Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

Features classification using support vector machine for a facial expression recognition system

Rajesh A. Patil
Vineet Sahula
Atanendu S. Mandal

SPIE

IS&T
imaging.org

Features classification using support vector machine for a facial expression recognition system

Rajesh A. Patil
Vineet Sahula
Malaviya National Institute of Technology
ECE Department
JLN Marg, Jaipur 302017 India
E-mail: rpatil8rec106@gmail.com

Atanendu S. Mandal
CEERI Pilani
Rajasthan, India

Abstract. *A methodology for automatic facial expression recognition in image sequences is proposed, which makes use of the Candide wire frame model and an active appearance algorithm for tracking, and support vector machine (SVM) for classification. A face is detected automatically from the given image sequence and by adapting the Candide wire frame model properly on the first frame of face image sequence, facial features in the subsequent frames are tracked using an active appearance algorithm. The algorithm adapts the Candide wire frame model to the face in each of the frames and then automatically tracks the grid in consecutive video frames over time. We require that first frame of the image sequence corresponds to the neutral facial expression, while the last frame of the image sequence corresponds to greatest intensity of facial expression. 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 SVM, which classify the facial expression into one of the classes viz happy, surprise, sadness, anger, disgust, and fear. © 2012 SPIE and IS&T.* [DOI: 10.1117/1.JEI.21.4.043003]

1 Introduction

The automatic acquisition and analysis of images to obtain the 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 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. The facial expression recognition system is an example of machine vision system.

1.1 *Motivation*

For a human beings, 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, facial 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 the subtlety, complexity, and variability of facial expressions.[2] An effective automatic expression recognition system could take human computer interaction to the next level. According to Ekman and Friesen[3] there are six basic facial expressions: happiness, sadness, fear, disgust, surprise, and anger. Although facial expression recognition looks simple, it is very difficult because of high variability that can be found in images containing a face. For example, we can observe extremely large variations in lighting conditions, resolution, pose, and orientation.

1.2 *Computer Vision*

The ability of a machine or robot to understand human expression is essential for successful human machine interaction. Intelligent machines do not currently take full advantage of the information conveyed through multimodal human expressions. Fortunately, multimodal expressions can be understood by associating high level domain knowledge with low level perceived elements. Combining high level human knowledge with perceived elements in a closed loop system provides a robust and flexible strategy toward designing intelligent machines, like a smart patient monitoring system, interactive tutoring system, that fully understand the expression of human intent.

Many applications, such as virtual reality, video-conferencing, user profiling, and customer satisfaction studies

Paper 12067 received Feb. 28, 2012; revised manuscript received Aug. 15, 2012; accepted for publication Aug. 28, 2012; published online Oct. 1, 2012.

for broadcast and web services, clinical psychology, neurology, pain assessment, lie detection, intelligent environments, and multimodal human computer interface require efficient facial expression recognition in order to achieve the desired results.[4] Therefore, the impact of facial expression recognition in the above mentioned application areas is constantly growing. It can be widely applied as a part of an effort to develop basic techniques for space teleconferencing in which the machine can recognize human facial expressions and then reproduce the human facial images with realistic expressions in a remote location. Computers in the future will be able to offer advice in response to the mood of the users. The reaction of people in the test-panels could be automatically monitored and forensic investigation could benefit from a method to automatically detect signs of extreme emotions, fear, or aggression as an early warning system.

2 Related Work

Good surveys on the research efforts regarding facial expression recognition in image sequences can be found in Refs. 1 and 2. Facial expression recognition problems in image sequences can be divided into three subproblems. (1) Face detection: before a facial expression can be analyzed, the face must be detected in a scene. (2) Feature extraction and tracking: once the face is detected from the image sequence, the next step is to extract the information about the exhibited facial expression and track in subsequent frames. (3) Classification: finally we need to classify the extracted facial expression information into a particular facial action or basic emotion.

A summary of the surveyed methods for automatic facial feature extraction is presented in Table 1. Methods can be broadly categorized into two classes. (1) Template based methods: methods proposed by Black, Wang, Essa, and Kimura will come under this category. (2) Feature based methods: methods proposed by Cohn, Kotsia, Seyed, Yaser, and Cohen will come under this category.

In all surveyed methods, the face should be in frontal view, but in a method proposed by Black and Yacoob[5] head motions are allowed. In a method proposed by Wang et al.[6] and Cohn et al.[7] facial landmark points should be labeled by hand manually, while in a method proposed by Kotsia and Pitas,[8] the Candide wire frame model should be fitted manually on the first frame of the image sequence. All surveyed methods, except the method proposed by Essa,[9] assume that faces should be without hair and glasses.

A summary of the methods used for facial expression classification is given in Table 2. In the surveyed methods, Lajevardi and Lech,[10] and Kotsia and Pitas[8] make the use of facial image sequences from the Cohn-Kanade database for testing. Lajevardi achieved an accuracy of 68.9%, while Kotsia 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.[7] detects action units, and from the combination of action units they detect facial expressions.

Robin et al.[14] develop a dynamic facial expression recognition (DFER) framework based on discrete choice models. The DFER consists in modeling the choice of a person who has to label a video sequence representing a facial expression. They proposed five models. The first assumes that only the last frame of the video triggers the choice of the expression. The second model has two components, the first captures the perception of the facial expression within each frame in the sequence, while the second determines which frame triggers the choice. The third model is an extension of the second model and assumes that the choice of the expression results from the average of perceptions within a group of frames. The fourth and fifth models integrate the panel effect inherent to the estimation data and are

Table 1 Comparing feature extraction methods.

Reference	Method	Remarks/limitations
Black[5]	Local parametrized model of image motion, optical algorithm	Head motion and light variation allowed
Wang[6]	Labeled graph fitting	Front views, face without hair and glasses, manual labeling on first frame
Cohn[7]	Optical flow algorithm of Lucas-Kanade	Front views, face without hair and glasses, manual labeling on first frame
Kotsia[8]	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
Essa[9]	Optical flow method	Front views, face with hair and glasses, light variation allowed
Seyed[10]	Log Gabor filters with Gaussian transfer functions	Only front view faces without hair and glasses allowed
Yaser[11]	Statistical characterization of motion pattern in specified regions of face	Only front view faces without hair and glasses allowed
Kimura[12]	Potential net fitting to normalized face image by Gaussian filter	Only front view faces without hair and glasses allowed
Cohen[13]	Piecewise Bezier volume deformation (PBVD) tracker	Front views, face with hair and glasses, manual labeling on first frame

Table 2 Comparing classification methods.

Ref.	Method	Test sequences	Accuracy
Black[5]	Temporal consistency of the mid-level predicates which describes the motion of the facial features	70 sequences of 40 subjects	88%
Wang[6]	Averaged Bsplines of feature trajectories	29 sequences of 8 subjects	95%
Cohn[7]	Discriminant functions	504 sequences of 100 subjects	88%
Kotsia[8]	Multi class SVM	Sequences from the Cohn-Kanade database	99.7%
Essa[9]	Spatio temporal motion energy templates	22 sequences of 8 subjects	100%
Seyed[10]	Naive Bayesian classifier	172 sequences of 100 subjects from the Cohn-Kanade database	68.9%
Yaser[11]	Rule-based, prepared dictionary of rules	46 sequences of 32 subjects	
Kimura[12]	3D emotion space (PCA)		
Cohen[13]	Bayesian classifier	Cohn-Kanade database	74%

respectively ex tensing the first and second models. The models are estimated using videos from the Facial Expressions and Emotions Database.

Nan and Junmei[15] point out that one class of expression may be induced of various mental inducements, so it is more important to judge the inducements of a facial expression than to identify the class of a facial expression. Two local regions such as the mouth and eye that are extracted from the CED-WYU(1.0) database. They used the mouth images to analyze the expression inducements. The different expression in the different region has different characteristics. From the recognition rate of the mouth region, it can be seen the recognition rate of happiness is the highest. It may be effective to distinguish happiness from other expressions in the mouth region. The sadness, anger, neutral, and disgust have an expression inducement intersection to some degree in the mouth region.

Geetha et al.[16] propose a feature extraction method for an integrated face tracking and facial expression recognition in real-time video. The method proposed by Viola and Jones is used to detect the face region in the first frame of the video. A rectangular bounding box is fitted over for the face region and the detected face is tracked in the successive frames using the cascaded support vector machine (SVM) and cascaded radial basis function neural network (RBFNN). The haar-like features are extracted from the detected face region and they are used to create a cascaded SVM and RBFNN classifiers. Each stage of the SVM classifier and RBFNN classifier rejects the nonface regions and pass the face regions to the next stage in the cascade thereby efficiently tracking the face. While the face is being tracked, features are extracted from the mouth region for expression recognition. The features are modeled using a multiclass SVM. The SVM finds an optimal hyper plane to distinguish different facial expressions.

Zhao and Zhang[17] propose a new kernel-based manifold learning method, called kernel discriminant isometric mapping (KDIsomap). KDIsomap aims to nonlinearly extract the discriminant information by maximizing the interclass scatter while minimizing the intraclass scatter in a reproducing kernel Hilbert space. KDIsomap is used to perform nonlinear dimensionality reduction on the extracted local binary patterns facial features, and produce low-dimensional discriminant embedded data representations with striking performance improvement on facial expression recognition tasks. The nearest neighbor classifier with the Euclidean metric is used for facial expression classification.

Anderson et al.[18] use facial motion to characterize monochrome frontal views of facial expressions and is able to operate effectively in cluttered and dynamic scenes, recognizing the six emotions universally associated with unique facial expressions, namely happiness, sadness, disgust, surprise, fear, and anger. Faces are located using a spatial ratio template tracker algorithm. Optical flow of the face is subsequently determined using a real-time implementation of a robust gradient model. The expression recognition system then averages facial velocity information over identified regions of the face and cancels out rigid head motion by taking ratios of this averaged motion. The motion signatures produced are then classified using SVM as either nonexpressive or as one of the six basic emotions.

2.1 *Proposed Work and Contribution*

Humans display their emotions through facial expressions. Humans detect and interpret facial expressions in a scene with little or no effort. But the development of an automated system that perform this task is difficult. There are several related problems: detection of image segment as a face, extraction of facial expression information, and classification of the expression. We are trying to develop a system that performs these operations in real-time. In the literature review we have seen that, 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. In our approach, manual fitting of a model on a neutral frame is not required and our approach is applicable for frontal as well as tilted faces. We did the hardware implementation

of our algorithm in order to get a real-time system. 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 of 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 sources of variation from the actual information we are looking for. We use the active appearance model (AAM),[19] which enables us to automatically create a model of a face in an image. The created models are realistic looking faces. Thus the variation in light, resolutions, pose, and orientation will have no effect on expression recognition. We propose a method which makes use of the Candide wire frame model. The Viola-Jones algorithm[20] is used for face detection in an image sequence, whereas facial feature tracking is done using the active face model proposed by Ahlberg.[21] The active face model is designed using AAM. Expression classification is performed by multiclass SVM using the one against all approach. Among various available methods of SVM, we got maximum accuracy for the one against all approach and it needs only six classifiers, whereas the one against one approach needs 15 classifiers. Let us consider an image sequence containing face. The first frame of the sequence belongs to a neutral facial expression and the last frame corresponds to a fully expressed state. Our approach uses automatic fitting of a wire frame model on the first frame of the image sequence as against the manual fitting used in Ref. 8. The tracking system allows the grid to follow the evolution of the facial expression over time in an image sequence, until it reaches its highest intensity, producing the deformed Candide grid at each video frame. The geometrical displacement of the 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 multiclass SVM classifier, which classifies facial expression into one of the classes such as happy, anger, sadness, surprise, disgust, and fear. No texture information is required, only geometrical information is fed to the SVM classifier. Our framework is different from the one proposed in Ref. 8. The uthors in Ref. 8 use the pyramidal Kanade Lucas Tomasi tracker,[22] based on optical flow computation, whereas we make use of an active face model proposed in Ref. 21 for tracking, based on an active appearance algorithm. The Kanade Lucas Tomasi tracker is used only for grayscale images and frontal faces, whereas the active appearance algorithm can be used for color images and nonfrontal faces also. The active appearance algorithm is mainly used for tracking by researchers, but in our approach along with SVM we use it for facial expression recognition scheme.

The rest of the paper is organized as follows. The system used for facial expression recognition is described in Sec. 3. Results are given in Sec. 4. We conclude and discuss limitations in Sec. 5.

3 Facial Expression Recognition System

The proposed framework is composed of three subsystems. The first is used for face detection. The second is for Candide grid node coordinate displacement extraction and the third is used for grid node displacement classification. Face detection is performed by the Viola Jones algorithm. The Candide wire frame model is fitted on the first frame of the facial image sequence. The Viola-Jones algorithm gives face location from which scaling and translation parameters of the model are determined. That fits the Candide model on the face approximately. Correct fitting will be obtained by running the active appearance algorithm. The grid node information extraction is performed by the active face model tracking system, while the grid node information classification is performed by a six class SVM system. Six class SVM classifies facial expression into one of the six basic classes happy, surprise, sadness, anger, disgust, and fear. The flow diagram of the proposed framework is shown in Fig. 1.

3.1 *Face Detection*

In the 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 it seemed 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, the face area of an image was detected using the Viola-Jones method[20] based on the Haar-like features and AdaBoost learning algorithm. The Viola-Jones method is an object detection algorithm providing competitive object detection rates in real-time. It was primarily designed for the problem of face detection. The features used by Viola and Jones are derived from pixels selected from rectangular areas imposed over the picture and show high sensitivity to the vertical and horizontal lines. AdaBoost is an adaptive learning algorithm that can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built iteratively are made to fix instances misclassified by previous classifiers. At each iteration a distribution of weights is updated such that, the weights of each incorrectly classified example are increased (or alternatively, the weights of each correctly classified example are decreased), so that the new classifier focuses more on those examples. The result of the face detection algorithm is shown in Fig. 2.

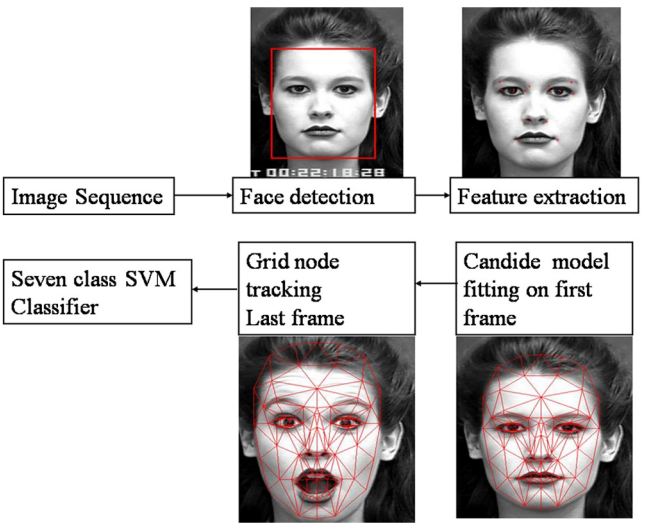

Fig. 1 Flow diagram for the facial expression recognition system.

Fig. 2 Face detection.

3.2 *Extraction of Candide Grid Node Coordinates*

3.2.1 *Candide wire frame model*

We have used the 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 at the British Telecom created another version with 160 vertices and 238 triangles covering the entire front 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.[23] The 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 Fig. 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 Ref. 23 can be expressed as in Eq. (1):

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

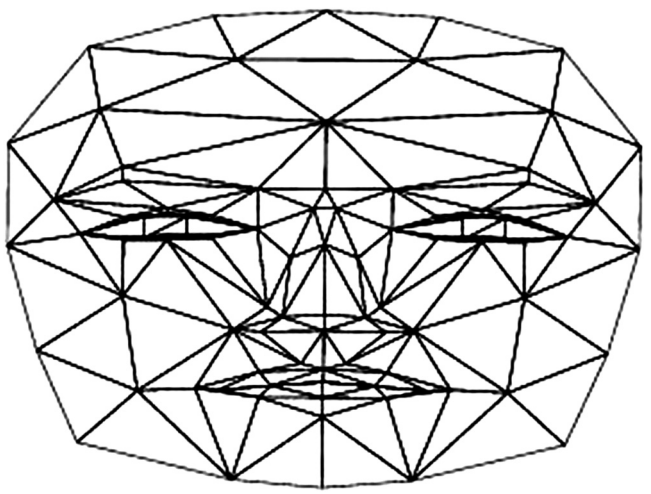

Fig. 3 Candide wire frame model.

Here the resulting vector V contains (x, y, z) coordinates of vertices of the model. $\bar{V}$ is a vector containing vertex coordinates of the 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 a shape parameter. A_i represents an animation unit. There are 65 animation units, such as lips stretched, nose wrinkled, inner brow raised, outer brow raised, etc., whereas α_i is an 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 the global motion of the model, six more parameters, three for rotation, one for scaling, and two for translation, are added to formula in Eq. (1):

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

Here $R = (\theta_x, \theta_y, \theta_z)$ is a 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 Eq. (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,[21] as in Eq. (4):

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

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 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 choose to work with 33×40 pixel images which are conveniently small and effective for image warping. Barycentric coordinate computation and texture mapping is used to get a geometrically normalized image.

3.2.2 *Texture synthesis*

We performed 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 the variance. A face vector x can be parametrized as in Eq. (5), where $\bar{x}$ is the mean face, and we could synthesize a face image as in Eq. (6):

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

and

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

3.2.3 *Tracking*

Facial tracking means to find an optimal adaptation of the model to frame in the 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 that 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. Now we map the input image (i) on the model, and reshape the model to the standard shape $\bar{V}$ and get the resulting normalized image as a vector as in Eq. (7) and then we compute texture parameters from normalized image as in Eq. (8). Synthesized texture would be given as Eq. (9).

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

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

and

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

The residual image is calculated as in Eq. (10) and the summed square error is selected as the error measure and is given as in Eq. (11).

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

and

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

For good model adaptation, the residual image and error e is much smaller. Now we find the update vector $\triangle p$ by multiplying the residual image with an update matrix U. The new error measure for the updated parameter is as in Eq. (13).

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

and

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

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

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

for $k = 1, 2, 3, \ldots$. If $e_k < e$, then we update $e_k \rightarrow e$ and $p + \frac{1}{2^k}\Delta p \rightarrow p$ Iterate the scheme and declare the convergence when $e_k > e$. The model matching and texture approximation process is shown in Fig. 4. A correct model adaptation is shown in the top row, and a wrong adaptation is shown in the bottom row. The first image in both the rows shows a model adapted on a face image. The second image in both the rows is a texture of a face image mapped on a geometrically normalized Candide wire frame model. The normalized texture is approximated by the eigentextures producing the synthesized image. The residual image is computed by subtracting the normalized image and synthesized image. From the figure it is clear that the normalized image and the synthesized image are more similar for better model adaptation. Analysis of the residual image tells us how to improve model adaptation.

3.2.4 *Creating update matrix*

Assuming that $r(i, p)$ is linear in p, that is

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

G is a gradient matrix. Taylor expanding $r(i, p)$ around $p + \Delta p$ we get

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

Term O represents higher order derivatives of $r(i, p)$. We want to find Δp that minimizes

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

Minimizing the above equation is a least squares problem with the solution

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

which gives an update matrix U as the negative pseudo inverse of the gradient matrix G

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

The gradient matrix G is calculated from the training images in advance. The j'th column in G is given by

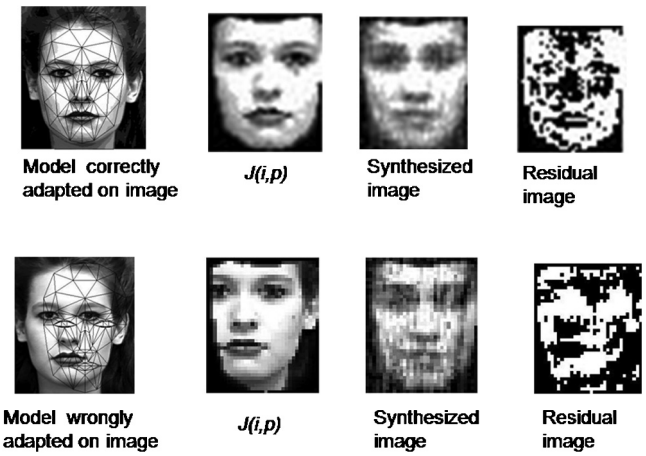

Fig. 4 Candide wire frame model fitting on face image.

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

The approximation could be as

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

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. 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

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

where N is the number of training images and K is the number of steps to perturb the parameter.[24]

3.3 *Model Fitting on the First Frame*

To fit the Candide model on the first frame of the image sequence, we consider scaling, rotation, and translation parameters. For initial fitting, a rough estimate of the scaling and translation parameters is very essential. Our face detection algorithm gives a rectangle enclosing a face as shown in Fig. 2. The top left point of this rectangle has coordinates (x, y). The width of the rectangle is W_d. The height of the rectangle is H_t. From this data we get a rough estimate of translation parameters (t_x, t_y)

$$t_x = x + W_d/2 \tag{23}$$

and

$$t_y = y + H_t/2. \tag{24}$$

The Candide model is fitted manually on 100 images. From the manual fitting we have selected a rough estimate of the scaling parameter as $s = 0.78 * W_d$. With a rough estimation of the scaling and translation parameters, the Candide model roughly fits on the face image. Initially, rotation is assumed to be zero. Once the model roughly fits on the face image, exact fitting is obtained by the active appearance algorithm. A separate update matrix is constructed only for the scaling, translation, and rotation parameters. The update matrix is constructed using Eq. (19), where $N = 40$ and $K = 20$. For scaling, $h = 100 \times 0.01$ in the range $[-hk, hk] = [-20, 20]$. For translation, h is relative to the size of the model $h = 0.01 \times (s + 1)$ in the range $[-hk, hk] = [-0.2 \times (s + 1), 0.2 \times (s + 1)]$ For rotation, h has to be converted into radians $h = 0.01 \times 180/\pi$ in the range $[-hk, hk] = [-11.459, 11.459]$. The Candide model is manually fitted on 40 different face images. All parameters are varied according to the above steps, and residual images are collected. The gradient matrix is computed using Eq. (22). From the gradient matrix, the update matrix is computed using Eq. (19). For an unknown image, when the model fits roughly, the residual image is computed. The residual image gets multiplied by an update matrix to get an updated parameter vector $p = [s, t_x, t_y, \theta_x, \theta_y, \theta_z]$. With this new parameter, the model gets deformed, again the residual image is computed, and the process repeats until the model fits exactly, where the error e is reduced to a minimum. Once the model fits properly on the first frame, in the subsequent frames only animation parameters need to be processed.

3.4 *Classification Using SVM*

The classification is performed only on the basis of geometrical information, not taking into consideration any luminance or color information. The geometric 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 multiclass SVM 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, sadness, anger, fear, and disgust. The 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. Basically SVM is a binary classifier, which classifies data in two classes.[25,26] We are extending the approach for six classes.

In this scheme, there is one binary SVM for each class to separate members of that class from members of other classes. We have a number of classifiers equal to a number of classes. Classifier i, j is trained using all patterns from class i as positive instances, and all patterns from the rest of the class are assumed to be in class j as negative instances. The class for which the decision function gives a maximum value will be declared as class of the new instance. We have a video database that contains the facial image sequences. This database is clustered into six different classes 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

$$d_{ij} = [\,\Delta x_{ij} \quad \Delta y_{ij}\,]^T, \tag{25}$$

where $i = 1, \ldots, E$ and $j = 1, \ldots, N$, and Δx_{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. The vector g_j is called grid deformation feature vector, which contains the geometrical displacement of every grid node

$$g_j = [d_{1,j}, d_{2,j}, \ldots, d_{E,j}]^T, \quad (26)$$

where $j = 1, \ldots, N$. The dimension of the vector g_j is $D = 111 \times 2 = 226$ 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 the 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 as

$$\min_{w,b,\xi} \sum_{k=1}^{6} W_k^T W_k + C \sum_{j=1}^{N} \sum_{k \neq l_j} \xi_j^k, \quad (27)$$

with the constraints $W_{l_j}^T \phi(g_j) + b_{l_j} \geq W_k^T \phi(g_j) + b_k + 2 - \xi_j^k \; \xi_j^k \geq 0$, $j = 1, \ldots, N$, and $k\{1, \ldots, 6\}$. 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. $b = [b_1, \ldots, b_6]$ is the bias vector, and $\xi = [\xi_1^1, \ldots, \xi_i^k, \ldots, \xi_N^6]$ is the slack variable vector. The decision function is given as

$$h(g) = \arg \max_{k=1,\ldots,6} (W_k^T \phi(g) + b_k). \quad (28)$$

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

4 Experimental Results

We have used the Cohn-Kanade database,[27] for constructing the update matrix as well as for SVM training. The Viola-Jones algorithm is successfully used for face detection in a scene. The Candide wire frame model fits on the first frame and tracked in the subsequent frame using the active appearance algorithm which is implemented in MATLAB. For texture synthesis, 40 images of different persons with different expressions are considered as training images.

Fig. 5 Model fitting and tracking results.

Table 3 Confusion matrices and accuracy with 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%

The Candide model is manually adapted to these images. These images are then geometrically normalized (33×4 pixel) to a standard shape, and then PCA is applied on them. For constructing the update matrix, we have selected seven animation units. These are the upper lip raised, jaw dropped, lip stretched, eyebrow lowered, eyebrow raised, lip corner depressed, and nose wrinkled. The Candide model has adapted manually 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. The number of steps we have selected is $K = 20$, and number of images we have selected is $N = 40$. Then

Table 4 Confusion matrices and accuracy with six classes.

Expression	Happy	Surprise	Sad	Anger	Disgust	Fear
Happy	76%	0	0	0	0	0
Surprise	8%	94%	0	0	0	0
Sad	0	0	77%	8%	38%	15%
Anger	8%	0	23%	84%	0	0
Disgust	0	6%	0	8%	62%	15%
Fear	8%	0	0	0	0	70%

Table 5 Confusion matrices and accuracy with six classes obtained by Kotsia.[8]

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 6 Confusion matrices and accuracy with seven classes obtained by Cohen.[13]

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

the update matrix is computed. On each frame, the update matrix is multiplied with the residual image to get an updated parameter vector. The difference between the vertex coordinates of the first and last frame is used as input to the SVM. Six class SVM classification is implemented in MATLAB. SVM is trained using image sequences from the Cohn-Kanade database. From the database, 25 image sequences of each class are selected for training. After training, the same image sequences are used for testing. In testing, 15 image sequences are not correctly classified. These are removed from the training database. So only 135 image sequences are used in training SVM. Confusion matrices and accuracy for four classes is shown in Table 3. 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 the facial expressions that are correctly classified, while the off diagonal entries correspond to misclassification rates. When only four expressions (happy, sad, anger, and surprise) are considered, the overall accuracy is 85%. But when two expressions (fear and disgust) are added, the accuracy decreases to 79.4%. Out of 73 samples it identifies 58 samples correctly. Confusion matrices and accuracy for six classes is shown in Table 4. Fear makes confusion with disgust and sadness, while disgust makes confusion with sad. Accuracy depends on the initial fitting of model on the first frame. In Ref. 8, Kotsia and Pitas obtained 99.7% accuracy as shown in Table 5, but the initial fitting of the model is done manually. In our case, it is done automatically. Confusion matrices and accuracy obtained by Cohen et al.[13] is given in Table 6. They got average accuracy 73% for the Cohn-Kanade database. They have considered a neutral expression as one of the facial expressions. So total they have seven facial expressions. Our work is going on to improve classification accuracy for a large number of expressions. Accuracy of our system depends on accuracy of the face detection algorithm, model fitting, tracking, as well as on the SVM classifier. The model fits almost on all the face images. Model fitting accuracy is 99%. In tracking, we got 88% accuracy. The SVM classifier works with 92% accuracy. The overall accuracy of our system is 79.4%. We are trying to improve tracking accuracy by including a variety of face images while constructing AAM. Some of the results of correct model fitting and tracking are shown in Fig. 5.

5 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 the active shape model, and active appearance algorithm. A face is detected from a scene in a first frame of the image sequence and the Candide wire frame model is automatically fitted on it. As the facial expression changes in subsequent frames, the model deforms its shape. When the last frame is reached, the model is fully deformed. The difference between Candide grid node coordinates of the first and last frame is given as input to a six class SVM system. Only geometrical information is given to the SVM, no texture information is given to the SVM. The system is fully automatic. Manual fitting of the Candide wire frame model on the first frame of the image sequence is not required.

Presently, the system recognizes only six 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 a larger number of expressions (with inspiration from Indian classical performing art). The proposed algorithm is applied 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 the 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. B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognit.* **36**(1), 259–275 (2003).

2. M. Pantic and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(12), 1424–1445 (2000).
3. P. Ekman and W. V. Friesen, *Facial Action Coding System Manual*, Consulting Psychologists Press, Palo Alto (1978).
4. C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: a comprehensive study," *Image Vision Comput.* **27**, 803–816 (2009).
5. M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *Int. J. Comput. Vision* **25**(1), 23–48 (1997).
6. M. Wang, Y. Iwai, and M. Yachindai, "Expression recognition from timesequential facial images by use of expression change model," in *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pp. 324–329, IEEE Computer Society, Japan (1998).
7. J. F. Cohn et al., "Feature point tracking by optical flow discriminates subtle differences in facial expression," in *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pp. 396–401, IEEE Computer Society, Japan (1998).
8. I. Kotsia and I. Pitas, "Facial expression recognition in image sequences using geometric deformation features and support vector machines," *IEEE Trans. Image Process.* **16**(1), 172–187 (2007).
9. I. A. Essa and A. P. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7), 757–763 (1997).
10. S. Lajevardi and M. Lech, "Facial expression recognition from image sequences using optimized feature selection," in *23rd Int. Conf. on Image and Vision Computing New Zealand*, pp. 1–6, IEEE Computer Society, New Zeland (2008).
11. Y. Yacoob and L. Davis, "Recognizing human facial expressions from long image sequences using optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(6), 636–642 (1996).
12. S. Kimura and M. Yachida, "Facial expression recognition and its degree estimation," in *Proc. Computer Vision and Pattern Recognition*, pp. 295–300, IEEE Computer Society, San Jaun Puerto Rica (1997).
13. I. Cohen et al., "Facial expression recognition from video sequences: temporal and static modeling," *Comput. Vis. Image Und.* **91**(1–2), 160–187 (2003).
14. T. Robin, M. Bierlaire, and J. Cruz, "Dynamic facial expression recognition with a discrete choice model," *J. Choice Model.* **2**(1), 95–148 (2010).
15. Z. Nan and G. Junmei, "The inducement analysis of local facial expression recognition," in *Int. Conf. on System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, pp. 306–309, IEEE, Guiyang China (2011).
16. A. Geetha, V. Ramalingam, and S. Palanivel, "An integrated face tracking and facial expression recognition system," *J. Intell. Learning Syst. Appl.* **3**, 201–208 (2011).
17. X. Zhao and S. Zhang, "Facial expression recognition based on local binary patterns and Kernel discriminant Isomap," *Sensors* **11**(10), 9573–9588 (2011).
18. K. Anderson and P. McOwan, "A real-time automated system for the recognition of human facial expressions," *IEEE Trans. Syst. Man Cybern. B, Cybern.* **36**(1), 96–105 (2006).
19. T. F. Cootes, G. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. 5th European Conf. on Computer Vision*, pp. 484–498, Springer, Verlag, Feriburg Germany (1998).
20. P. Viola and M. Jones, "Robust real-time object detection," Cambridge Research Laboratory Technical Report Series (2001).
21. J. Ahlberg, "Fast image warping for active models," Tech. Rep. Report No. LiTH-ISY-R-2355, Linkoping University, Dept. of EE (2001).
22. J. Y. Bouguet, "Pyramidal implementation of the Lucas-Kanade feature tracker," Tech. Rep., Intel Corporation, Microprocessor Research Labs (1999).
23. J. Ahlberg, "Candide-3 an updated parameterized face," Tech. Rep. Report No. LiTH-ISY-R-2326, Linkoping University, Dept. of EE (2001).
24. J. Ahlberg, "An active model for facial feature tracking," *EURASIP J. Appl. Signal Process.* **2002**(6), 566–571 (2002).
25. J. Weston and C. Watkins, "Multi-class support vector machines," Tech. Rep. CSD-TR-98-04 (2004).
26. C. W. Hsu, C. C. Chang, and C. J. Lin, *A Practical Guide to Support Vector Classification*, National Taiwan University, Taipei, Taiwan (2002).
27. T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. IEEE Int. Conf. Face and Gesture Recognition*, pp. 46–53, IEEE Computer Society, Pittsburgh (2000).

Biographies and photographs of the authors are not available.