Face with hair and glasses, manual labeling on first frame
Black [21]	Local parametrized model of image motion, optical algorithm	head motion and light variation allowed
Kotsia[22]	Candide wire frame model fitting and Pyramidal Kanade Lucas Tomasi tracker	Frontal views, face without glass allowed Manual fitting of model on first frame is necessary

Summary of the methods used for facial expression classification is given in Table II. In the surveyed methods Seyed Mehdi Lajevardi and Margaret Lech [4], and Irene Kotsia and Pitas [22] make the use of facial image sequences from Cohn Kanade database for testing. Seyed achieved accuracy of 68.9%, while Irene achieved 91.4% accuracy. In a method proposed by Kimura, Yachida and Wang et al. only three facial expressions anger, happiness and surprise are detected. Cohn et al. [12] detects action units, and from the combination of

Table II
COMPARING CLASSIFICATION METHODS

Ref.	Method	Test sequences	Accuracy
Seyed [4]	Naive Bayesian Classifier	172 sequences of 100 subjects from Cohn Kanade database	68.9%
Yaser [6]	rule based, prepared dictionary of rules	46 sequences of 32 subjects	—–
Essa [7]	Spatio temporal motion energy templates	22 sequences of 8 subjects	100%
Kimura [10]	3D emotion space (PCA)		—–
Cohn [12]	Discriminant functions	504 sequences of 100 subjects	88%
Wang [14]	Averaged Bsplines of feature trajectories	29 sequences of 8 subjects	95%
M. Valstar et al. [16]	Neural networks	Cohn kanade database	76.2%
Cohen [19]	Bayesian Classifier	Cohn kanade database	74%
Black [21]	temporal consistency of the mid level predicates which describes the motion of the facial features	70 sequences of 40 subjects	88%
Kotsia [22]	Multi class SVM	sequences from Cohn Kanade database	99.7%

action units they detect facial expressions.

From this survey we got the information that till today nobody got 100% accuracy. Most of the proposed methods are not real time, they are applicable only for frontal faces, tilted faces are not allowed. Most of the methods are semi automatic, they need initial fitting or labeling manually. The limitations in automatic expression recognition are to a large extent the result of high variability that can be found in images that contains a face. We will see an extremely large variety in lighting conditions, resolution, pose and orientation. In order to be able to analyze all these images correctly, an approach seems to be desirable that can detect and separate these source of variation from the actual information we are looking for. We use Active appearance model (AAM), which enables us to automatically create a model of a face in an image. The created models are realistic looking faces. Thus the variety in light variations, resolutions, pose and orientation will have no effect on expression recognition.

In our proposed work we make use Viola Jones algorithm for face detection in an image sequence. Facial features from the first frame of the image sequence, such as eyebrow corners, eye corners, nostrils and lip corners are detected using normalization and thresholding algorithm. The Candide wire frame

model is fitted on the first frame of the image sequence using optimization algorithm. Facial feature tracking in subsequent frames is done using active appearance algorithm proposed by Cootes [24]. 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. The last frame corresponds to fully expressed state. After fitting the model on the first frame, we set animation parameters of the model to zero in order to get the shape of model for the neutral facial expression of the same face. Geometrical displacement of the Candide node coordinates between the neutral facial expression frame and the last frame is given as input to multi class SVM. SVM is trained for the seven classes using Cohn Kanade database. SVM classifier, classifies facial expression into one of the seven classes such as happy, anger, sadness, surprise, disgust, fear and neutral. Our frame work differs from the one proposed by Irene Kotsia and Pitas [22], who have used pyramidal Kanade Lucas Tomasi tracker [23], which is based on optical flow computation and classifies the facial expression into one of the six basic expressions. In their approach the first frame should always be of neutral facial expression, which is always not possible. We make use of Active appearance algorithm for tracking and we classify facial expression into seven expressions. It is not necessary that the first frame should be always of neutral facial expression. We consider tilted faces making angle $\pm 30^0, \pm 45^0, \pm 60^0$ with respect to y axis. 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 four subsystems. The first subsystem is used for face detection. The second is for facial feature extraction and the Candide wire frame fitting on the first frame of the image sequence. The third subsystem is used for feature tracking and geometrical displacement extraction of the Candide grid nodes and the fourth subsystem is used for grid node displacement classification. Face detection is performed by Viola Jones algorithm. Facial feature extraction is performed by image normalization and thresholding techniques. Feature tracking in subsequent frames is performed by Active appearance algorithm, 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.

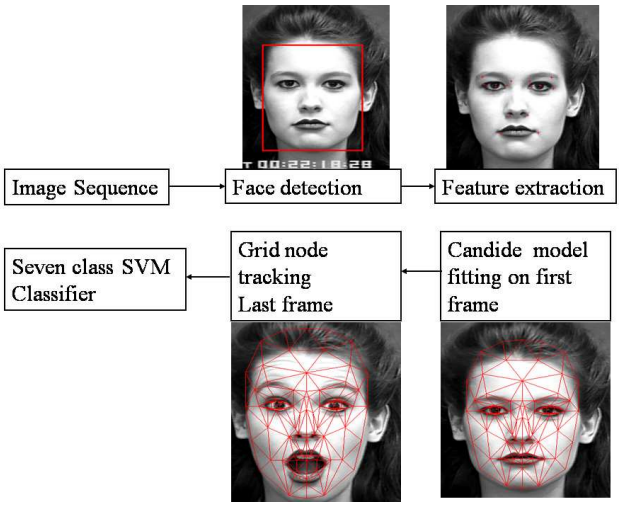

Figure 1. Flow diagram for facial expression recognition system

A. Face Detection

In present work, as we wish the system to be fully automatic, we have to 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. In our approach face area of an image was detected using the Viola Jones algorithm. The result of face detection algorithm is shown in Figure 2.

Figure 2. Face detection

B. Feature extraction

Detection and location of the face as well as extraction of facial features from images is an important stage for numerous facial image interpretation tasks. Detection of facial feature points, such as corners of eyes, lip corners, nostrils from the images are crucial. We have developed a method to detect 14 facial feature points such as eyebrows corners, eyes corners, eyeballs, nostrils and lip corners [25]. After detecting face region, the face image is divided into different regions of interest. It is divided horizontally into three parts, such that upper region contains eyes, middle contains nostrils, and lower contains mouth. Then the upper region is divided vertically into two parts such that each one will contain one eye. Each eye region is again divided horizontally into two parts so that

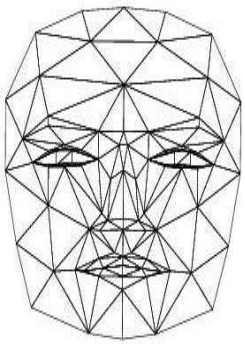

Figure 3. Candide wire frame model

eyes and eyebrows are separated. In eye regions horizontal and vertical histogram analysis is performed and peaks are found to detect eyeballs. In each region we perform image normalization and thresholding so that image is converted into binary image. Then the regions are scanned vertically to detect corners in order to get desired features. Our method detects feature points from expressionless as well as faces with expressions such as smile, surprise, sad etc.

C. Extraction of Candide grid node coordinates

1) Candide wire frame model: We have used Candide wire frame model for tracking. The Candide model was created by Mikael Rydfalk at the Linkoping Image Coding Group in 1987. Later, Bill Welsh [26] at British Telecom created another version with 160 vertices and 238 triangles covering the entire frontal head (including hair and teeth) and the shoulders. This version, known as Candide-2 is delivered with only six Action Units. A third version of Candide has been derived from the original one by Jorgen Ahelberg [27]. Candide wire frame is a parametrized face mask specifically developed for model-based coding of human faces. A frontal view of the model can be seen in Figure 3. It has 113 vertices and 184 triangles. The small number of its triangles allows fast face animation with moderate computing power.

The geometry of the model as discussed in [27] 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. There are 113 vertices. V is of dimension 3×113. $\overline{V}$ 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. Dimension of S is 3×113. The parameter σ_i is shape parameter. There are 65 animation units such as lip stretched, nose wrinkle, inner brow raiser, outer brow raiser etc. A_i represents animation unit and α_i is animation parameter. Dimension of A is 3×113. The difference between shape and animation mode 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 three for rotation, one for scaling, and two for translation are added to formula in (1).

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

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

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

p is a parameter vector of length 8×1 i.e. only global parameters.

2) Model fitting on face image: 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. We have set of feature points F obtained from our feature extraction algorithm. We have extracted 14 facial feature points, and each point has x and y coordinates, so F is vector of length 28×1. We know the vertices of Candide model $\overline{V}$. The goal of model adaptation is to find the deformed model V that fulfills

$$min \parallel V - F \parallel^2 \tag{4}$$

In above equation out of 113 vertices, we consider only 14 vertices corresponding to our 14 extracted feature points. Dimension of V is reduced to 28×1.

The action units and shape units can be applied to scaled model, which simplifies (2) as in (5). For small rotations we can write (6) and (7). Then we can write (5) as in (8)

$$V = R(s\overline{V} + S\sigma + A\alpha) + t \tag{5}$$

$$R\overline{V} = (\theta_x\theta_y\theta_z)\overline{V} = (I + r_x\theta_x)(I + r_y\theta_y)(I + r_z\theta_z)\overline{V} \tag{6}$$

$$R\overline{V} = (r_x\theta_x + r_y\theta_y + r_z\theta_z + I)\overline{V} \tag{7}$$

$$\begin{aligned} V =& (\sum_{i=1}^{3} \beta_i s_i)\overline{V} + \sum_{i=1}^{14} S_i\sigma_i + \sum_{i=1}^{65} A_i\alpha_i + (\sum_{i=x}^{y,z} r_i\theta_i)\overline{V}) \\ &+ \sum_{i=1}^{3} \tau_i t_i \end{aligned} \tag{8}$$

We write (8) as matrix vector multiplication as $V = Cp$ where C defines the allowed deformations as in (9) and p is the parameter vector as in (10). C is of size 28×88. (3+14+65+3+3) (scaling+shape+animation+rotation+translation). p is a parameter vector of size 88×1 which gives scaling, shape, animation, rotation and translation parameter values.

$$\begin{aligned} C =& [s_1\overline{V}, s_2\overline{V}, s_3\overline{V}, S_1, ...S_{14}, A_1, ... \\ & .....A_{65}, \theta_x\overline{V}, \theta_y\overline{V}, \theta_z\overline{V}, t_1, t_2, t_3] \end{aligned} \tag{9}$$

$$\begin{aligned} p =& [\beta_1, \beta_2, \beta_3, \sigma_1, ...\sigma_{14}, \alpha_1, ...\alpha_{65}, \\ & r_x, r_y, r_z, \tau_1, \tau_2, \tau_3] \end{aligned} \tag{10}$$

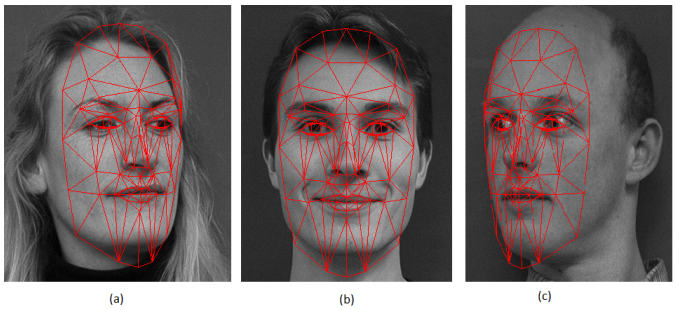

Figure 4. Candide model fitting on different faces (a) angle $+60^0$ (b) angle 0^0 (c) angle-60^0

So (8) can be solved as in (11) which is a least square optimization problem whose solution is as in (12)

$$min \parallel V - F \parallel^2 = min \parallel Cp - F \parallel^2 \tag{11}$$

$$p = (C^T C)^{-1} C^T F \tag{12}$$

Instead of computing 88 parameter values at a time, first we find the solution only for global parameters using (13) and (14)

$$C_1 = [s_1\overline{V}, s_2\overline{V}, s_3\overline{V}, \theta_x\overline{V}, \theta_y\overline{V}, \theta_z\overline{V}, t_1, t_2, t_3] \tag{13}$$

$$p_1 = [\beta_1, \beta_2, \beta_3, r_x, r_y, r_z, \tau_1, \tau_2, \tau_3] \tag{14}$$

$C_1$is of size 28×9 i.e. scaling, rotation and translation. p_1 is a parameter vector of size 9×1 which gives scaling, rotation and translation parameters. We consider rotation of faces only in y direction. Initially we set $\theta_y = 0$, then we find p_1. Then model is geometrically normalized to standard shape, texture is mapped on this model, synthesized image is computed. Then we compute residual image, from which summed squared e rror (SSE) is computed. All these methods are explained in subsequent sections. We repeat the procedure for $\theta_y = \pm 30^0, \pm 45^0, \pm 60^0$. The value of θ_y for which SSE is minimum, will give us rotating angle of face with respect to y axis. Then the shape unit parameters and animation unit parameters calculation is done using (15) and (16)

$$C_2 = [S_1, ...S_{14}, A_1, ...A_{65}] \tag{15}$$

$$p_2 = [\sigma_1, ...\sigma_{14}, \alpha_1, ...\alpha_{65}] \tag{16}$$

$C_2$is of size 28×79 i.e. animation and shape. p_2 is a parameter vector of size 79×1 which gives shape and animation parameters.Once the model is adapted properly on the first frame, for the subsequent frames only animation parameters α will change. For neutral frame $\alpha_1, ...\alpha_{65} = 0$. If the frame is of any expression other than the neutral then $\alpha_1, ...\alpha_{65} \neq 0$. From the analysis of different animation parameters we can recognize the facial expressions. For non neutral facial expression frame, once the model fits properly, then we set $\alpha_1, ...\alpha_{65} = 0$ so as to get shape of wire frame for neutral facial expression of the same face. So it is not necessary that first frame should be always of neutral facial expression. Model fitted on different faces, with different rotation is shown in Figure 4.

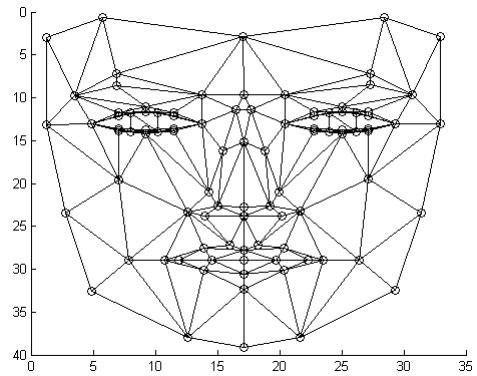

Figure 5. Geometrically normalized Candide model

3) Geometrically normalized model: 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 choose to work with 33×40 pixels images which are conveniently small and effective for image warping. Geometrically normalized model is shown in Figure 5.

4) Image Warping : A) Barycentric coordinate computation

Let us consider a triangle T defined by three vertices r_1, r_2, r_3. Any point r located on this triangle may then be written as a weighted sum of these three vertices, as in (17), where λ_1, λ_2, and λ_3 are the area coordinates. These are subjected to the constraint given in (18), and λ_3 is given by (19).

$$r = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3, \tag{17}$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{18}$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2 \tag{19}$$

Given a point r inside a triangle it is also desirable to obtain the barycentric coordinates λ_1, λ_2 and λ_3 at this point. We can write the barycentric expansion of vector r having Cartesian coordinates (x, y) in terms of the components of the triangle vertices (r_1, r_2, r_3) as

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \tag{20}$$

$$y = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3. \tag{21}$$

On substituting $\lambda_3 = 1 - \lambda_1 - \lambda_2$ into (20) and (21) results into (22) and (23).

$$x = \lambda_1 x_1 + \lambda_2 x_2 + (1 - \lambda_1 - \lambda_2) x_3 \tag{22}$$

$$y = \lambda_1 y_1 + \lambda_2 y_2 + (1 - \lambda_1 - \lambda_2) y_3 \tag{23}$$

On rearranging we obtain (24) and (25).

$$\lambda_1 (x_1 - x_3) + \lambda_2 (x_2 - x_3) + x_3 - x = 0 \tag{24}$$

$$\lambda_1 (y_1 - y_3) + \lambda_2 (y_2 - y_3) + y_3 - y = 0 \tag{25}$$

This linear transformation may be written as

$$R \cdot \lambda = r - r_3 \tag{26}$$

where, $R = \begin{bmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{bmatrix}$ $\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$ $r = \begin{bmatrix} x \\ y \end{bmatrix}$ and $r_3 = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$

Now the matrix *R* is invertible, since $r_1 - r_3$ and $r_2 - r_3$ are linearly independent (if this were not the case, then r_1, r_2 and r_3 would be collinear and would not form a triangle). Solving (26) for λ results into (27), (28), (29).

$$\lambda_1 = \frac{(y_2 - y_3)(x - x_3) - (x_2 - x_3)(y - y_3)}{\mid R \mid} \tag{27}$$

$$\lambda_2 = \frac{(x_1 - x_3)(y - y_3) - (y_1 - y_3)(x - x_3)}{\mid R \mid} \tag{28}$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2 \tag{29}$$

Since barycentric coordinates are a linear transformation of Cartesian coordinates, it follows that they vary linearly along the edges and over the area of the triangle. If a point lies in the interior of the triangle, all of the Barycentric coordinates lie in the open interval (0,1). If a point lies on an edge of the triangle, at least one of the area coordinates $\lambda_1, \lambda_2, \lambda_3$ is zero, while the rest lie in the closed interval [0,1]. Point *r* lies inside the triangle if and only if $0 < \lambda_i < 1 \ \forall i$ in 1,2,3. Otherwise, *r* lies on the edge or corner of the triangle if $0 \leq \lambda_i \leq 1 \ \forall i$ in 1,2,3.

Otherwise, *r* lies outside the triangle [28].

B) Texture mapping

We use a mesh of triangles (in two different shapes) to warp an image from one shape to another. In our case, the destination can be regarded as a two dimensional mesh, which simplifies things.

We thus have following.

- One destination mesh M containing triangles $T_1, ..., T_N$. Each triangle is a triplet of (2-D) vertex coordinates. That is, $T_n = [r_1, r_2, r_3]$ and $r_i = [\ x_i \quad y_i]^T$
- One source mesh M'.
- One source image *f* '(x).
- One destination image *f* (x)g.

The warping process can now be performed as

1) For each pixel coordinate *x* in the destination image, compute its barycentric coordinates $\lambda_1, \lambda_2, \lambda_3$ with respect to the first triangle T_1 in the destination mesh. If the barycentric coordinates are not valid, try the next triangle until the correct triangle T_n is found.
2) Compute the source coordinate *x*' from the barycentric coordinates applied to the corresponding triangle in the source mesh. That is, $x' = T_n'[\ \lambda_1 \quad \lambda_2 \quad \lambda_3\ ]^T$
3) Interpolate the source image *f* ' in *x*' and set *f*(*x*) = *f* '(*x*').

The object coordinates (the normalized/ destination shape) are fixed, and the texture coordinates are variable. This has an important implication: the barycentric coordinates are always the same for each pixel in the destination image. This means that we can compute them once, and store $\lambda_1, \lambda_2, \lambda_3$ and *n* for each pixel [29].

5) Texture Synthesis: 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 PCA has been performed and the eigentextures (geometrically normalized eigenfaces) have been computed [29], as in (30).

$$x = \overline{x} + X\xi, \tag{30}$$

Here, $\overline{x}$ 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. 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 (31), where $\overline{x}$ is the mean face, and we could synthesize a face image as in (32).

$$\xi = X^T(j - \overline{x}) \tag{31}$$

$$x = \overline{x} + XX^T(j - \overline{x}) \tag{32}$$

6) 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 as in (33).

$$j(i, p) = j(i, V(p)) \tag{33}$$

We compute texture parameters from normalized image as in (34). Then synthesized texture would be given as in (35), and residual image is calculated as in (36).

$$\xi(i, p) = X^T(j(i, p) - \overline{x}) \tag{34}$$

$$x(i, p) = \overline{x} + XX^T(j(i, p) - \overline{x}) \tag{35}$$

$$r(i, p) = j(i, p) - x(i, p) \tag{36}$$

Summed square error (SSE) is selected as error measure and is given as in (37). For good model adaptation residual image and error e is much smaller. Find the update vector $\triangle p$ by multiplying residual image with an update matrix U as in (38), and the new error measure for updated parameter is given as

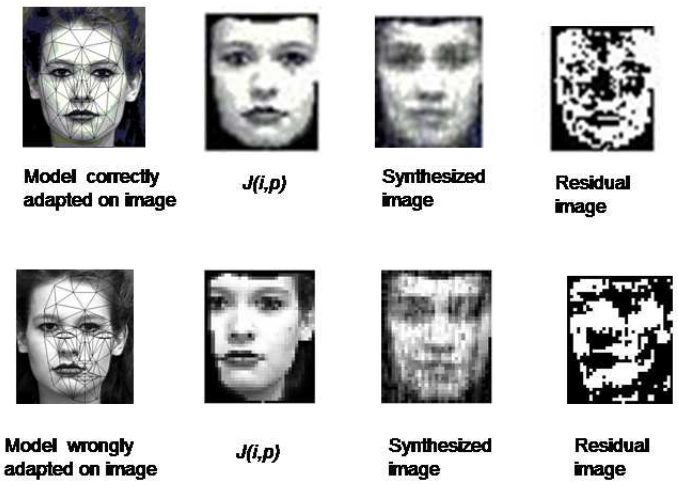

Figure 6. Candide wire frame model fitting on face image

in (39).

$$e = \|r(i,p)\|^2 \tag{37}$$

$$\triangle p = \mathrm{U} r(p) \tag{38}$$

$$e_0 = \|r(i,p+\triangle p)\|^2 \tag{39}$$

if $e_0 < e$ we update $e_0 \rightarrow e$ and $p + \triangle p \rightarrow p$ if not, try smaller steps. e is recomputed as

$$e_k = \left\| r(i, p + \frac{1}{2^k}\triangle p) \right\|^2 \tag{40}$$

for $k = 1, 2, 3, \ldots\ldots$ if $e_k < e$ we update $e_k \rightarrow e$ and $p + \frac{1}{2^k}\triangle p \rightarrow p$ Iterate the scheme and declare the convergence when $e_k > e$. The model matching and texture approximation process is shown in Figure 6. A correct model adaptation is shown in top row, and wrong adaptation is shown in bottom row. First image in both the rows show a model adapted on face image. Second image in both the rows is a texture of face image mapped on geometrically normalized Candide wire frame model. The normalized texture is approximated by the eigentextures producing the synthesized image. Residual image is computed by subtracting normalized image and synthesized image. From the figure it is cleared that normalized image and synthesized image are more similar for better model adaptation. Analysis of residual image tells us how to improve model adaptation.

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

$$\frac{\partial}{\partial p} r(i,p) = G \tag{41}$$

Taylor expanding $r(i,p)$around $p + \triangle p$ as in (42). We want to find $\triangle p$ that minimizes error as in (43).

$$r(i, p+\triangle p) = r(i,p) + G\triangle p + O(\triangle p^2) \tag{42}$$

$$e(i, p+\triangle p) = \|r(i,p) + G\triangle p\|^2 \tag{43}$$

Minimizing above equation is least square problem with the solution as in (44). Which gives update matrix U as the negative pseudo inverse of the gradient matrix G. as in (45).

$$\triangle p = -(G^T G)^{-1} G^T r(i,p) \tag{44}$$

$$\mathrm{U} = -(G^T G)^{-1} G^T \tag{45}$$

Gradient matrix G is calculated from training images in advance. j^{th} column in G is given by (46). The approximation could be as in (47).

$$G_j = \frac{\partial}{\partial p_j} r(i,p) \tag{46}$$

$$G_j \approx \frac{r(i, p+hq_j) - r(i, p-hq_j)}{2h} \tag{47}$$

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. 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

$$G_j \approx \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{r(i_n, p_n + khq_j) - r(i_n, p_n - khq_j)}{2h} \tag{48}$$

where N is the number of training images and K is the number of steps to perturb the parameter [30].

D. 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 6 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 [31]. 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.

Basically SVM is a binary classifier, which classifies data in two classes. To use it for multiclasses, mainly two schemes are used.

a) One against One multi-class SVM classifier*:* In this approach $C(C-1)/2$ classifiers are constructed. Where *C* is the number of classes. Classifier i, j is trained using all patterns from class i as positive instances, and all patterns from class *j* as negative instances and disregarding the rest. To combine obtained classifiers a simple voting scheme is used. When classifying a new instance each one of the base classifier casts a vote for one of the two classes used in its training. The class that gets maximum vote will be declared as class of new instance.

b) One against All multi-class SVM classifier*:* In this scheme there is one binary SVM for each class to separate members of that class from members of other classes. We have number of classifiers equal to number of classes. Classifier i, j is trained using all patterns from class *i* as positive instances, and all patterns from rest of the classes is assumed to be in class *j* as negative instances. The class for which decision function gives maximum value will be declared as class of new instance.

In our approach we have used one against all scheme. We have a video database that contains the facial image sequences. This database is clustered into seven different classes, each one representing one of the seven 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

$$\mathrm{d}_{ij} = \quad [\triangle x_{ij} \quad \Delta y_{ij}]^T \tag{49}$$

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 (50). The vector g_j is called grid deformation feature vector, which contains the geometrical displacement of every grid node.

$$g_j = [d_{1,j} d_{2,j} ..... d_{E,j}]^T \tag{50}$$

where $j = 1, ..., N$. 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 (51).

$$\min_{w,b,\xi} \sum_{k=1}^{7} W_k^T W_k + C \sum_{j=1}^{N} \sum_{k \neq l_j} \xi_j^k \tag{51}$$

with the constraints $W_{l_j}^T \phi(g_j) + b_{l_j} \geq W_k^T \phi(g_j) + b_k + 2 - \xi_j^k$ $\xi_j^k \geq 0$, $j = 1, ..., N$, $k \{1, ..., 7\}$. 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. $b = [b_1, ..., b_7]$ is bias vector, and $\xi = \left[\xi_1^1, ..., \xi_i^k, ..., \xi_N^7\right]$ is the slack variable vector [22]. The decision function is given in (52)

$$h(g) = \arg \max_{k=1,...,7} (W_k^T \phi(g) + b_k) \tag{52}$$

Using this procedure, a test grid deformation feature vector is classified to one of the seven facial expressions.

IV. Experimental results

We have used Cohn-Kanade database [32], and IMM database [33] 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 which is implemented in MATLAB. We use active appearance algorithm as well as SVM. They both need training. In training part we consider 40 face images of different persons with different facial expressions. The Candide wire frame model is manually adapted on these images and then these images are geometrically normalised to standard shape (33×40 pixels) . Principal component analysis is applied on them and then we collect 40 eigen faces in matrix X which could represent 90% variance. The candide model is again adapted manually on these 40 face images and all animation parameters have been perturbed one by one in steps of 0.01 in the range [-0.2, 0.2] to collect residual images. The number of steps we have selected is K=20, and number of images we have selected is N=40. Then using equation (48) we compute gradient matrix. From gradient matrix, using equation (44) we compute update matrix. For texture mapping we need to compute barycentric coordinate computation of triangles of Candide wire frame model. In case of SVM training we consider 25 image sequences of each expressions, i.e. 175 image sequences from the Cohn-Kanade database. On these image sequences model is fitted using our AAA, and then we extract coordinate difference between neutral and fully expressed frame to construct geometrical deformation feature vector g, which is used to train SVM. These are offline computations. In online computations, first global parameters $p_1$are computed. Then shape and animation parameters p_2 are computed. Once the model fits on the initial frame, animation parameters are made zero in order to get shape of model for same face with neutral facial expression. Then on each frame of image sequence animation parameters are updated inaccordance with facial expression. On each frame we compute synthesized image and geometrically normalised image and residual image. Residual image gets multiplied with update matrix to get updated animation parameters. Process is repeated till error becomes zero where model gets correctly adapted on face image. Same sequence is repeated till we reach to last frame. Then the difference between tha last frame and neutral frame vertex coordinates is given to SVM. Then SVM classifies the expression into one of the seven facial expressions. Some model fitting and tracking results are shown in figure 7.

Figure 7. Model fitting and tracking results

Table III
CONFUSION MATRICES AND ACCURACY WITH 4 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 IV
CONFUSION MATRICES AND ACCURACY WITH 7 CLASSES

Expression	Happy	Surprise	Sad	Anger	Disgust	Fear	Neutral
Happy	76%	0%	10%	0%	0%	0%	0%
Surprise	8%	90%	0%	0%	0%	0%	4%
Sad	0%	0%	73%	8%	28%	15%	0%
Anger	8%	0%	17%	84%	0%	0%	0%
Disgust	0%	6%	0%	8%	62%	15%	4%
Fear	8%	0%	0%	0%	0%	70%	0%
Neutral	0%	4%	0%	0%	10%	0%	92%

Confusion matrices and accuracy for four classes is shown in Table III. The confusion matrix is a $n \times n$ matrix containing the information about the actual class label (in its columns) and the label obtained through classification (in its rows). The diagonal entries of the confusion matrix are the rates of facial expression that are correctly classified, while the off diagonal entries correspond to misclassification rates. When only four expressions (happy, sad, anger, surprise) are considered overall accuracy is 85%. But when three expressions (fear and disgust and neutral) are added, accuracy decreases to 78%. Confusion matrices and accuracy for seven classes is shown in Table IV. Fear makes confusion with disgust and sad, while disgust makes confusion with sad. Accuracy depends on initial fitting of model on first frame. In [22] Kotsia and Pitas obtained 99.7% accuracy as shown in Table V, but initial fitting of model is done manually. In our case, it is done automatically. As they have considered only six basic facial expressions, neutral facial expression may be classified as one of the six basic facial expression, which is incorrect. Confusion matrices and accuracy obtained by Cohen et al. [19] is given in Table VI. They got average accuracy 73% for Cohn Kanade database. Our work is going on to improve classification accuracy for large number of expressions.

Table V
CONFUSION MATRICES AND ACCURACY WITH 6 CLASSES OBTAINED BY KOSTIA [22]

Expression	Happy	Surprise	Sad	Anger	Disgust	Fear
Happy	100%	0	0	0	0	0
Surprise	0	100%	0	0	0	0
Sad	0	0	100%	3.3%	0	0
Anger	0	0	0	96.7%	0	0
Disgust	0	0	0	0	100%	0
Fear	0	0	0	0	0	100%

Table VI
CONFUSION MATRICES AND ACCURACY WITH 7 CLASSES OBTAINED BY COHEN [19]

Expression	Happy	Surprise	Sad	Anger	Disgust	Fear	Neutral
Happy	86.22%	0	1.58%	4.76%	1.13%	13.57%	1.03%
Surprise	0	93.93%	3.17%	1.14%	2.27%	1.90%	1.03%
Sad	0	4.04%	61.26%	6.09%	9.09%	3.80%	5.78%
Anger	4.91%	0	13.25%	66.46%	10.90%	7.38%	3.51%
Disgust	5.65%	0	11.19%	14.28%	62.27%	7.61%	8.18%
Fear	3.19%	2.02%	3.96%	5.21%	10.9%	63.8%	1.85%
Neutral	0	0	5.55%	2.04%	3.40%	1.19%	78.59%

V. CONCLUSIONS

We have successfully used the active shape model and SVM for facial expression recognition. In our methodology, we use a tracking system based on active shape model, and 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 frame, model deform its shape. When the last frame is reached, model is fully deformed. When the model is fitted on first frame, animation parameters are set to zero, in order to get shape of model for neutral facial expression. Difference between Candide grid node coordinates of neutral facial expression frame and last frame is given as input to a seven class SVM system. Only geometrical information is given to SVM, no texture information is given to SVM. The system is fully automatic. Manual fitting of Candide wire frame model on first frame of image sequence is not required and it is not necessary that the first frame of image sequence should always be of neutral facial expression. We consider frontal as well as faces making angle $\pm 30^0, \pm 45^0, \pm 60^0$ with respect to y axis.

From the literature review, it is clear that most of the methods identifies facial expressions from frontal face images.The major limitation in detecting facial expressions of tilted or rotated faces is that they are view dependent between the training and testing face images i. e. we have to train the algorithm with face images of specific views. Hence we should know viewing angle of test images before the recognition procedure is performed, which is still a challenging problem in machine vision. We developed the algorithm to determine viewing angle w. r. t. y axis, and we are trying to develop it for x axis also. Presently, the system recognizes only seven basic facial expressions. Nevertheless, this is unrealistic on the grounds that 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. We propose and the work is undergoing to extend the technique for larger number of expressions (inspiration from Indian classical performing art). The proposed algorithm is applicable for color images also. It can be applied for tilted faces also. Manual intervention for accurate normalization of test faces and localization of feature points and manual warping of video sequences is not required. The recognition of facial expressions in image sequences with significant head movement is a challenging problem. It is required by many applications such as human-computer interaction and computer graphics animation. In the proposed approach head movement is allowed. To make the interaction with such systems faster, we are planning for the hardware implementation of proposed algorithm. One of the limitations of this method is that rigorous training is required for constructing the update matrix, which is computationally expensive.

REFERENCES

[1] P. Ekman and W. V. Friesen, "Facial action coding system manual," *Palo Alto consulting Psychologists Press*, 1978.

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

[3] 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.

[4] 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.

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

[6] Y. Yacoob and L. Davis, "Recognizing human facial expressions from long image sequences using optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 636–642, June 1996.

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

[8] A. Pentland, B. Moghaddam, and T. starner, "View based and modular eigenspaces for face recognition," *Proc. Computer vision and Pattern Recognition*, pp. 84–91, 1994.

[9] E. Simoncelli, "Distributed representation and analysis of visual motion," *PhD thesis, Massachusetts Inst. of Technology*, 1993.

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

[11] H. Wu, T. Yokoyama, D. Pramadihanto, and M. Yachinda, "Face and facial feature extraction from color image," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, pp. 343–350, 1996.

[12] 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.

[13] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. Joint Conf. Artificial Intelligence*, pp. 674–680, 1981.

[14] 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.

[15] J. Buhmann, J. Lange, and C. von der Malsburg, "Distortion invariant object recognition matching hierarchically labelled graphs," *Proc. Int'l Joint Conf. Neural Networks*, pp. 155–159, 1989.

[16] 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.

[17] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, 2001.

[18] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, 1998.

[19] I. Cohen, N. Sebe, A. Garg, L. S. Chen, and T. S. Huang, "Facial expression recognition from video sequences: Temporal and static modeling," *Computer Vision and Image Understanding*, vol. 91, pp. 160–187, 2003.

[20] T. Hai and T. Huang, "Connected vibrations: a modal analysis approach for non-rigid motion tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 735 –740, June 1998.

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

[22] 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.

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

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

[25] R. Patil, V. Sahula, and A. Mandal, "Automatic detection of facial feature points in image sequences," *International conference on Image information processing ICIIP*, Nov. 2011.

[26] B. Welsh, "Model based coding of images," *Ph.D. dessertation, British Telecom Research Lab*, Jan 1991.

[27] J. Ahlberg, "Wincandide 1.3 user's manual," Tech. Rep. Report No. LiTH-ISY-R-2344, Dept. of EE, Linkoping University, 2001.

[28] E. W. Weisstein, "Barycentric coordinates," *MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/BarycentricCoordinates.html*.

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

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

[31] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," *National Taiwan University, Taipei 106, Taiwan*.
[32] 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.
[33] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, "The IMM face database - an annotated dataset of 240 face images," may 2004.
[34] C. Kyrkou and T. Theocharides, "Scope: Towards a systolic array for SVM object detection," *Embedded Systems Letters, IEEE*, vol. 1, pp. 46 –49, aug. 2009.
[35] A. Mathur and G. Foody, "Multiclass and binary SVM classification: Implications for training and classification users," *Geoscience and Remote Sensing Letters, IEEE*, vol. 5, pp. 241 –245, april 2008.
[36] 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*, pp. 2963 –2967, July 2010.
[37] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw*, vol. 13, pp. 415–425, March 2002.
[38] J. Weston and C. Watkins, "Multi-class support vector machines," Tech. Rep. CSD-TR-98-04, 2004.
[39] J. Strom, F. Davoine, and J. Ahlberg, "Very low bit rate facial texture coding," *Proc. Int. Workshop on Synthetic/Natural Hybrid Coding and 3D Imaging*, pp. 237–240, September 1997.
[40] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Disc*, vol. 2, no. 2, 1998.
[41] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-training and application," *Computer Vision Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
[42] K. Anderson and W. Peter, "A real time automated system for the recognition of human facial expressions," *IEEE Trans. on systems, Man, and Cybernetics Part B Cybernetics*, vol. 36, pp. 96–105, Feb 2006.
[43] A. Lanitis, C. Taylor, and T. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 743–756, 1997.
[44] C. J. Lien, T. Kanade, J. F. Cohn, and C. Li, "Detection, tracking, and classification of action units in facial expressions," *J. Robot Auton. Sys.*, July 1999.
[45] Y. Tian, T. Kanade, and J. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 97–115, Feb 2001.
[46] Y. Ming-Hsuan, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, 2002.
[47] S. B. Gokturk, C. Tomasi, B. Girod, and J.-Y. Bouguet, "Model-based face tracking for view-independent facial expression recognition," *Proc. 5th IEEE Int. Conf. Automatic Face and Gesture Recognition.*, pp. 287–293, May 2002.
[48] M. Pontil and A. Verri, "Support vector machines for 3d object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 637–646, Jun 1998.
[49] H. Drucker, W. Donghui, and V. Vapniki, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, pp. 1048–1054, Sept 1999.
[50] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, "SVM and kernel methods matlab toolbox," *Perception Systems Information, INSA de Rouen, France*, 2005.