基于局部 SVM分类器的表情识别方法

孙正兴,徐文晖
(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

摘 要:提出了一种新的视频人脸表情识别方法. 该方法将识别过程分成人脸表情特征提取和分类 2个部分,首先采用基于点跟踪的活动形状模型 (ASM)从视频人脸中提取人脸表情几何特征;然后,采用一种新的局部支撑向量机分类器对表情进行分类. 在 Cohn-Kanade数据库上对 KNN、SVM、KNN-SVM和 LSVM 4种分类器的比较实验结果验证了所提出方法的有效性.

关键字:人脸表情识别;局部支撑向量机;活动形状模型;几何特征
中图分类号: TP391 **文献标识码**: A **文章编号**: 1673-4785(2008)05-0455-12

Facial expression recognition based on local SVM classifiers

SUN Zheng-xing, XU Wen-hui
(State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract: This paper presents a novel technique developed for the identification of facial expressions in video sources The method uses two steps: facial expression feature extraction and expression classification First we used an active shape model (ASM) based on a facial point tracking system to extract the geometric features of facial expressions in videos Then a new type of local support vector machine (LSVM) was created to classify the facial expressions Four different classifiers using KNN, SVM, KNN-SVM, and LSVM were compared with the new LSVM. The results on the Cohn-Kanade database showed the effectiveness of our method

Keywords: facial expression recognition; local SVM; active shape model; geometry feature

Automatic facial expression recognition has attracted a lot of attention in recent years due to its potentially vital role in applications, particularly those using human centered interfaces Many applications, such as virtual reality, video-conferencing, user profiling, and customer satisfaction studies for broadcast and web services, require efficient facial expression recognition in order to achieve their desired results Therefore, the impact of facial expression recognition on the above-mentioned applications is constantly growing

Several approaches have been reported in the literature to automate the recognition of facial expressions in mug shots or video sequences Early methods used mug shots of expressions that captured characteristic images at the apex[1-2]. However, according to psychologists[3], analysis of video sequences produces more accurate and robust recognition of facial expressions These methods can be categorized based on the data and features they use, as well as the classifiers created for expression recognition In summary, the classifiers include Nearest Neighbor classifier[4], Neural Networks[5], SVM[6], Bayesian Networks[7], AdaBoost classifier[6] and hidden Markov model[8]. The data used for automated facial expression analysis (AFEA) can be geometric features or texture features, for each there are different feature extraction methods Though facial expression recognition has made remark-

收稿日期: 2008-07-11.
基金项目: National High Technology Research and Development Program (863) of China (2007AA01Z334); National Natural Science Foundation of China (69903006, 60373065, 0721002).
通信作者:孙正兴. E-mail: szx@nju.edu.cn

able progress, recognizing facial expressions with high accuracy is a difficult problem[9]. AFEA and its effective use in computing presents a number of difficult challenges In general, two main processes can be distinguished in tackling the problem: 1) Identification of features that contain useful information and reduction of the dimensions of feature vectors in order to design better classifiers 2) Design and implementation of robust classifiers that can learn the underlying models of facial expressions

We propose a new classifier for facial expression recognition, which comes from the ideas used in the KNN-SVM algorithm. Ref [10] proposed this algorithm for visual object recognition This method combines SVM and KNN classifiers and implements accurate local classification by using KNN for selecting relevant training data for the SVM. In order to classify a sample x, it first selects k training samples nearest to the sample x, and then uses these k samples to train an SVM model which is then used to make decisions KNN-SVM builds a maximal margin classifier in the neighborhood of a test sample using the feature space induced by the SVM's kernel function. But this classifier discards nearest-neighbor searches from the SVM learning algorithm. Once the K-nearest neighbors have been identified, the SVM algorithm completely ignores their similarities to the given test example. So we present a new classifier based on KNN-SVM, called local SVM (LSVM), which incorporates neighborhood information into SVM learning The principle behind LSVM is that it reduces the impact of support vectors located far away from a given test example.

In this paper, a system for automatically recognizing the six universal facial expressions (anger, disgust, fear, joy, sadness, and surprise) in video sequences using geometrical feature and a novel class of SVM called LSVM is proposed The system detects frontal faces in video sequences and then geometrical features of some key facial points are extracted using active shape model (ASM) based tracking In each video sequence, the first frame shows a neutral expression while the last frame shows an expression with maximum intensity. For each frame, we extract geometric features as a static feature vector, which represents facial contour information during changes of expression. At the end, by subtracting the static features of the first frame from those of the last, we get dynamic geometric information for classifier input Then an LSVM classifier is used for classification into the six basic expression types

The rest of the paper is organized as follows Section 2 reviews facial expression recognition studies In Section 3 we briefly describe our facial point tracking system and the features extracted for classification of facial expressions Section 4 describes the Local SVM classifier used for classifying the six basic facial expressions in the video sequences Experiments, performance evaluations, and discussions are given in section 5. Finally, section 6 gives conclusions about our work.

1 Related work

Psychological studies have suggested that facial motion is fundamental to the recognition of facial expression. Experiments conducted by Bassili[11] demonstrated that humans do a better job recognizing expressions from dynamic images as opposed to mug shots Facial expressions are usually described in two ways: as combinations of action units, or as universal expressions The facial action coding system (FACS) was developed to describe facial expressions using a combination of action units (AU)[12]. Each action unit corresponds to specific muscular activity that produces momentary changes in facial appearance. Universal expressions are studied as a complete representation of a specific type of internal emotion, without breaking up expressions into muscular units Most commonly studied universal expressions include happiness, anger, sadness, fear, and disgust In this study, universal expressions were analyzed using the facial expression coding system.

Many automated facial expression analysis methods have been developed[13]. Mase[14] used optical flow (OF) to recognize facial expressions He was one of the first to use image-processing techniques to recognize facial expressions Black and Yacoob[15] used local parameterized models of image motion to recover non-rigid motion. Once recovered, these parameters were used as inputs to a rule-based classifier to recognize the six basic facial expressions Ref [16] used lower face tracking to extract mouth shape features and used them as inputs to an HMM based facial expression recognition system (recognizing neutral, happy, sad, and an open mouth). Bartlett[17] automatically detects frontal faces in the video stream and classifies them in seven classes in real time: neutral, anger, disgust, fear, joy, sadness, and surprise. An expression recognizer receives image regions produced by a face detector and then a Gabor representation of the facial image region is formed to be later processed by a bank of SVM classifiers Facial feature detection and tracking is based on active InfraRed illumination in Ref [18], in order to provide visual information under variable lighting and head motion. The classification is performed using a dynamic Bayesian network (DBN). COHEN et al[18] proposed a method for static and dynamic segmentation and classification of facial expressions For the static case, a DBN is used, organized in a tree structure. For the dynamic approach, a multi-level hidden Markov models (HMMs) classifier is employed.

These methods are similar in that they first extract some features from the images, then these features are used as inputs into a classification system, and the outcome is one of the pre-selected emotion categories They differ mainly in the features extracted from the video images and in the classifiers used to distinguish between the different emotions In the following sections, an automatic geometric feature based method is proposed, and then LSVM classifiers are used for recognizing facial expressions from video sequences

2　Geometrical feature extraction

Our work focused on the design of classifiers for improving recognition accuracy, following the extraction of geometric features using a model-based face tracking system. That is, the proposed process for facial expression recognition is composed of two steps: one ASM based geometric information extraction; the next LSVM based classification. Geometric feature information extraction is performed by ASM based automatic locating and tracking, while the classification of geometric information is performed by an LSVM Classifier. Fig 1 shows the proposed facial expression recognition scheme.

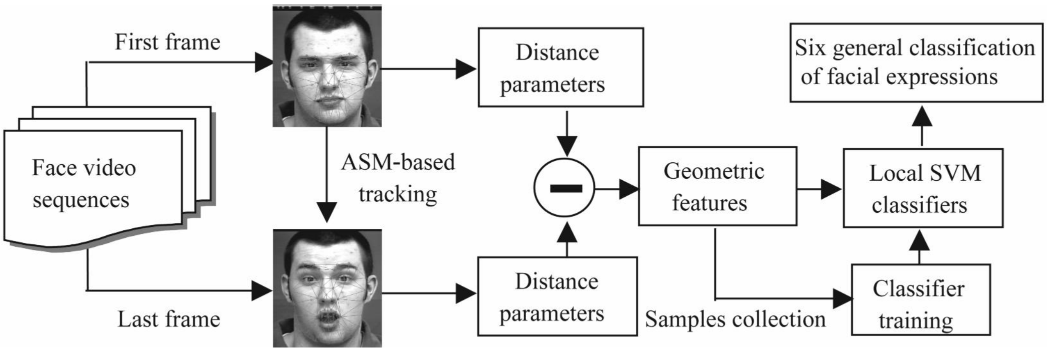

Fig 1　Process of facial expression recognition for video sequences

For each input video sequence, an AdaBoost based face detector is applied to detect frontal and near-frontal faces in the first frame Inside detected faces, our method identifies some important facial landmarks using the active shape model (ASM). ASM automatically localizes the facial feature points in the first frame and then trackes the feature points through the video frames as the facial expression evolves through time The first frame shows a neutral expression while the last frame shows an expression with the greatest intensity For each frame, we extract distance parameters between some key facial points At the end, by subtracting distance parameters from the first frame from those of the last frame, we get the geometric features for classification Then a LSVM classifier is used for classification into the six basic expression types

2 1 ASM based locating and tracking

ASM[19] is employed to extract shape information on specific faces in each frame of the video sequence The use of a face detection algorithm as a prior step has the advantage of speeding up the search for the shape parameters during ASM based processing ASM is built from sets of prominent points known as landmarks, computing a point distribution model (PDM) and a local image intensity model around each of those points The PDM is constructed by applying PCA to an aligned set of shapes, each represented by landmarks The original shapes and their model representation b_i ($i=1, 2, \cdots, N$) are related by means of the mean shape $\overline{u}$ and the eigenvector matrix Φ:

$$b_i = \Phi^{\mathrm{T}}(u_i - \overline{u}),\ u_i = \overline{u} + \Phi b_i. \qquad (1)$$

To reduce the dimensions of the representation, it is possible to use only the eigenvectors corresponding to the largest eigenvalues Therefore, Equ (1) becomes an approximation, with an error depending on the magnitude of the excluded eigenvalues Furthermore, under Gaussian assumptions, each component of the b_i vectors is constrained to ensure that only valid shapes are represented, as follows:

$$|b_i^m| \le \beta\sqrt{\lambda_m},\ 1 \le i \le N,\ 1 \le m \le M. \qquad (2)$$

Where, β is a regulating parameter usually set between 1 and 3 according to the desired degree of flexibility in the shape model m is the number of retained eigenvectors, and λ_m is the eigenvalues of the covariance matrix The intensity model is constructed by computing the second order statistics of normalized image gradients, sampled at each side of the landmarks, perpendicular to the shape's contour, hereinafter referred to as the profile In other words, the profile is a fixed-size vector of values (in this case, pixel intensity values) sampled along the perpendicular to the contour such that the contour passes right through the middle of the perpendicular The matching procedure is an alternation of image driven landmark displacements and statistical shape constraining based on the PDM. It is usually performed in a multi-resolution fashion in order to enhance the capture range of the algorithm The landmark displacements are individually determined using the intensity model, by minimizing the Mahalanobis distance between the candidate gradient and the model's mean

To extract facial feature points in case of expression variation, we trained an active shape model from the JAFFE (Japanese female facial expression) database[19], which contains 219 images from 10 individual Japanese females For each subject there are six basic facial expressions (anger, disgust, fear, happiness, sadness, surprise) and a neutral face 68 landmarks are used to define the face shape, as shown in Fig 2

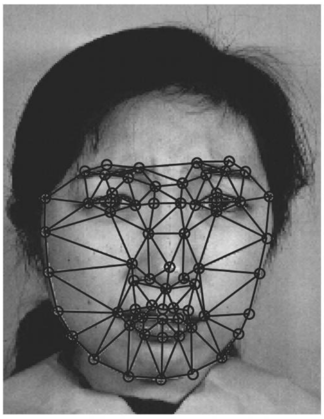

Fig 2 ASM training sample

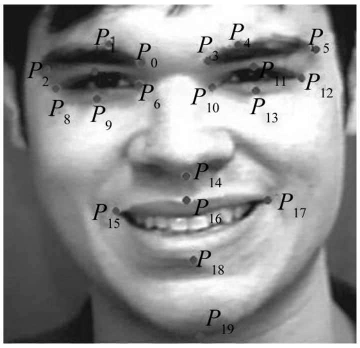

Fig 3　Facial characteristic points

2.2　Facial characteristic points model

The shape information extracted by ASM from a face image is used to compute a set of distance parameters that describe the appearance of facial features ASM extracts 68 facial points, however some of these don't reflect changes in facial expressions The first step is the selection of the 20 optimal key facial points, those which change the most with changes in expression. These key points P are defined as the facial characteristic points (FCPs, Fig 3), which were derived from the Kobayashi & Hara model[2]. In the second step the FCPs are converted into some distance parameters This parameterization has the advantage of providing the classifier with data that encode the most important aspects of the facial expressions The distance parameters are computed as the implicit fixed Euclidean distances between key points The complete list of such distance parameters is given in Table 1. In Table 1, $(P_i, P_j)_x$ represents the horizon distance between points P_i and P_j, $(P_i, P_j)_y$ represents the vertical distance between points P_i and P_j. Because when facial expressions change, most movement is in the vertical direction, most of the distance parameters compute vertical distance. We extracted the differences between the last and the first frame's distance parameters as the geometric features The geometric features capture the subtle changes in facial expression which varied over the video sequence. Let V_{end} be the distance parameter of the last frame, V_{begin} be the distance parameter of the first frame,

$$x_i = V_{\text{end}} - V_{\text{begin}}, i \in \{1, 2, \cdots, N\}. \qquad (3)$$

Where x_i is the geometric feature of the i-th video sequence, which is defined as the difference between static features of the first frame and the last frame. The dimension of the geometric feature x_i is 18.

Table 1　The set of distance parameters

v_i	meaning	Visual feature	v_i	meaning	Visual feature	v_i	meaning	Visual feature
v_1	$(P_0, P_1)_y$	Left eyebrow	v_7	$(P_7, P9)_y$	Left eye	v_{13}	$(P_{14}, P_{16})_y$	Mouse
v_2	$(P_0, P_2)_y$	Left eyebrow	v_8	$(P_6, P_8)_y$	Left eye	v_{14}	$(P_{15}, P_{18})_y$	Mouse
v_3	$(P_3, P_4)_y$	Right eyebrow	v_9	$(P_6, P_9)_y$	Left eye	v_{15}	$(P_{14}, P_{15})_y$	Mouse
v_4	$(P_3, P_5)_y$	Right eyebrow	v_{10}	$(P_{11}, P_{13})_y$	Right eye	v_{16}	$(P_{14}, P_{17})_y$	Mouse
v_5	$(P_0, P_{14})_y$	Left eyebrow	v_{11}	$(P_{10}, P_{12})_y$	Right eye	v_{17}	$(P_{15}, P_{17})_x$	Mouse
v_6	$(P_3, P_{14})_y$	Right eyebrow	v_{12}	$(P_{10}, P_{13})_y$	Right eye	v_{18}	$(P_{14}, P_{19})_y$	Chin

3　Facial expression recognition based on local SVM

Effective facial expression recognition is a key problem in automated facial expression analysis The KNN[20] and SVM[21] classifiers have been successfully applied to facial expression recognition and improve facial expression recognition accuracy. We propose a further improvement, an LSVM classifier for facial expression recognition, with its roots in the KNN-SVM[10] classifier, but KNN-SVM decouples the nearest-neighbor search from the SVM learning algorithm. Once the K-nearest neighbors have been identified, the SVM algorithm completely ignores their similarities to the given test example. So we incorporated neighborhood information into SVM learning to improve the classification accuracy of KNN-SVM.

3. 1 Nearest neighbors and SVM

In this part we will give a brief description of nearest neighbors and SVM classifiers. Lets assume a classification problem with samples $D = \{ (x_i, y_i) \}$ with $i = 1, 2, \cdots, N$, $x_i \in R^d$ and $y_i \in \{1, -1\}$. For the K-nearest neighbor (KNN) algorithm, given a point x' in the n-dimensional feature space, an ordering function $f_{x'}: R^d \rightarrow R$ is defined. A typical ordering function is based on Euclidean metrics:

$$f_{x'}(x) = \| x - x' \|.$$

By means of an ordering function, it is possible to order the entire set of training samples x with respect to x'. This is equivalent to defining a function $r_{x'}: \{1, \cdots, N\} \rightarrow \{1, \cdots, N\}$ that maps the indexes of the N training points of the datasets. We define this function recursively.

$$\begin{cases} r_{x'}(1) = \underset{i=1,\cdots,N}{\arg\min} \| \Phi(x_i) - \Phi(x') \|^2, \\ r_{x'}(j) = \underset{i=1,\cdots,N}{\arg\min} \| \Phi(x_i) - \Phi(x') \|^2, \\ i \neq r_{x'}(1), \cdots, r_{x'}(j-1), j = 2, \cdots, N. \end{cases} \qquad (4)$$

In this way, $x_{r_{x'}(j)}$ represents the j-th point in the set $D = \{ (x_i, y_i) \}$ in terms of distance from x', namely the j-th nearest neighbor of x', with $f_{x'}(x_{r_{x'}(j)}) = \| x_{r_{x'}(j)} - x' \|$ being its distance from x' and $y_{r_{x'}(j)}$ is its classification. Given the above definition, the decision rule of the KNN classifier for binary classification is defined by

$$\mathrm{KNN}(x) = \mathrm{sign}\left(\sum\nolimits_{i=1}^{k} y_{r_{x'}(i)}\right). \qquad (5)$$

Support vector machines (SVMs) are based on statistical learning theory[22]. IIn the classification context, the decision rule of an SVM is generally given by $\mathrm{SVM}(x) = \mathrm{sign}(w \cdot \Phi(x) + b)$, where, $\Phi(x): R^d \rightarrow F$ is a mapping in some transformed feature space F. $w \in F$ and $b \in R$ are parameters such that they minimize an upper bound on the expected risk while minimizing the empirical risk. Such a bound is composed of an empirical risk term and a complexity term that depends on the VC dimension of the linear separator. Controlling or minimizing both terms permits control of the generalization error in a theoretically well-founded way. The learning procedure of an SVM can be summarized as follows. The minimization of the complexity term is achieved by minimizing the quantity $\frac{1}{2}\| w^2 \|$, namely maximizing the class separation margin. The empirical risk term is controlled through the following constraint:

$$y_i (w \cdot \Phi(x_i) + b) \geq 1 - \xi_i. \qquad (6)$$

Where, $\xi_i \in \{ i = 1, 2, \cdots, N \} \geq 0$. The presence of the slack variables ξ_i allows some misclassification in the training set. In fact, during model building, a nonlinear SVM is trained to solve the following optimization problem:

$$\begin{aligned} &\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_i, x_j), \\ &\mathrm{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq c, i = 1, 2, \cdots, n. \end{aligned} \qquad (7)$$

By reformulating such an optimization problem with Lagrange multipliers α_i $(i = 1, \cdots, N)$, it is possible to write the following decision rule:

$$\mathrm{SVM}(x) = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i \cdot y_i \cdot \Phi(x_i) \cdot \Phi(x) + b\right). \qquad (8)$$

Where, the mapping Φ appears only in the dot products $\Phi(x_i) \cdot \Phi(x)$. This is an important property, which allows kernelizing of the classification problem. Indeed, if a kernel function $k(\cdot, \cdot)$ satisfies Mercer's theorem, it is possible to substitute $k(x_i, x)$ with $\Phi(x_i) \cdot \Phi(x)$ in Equ. (7) obtaining thus a decision rule expressed as:

$$\mathrm{SVM}(x) = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i \cdot y_i \cdot k(x_i, x) + b\right). \qquad (9)$$

3. 2 KNN-SVM Classifier

KNN-SVM [10] combines localities and searches for a large margin separating surfaces by partitioning the entire transformed feature space through an ensemble of local maximal margin hyper planes. In order to classify a given point x' in the input space, we first find its K-nearest neighbors in the transformed feature space F, and then search for an optimal separating hyper plane only over these K-nearest neighbors. In practice, this means that an SVM is built over the neighborhood of each test point x'. Accordingly, the constraints in Equ. (6) become:

$$y_{r_{x'}(i)}\left[w\Phi(x_{r_{x'}(i)}+b)\right]\geq 1-\xi_{r_{x'}(i)},\ i=1,\cdots,k \qquad (10)$$

Where, $r_{x'}:\{1,\cdots,N\}\rightarrow\{1,\cdots,N\}$ is a function, which maps the indexes of the training point defined in Equ. (4). In this way, $x_{r_{x'}(j)}$ is the j-th point of the set D in terms of distance from x' and thus

$$j<k\Rightarrow\|\Phi(x_{r_{x'}(j)})-\Phi(x')\|<\|\Phi(x_{r_{x'}(k)})-\Phi(x')\|$$

because of the monotonicity of the quadratic operator. The computation is expressed in terms of kernels as:

$$\|\Phi(x)-\Phi(x_i)\|^2=\langle\Phi(x),\Phi(x)\rangle_F+\langle\Phi(x'),\Phi(x')\rangle_F-2\langle\Phi(x),\Phi(x')\rangle_F=k(x,x)+k(x',x')-2k(x,x'). \qquad (11)$$

In the case of linear kernels, the ordering function can be built using the Euclidean distance, whereas if the kernel is not linear, the ordering can be different. If the kernel is the RBF kernel, the ordering function is equivalent to using the Euclidean metric. The decision rule associated with the method is:

$$\text{SVMNN}(x)=\text{sign}\left(\sum_{i=1}^{k}a_{r_x(i)}\,y_{r_s(i)}\,k(x_{r_x(i)},x)+b\right). \qquad (12)$$

3.3 Local support vector machines

KNN-SVM is a combination of KNN and SVM. But this method abandons nearest-neighbor searches in the SVM learning algorithm. Once K-nearest neighbors are identified, the SVM algorithm completely ignores their similarity to the given test example when solving the dual optimization problem given in Equ. (7).

So we developed a new LSVM algorithm, which incorporates neighborhood information directly into SVM learning. The principle of LSVM is to reduce the impact of support vectors located far away from a given test example. This can be accomplished by weighting the classification error of each training example according to its similarity to the test example. The similarity is captured by a distance function σ, the same as the approach used by KNN.

For each test sample x', we construct its local SVM model by solving the following optimization problem:

$$\min\frac{1}{2}\|w\|_2^2+C\sum_{i=1}^{n}\sigma(x',x_i)\xi_i,$$
$$\text{s.t.}\quad y_i(w^{T}x_i-b)\geq 1-\xi_i, \qquad (13)$$
$$\xi_i\geq 0,\ i=1,2,\cdots,n.$$

Where, $\sigma(x',x_i)$ is the L_2 distance between x' and x_i. The solution to Equ. (13) identifies the decision surface as well as the local neighborhood of the samples. The function σ penalizes training examples that are located far away from the test example. As a result, classification of the test example depends only on the support vectors in its local neighborhood. To further appreciate the role of the weight function, consider the dual form of Equ. (13):

$$\max\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\Phi(x_i,x_j),$$
$$\text{s.t.}\quad\sum_{i=1}^{n}\alpha_i y_i=0,\ 0\leq\alpha_i\leq c\sigma(x',x_i), \qquad (14)$$
$$i=1,2,\cdots,n.$$

Compared to Equ. (7), the difference between LSVM and SVM is that the constraint on the upper bound for α_i has been modified from c to $c\sigma(x',x_i)$. This modification has the following two effects: It reduces the impact of distant support vectors, and Non-support vectors of the nonlinear SVM may become support vectors of LSVM.

3.4 LSVM in facial expression recognition

For facial expression recognition using LSVM, geometric features are used as an input. Six classes were considered in the experiments, each one representing one of the basic facial expressions (anger, disgust, fear, happiness, sadness, and surprise). The LSVM classifies geometric features as one of these six basic facial expressions. Pseudo code of the basic version of the facial expression algorithm is given in Fig 4.

```
Input: Geometric feature sample of facial expression $x'$
        Training set: $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where $x_i \in R^d$, $x_i$ is the $i$-th
        geometric feature, $y_i = \{1, 2, 3, 4, 5, 6\}$, $y_i$ is the facial expression classifications
        Number of nearest neighbors $k$.
Output: facial expression classifications $y_p = \{1, 2, 3, 4, 5, 6\}$
1.  Find $k$ samples $(x_i, y_i)$ with minimal values of $k(x_i, x_i) - 2k(x, x_i)$,
2.  Train an modified multi-class SVM model on the $k$ selected samples, the modified SVM
    model incorporates the neighborhood information,
3.  Classify $x_i$ using this model, get the result $y_p$,
4.  return $y_p$.
```

Fig 4 The LSVM classifier for facial expression recognition

The LSVM makes binary decisions There are a number of methods for making multi-class decisions with a set of binary classifiers We employed pair-wise partitioning strategies For pair-wise partitioning (1:1), the SVM were trained to discriminate all pairs of emotions For six categories that makes 15 SVMs

4 Experiments and evaluations

In order to validate our proposed approach for facial expression recognition, we carried out experiments on a machine with a Pentium 4/2.0G CPU, 1GB memory, WindowsXP, and Visual C ++ 6.0. The Cohn-Kanade database[23] was used to recognize facial expression as one of the six basic facial expression classes (anger, disgust, fear, happiness, sadness, and surprise). Each video sequence starts with a neutral expression and ends with the peak of the facial expression. This database is annotated with AUs (Action Units). These combinations of AUs were translated into facial expressions according to Ref [24], in order to define the corresponding ground truth for the facial expressions All the subjects were used to form the database for the experiments The database contains 480 video sequences, containing 84 expressions of " fear ", 105 of " surprise ", 92 of " sadness ", 36 of " anger ", 56 of " disgust " and 107 of " happiness ". The upper row of Fig 5 shows the extraction of facial feature points in the initial frames in the video sequences for the 6 basic expression types, while the lower row shows that of the last frames of those video sequences

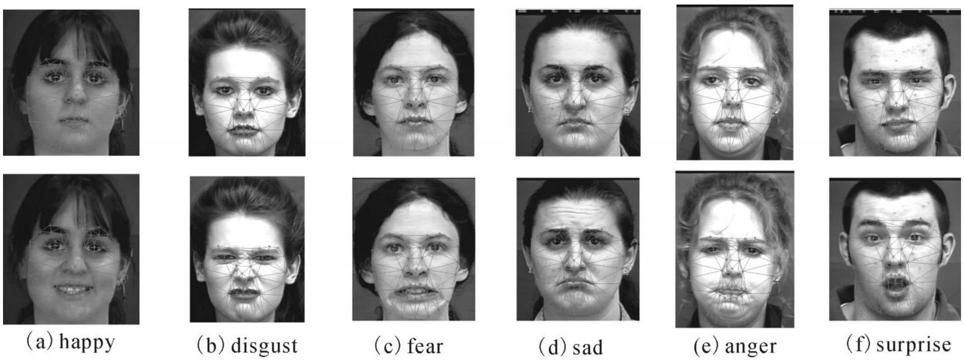
(a) happy (b) disgust (c) fear (d) sad (e) anger (f) surprise

Fig 5 ASM based facial feature points extraction examples

In our experiments, three classification algorithms, KNN, nonlinear SVM and SVM-NN were compared with our LSVM classifier to show its effectiveness Both KNN-SVM and LSVM employ a linear kernel The parameters of the classification algorithm, i.e. the k in KNN, c in SVM, bandwidth λ in the RBF kernel and k in LSVM were determined by 10-fold cross validation on the training set To implement the proposed LSVM algorithm, we modified the C ++ code of the LIBSVM (http://www. csie. ntu. edu. tw/cjlin/libsvm) tool developed by Chang and Lin to use $C\sigma$ as its upper bound constraint for α instead of c.

In order to make maximal use of the available data and produce averaged classification accuracy results, the experimental results reported in this study were obtained based on applying a 5-fold cross validation to the data sets More specifically, all image sequences contained in the database were divided into six classes, each one corresponding to one of the six basic facial expressions to be recognized. Five sets containing 20% of the data for each class, chosen randomly, were created. One set containing 20% of the samples for each class was used for the test set, while the remaining sets formed the training set After the classification procedure was performed, the samples forming the testing set were incorporated into the current training set, and a new set of samples (20% of the samples for each class) was extracted to form the new test set The remaining samples formed a new training set This procedure was repeated five times The average classification accuracy is the mean value of the percentages of the correctly classified facial expressions

First, we tested the facial expression recognition accuracy based on our proposed classifier LSVM. Confusion matrices were used to evaluate accuracy. The confusion matrix is a matrix containing 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 correctly classified facial expressions, while the off-diagonal entries correspond to misclassification rates The confusion matrix shown in Table 2 presents the results obtained while using the LSVM classifier. From this table, it can be seen that our method achieves 89. 11% overall recognition of facial expressions The confusion matrix confirms that some expressions are harder to differentiate than others Expressions identified as surprise or happiness are recognized with the highest accuracy (91. 32% and 92. 48%). For disgust, the recognition rate was 88. 32%, for fear it was 89. 64%, for sadness is 86. 38%, and for anger is 86. 54%. As can be seen, the most ambiguous facial expression was sadness The main reason is that both surprise and happiness cause obvious geometric shape changes when the facial expression moves from neutral to peak, while others may not produce enough geometric information to be as clearly discriminated.

Table 2　Confusion matrix based on LSVM

Inputs Results (%)	Happy	Surprise	Disgust	Fear	Sad	Anger
Happy	91. 32	1. 79	2. 11	0. 32	2. 58	1. 88
Surprise	2. 68	92. 48	1. 83	1. 32	0	1. 69
Disgust	2. 09	2. 17	88. 32	3. 08	1. 08	3. 26
Fear	0	0	3. 57	89. 64	4. 36	2. 43
Sad	1. 62	3. 56	2. 46	1. 78	86. 38	4. 20
Anger	2. 29	0	1. 71	3. 86	5. 60	86. 54

In addition, we conducted experiments to evaluate the performance of our proposed algorithms in comparison to KNN, nonlinear SVM and SVM-NN algorithms A 5-fold cross validation was also employed in these experiments Table 3 summarizes the results of our experiments Firstly, observe that, for the six basic facial expressions, nonlinear SVM outperforms the KNN algorithm. The accuracy of SVM is 85. 06% while the accuracy of KNN is only 78. 96%. Secondly, observe that SVM-NN fails to improve the accuracy over nonlinear SVM. In fact, the SVM-NN performance degrades, as classification accuracy drops from 85. 06% to 85. 03% when using SVM-NN instead of nonlinear SVM. One possible explanation for the poor performance of SVM-NN is the difficulty of choosing the right number of nearest neighbors (K) when the number of training examples is small. We observed that the LSVM algorithm consistently outperformed nonlinear SVM and KNN-SVM for the six basic facial expressions. We also found both surprise and happiness recognized with higher accuracy than other facial expressions, with the exception of the KNN classifier. Fig 6 shows the ROC curves of the four classifiers. It demonstrates that the LSVM classifier outperforms the SVM, KNN-SVM and KNN classifiers.

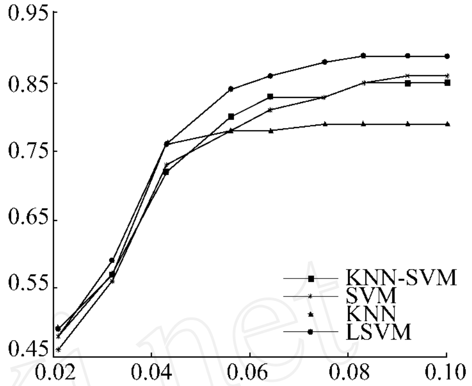

Fig 6 Roc curves of the four classifiers

Table 3 Classification accuracies (%) for SVM, KNN, KNN-SVM and LSVM

Inputs Results/%	Happy	Surprise	Disgust	Fear	Sad	Anger	Average accuracy
LSVM	91.32	92.48	88.32	89.64	86.38	86.54	89.11
SVM	88.09	88.67	85.86	85.71	82.41	79.62	85.06
SVM-NN	87.55	87.92	84.23	84.68	84.97	80.85	85.03
KNN	82.09	78.38	79.36	80.62	77.41	75.92	78.96

Fig 7 shows the six basic facial expression recognition results from our proposed system. Our method recognizes facial expressions from video sequences. The first frame shows a neutral expression while the last frame shows an expression with great intensity. In the last frame, we extract geometric features and classify the expression using LSVM.

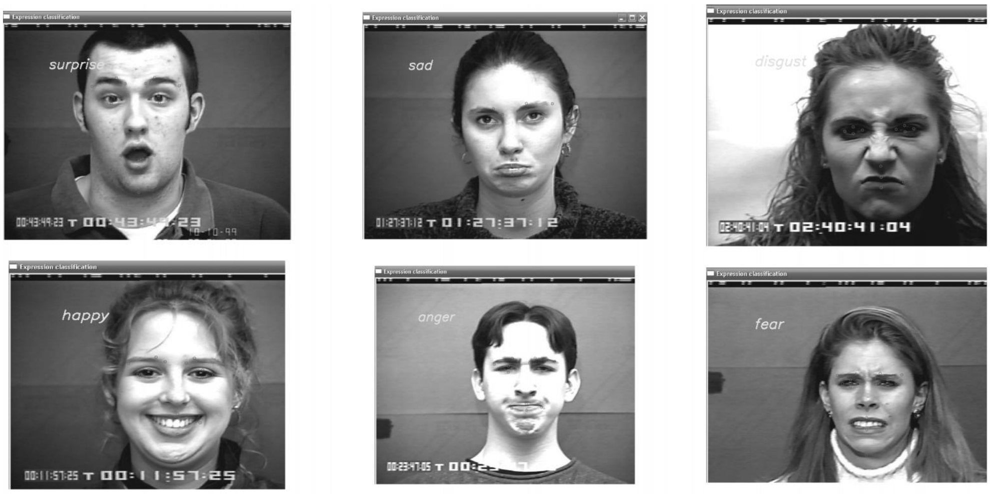

Fig 7 Facial expression recognition results in our proposed system

5 Conclusion

In this paper, we proposed an automatic method for recognizing prototypical expressions that include anger, disgust, fear, joy, sadness and surprise. We tracked the facial feature points using ASM and extracted geometric features from video sequences. To improve facial expression recognition accuracy, we pres-

ented a new LSVM classifier for classifying the expressions Experiments on laboratory data (Cohen-Kanade) showed 89. 11% recognition accuracy on 480 video sequences At the same time, we compared KNN, SVM, and KNN-SVM classifiers with the LSVM. The LSVM classifier produced the best experimental results

References:

[1] FRANCO L, TREVES A. A neural network facial expression recognition system using unsupervised local processing [C]// Proceeding of the 2nd International Symposium on Image and Signal Processing and Analysis (ISPA2001). Pula, Croatia, 2001: 628-632.

[2] LYONS M, AKAMATSU S. Coding facial expressions with gabor wavelets[C]//Proceeding of the Third IEEE International Conference on Automatic Face and Gesture Recognition. Nara, Japan: IEEE Computer Society, 1998: 454-459.

[3] BASSLI J. Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face[J]. J Personality Social Psychol, 1979, 37: 2049-2059.

[4] FASEL B, LUETTIN J. Recognition of asymmetric facial action unit activities and intensities [C]//Proceeding of 15th International Conference on Pattern Recognition (ICPR 2000). Barcelona, Spain, 2000: 1100-1103.

[5] TIAN Y L, KANADE T, COHN J F. Recognizing action units for facial expression analysis[J]. IEEE Transaction on PAMI, 2001, 23(2): 97-115.

[6] BARTLETT M S, LITTLEWORT G, FRANK M G, LAINSCSEK C, FASEL I, MOVELLAN J. Recognizing facial expression: Machine learning and application to spontaneous behavior[C]// Proceeding of Conference on Computer Vision and Pattern Recognition (CVPR 2005). San Diego, CA, USA: IEEE Computer Society, 2005: 568-573.

[7] COHEN I, SEVE N, COZMAN G G, CIRELO M C, HUANG T S. Learning Bayesian network classifier for facial expression recognition using both labeled and unlabeled data [C]// Proceeding of Conference on Computer Vision and Pattern Recognition (CVPR 2003). Madison, Wisconsin, USA: IEEE Computer Society, 2003: 595-601.

[8] LIEN J J, KANADE T, COHN J F, LI C C. Detection, tracking, and classification of action units in facial expression [J]. Journal of Robotics and Autonomous Systems, 2000, 31: 131-146.

[9] LISETTI C L, SCHIANO D J. Automatic facial expression interpretation: where human-computer interaction, artificial intelligence and cognitive science intersect[J]. Pragmatics & Cognition, 2000, 8: 185-235.

[10] ZHANG H, BERG A C, MAIRE M, MALIK J. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition[C]// Proceeding of Conference on Computer Vision and Pattern Recognition (CVPR 2006). New York, USA: IEEE Computer Society, 2006: 2126-2136.

[11] BASSLI J. Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face[J]. J Personality Social Psychol, 1979, 37: 2049-2059.

[12] EKMAN P, FRIESEN W V. Facial action coding system: investigator's guide[M]. Palo Alto: Consulting Psychologists Press, 1978: 156-163.

[13] PANTIC M, ROTHKRANTZ L J M. Automatic analysis of facial expressions: the state of the art[J]. IEEE Trans on PAMI, 2000, 22(12): 1424-1445.

[14] MASE K. Recognition of facial expression from optical flow [J]. IEICE Transactions on Communications (Special Issue on Computer Vision and its Applications), 1991, 10: 3474-3483.

[15] BLACK M J, YACOOB Y. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion [C]//Proceeding of Fifth International Conference on Computer Vision (ICCV95). Cambridge, MA, USA, 1995: 374-381.

[16] OLIVER N, PENTLAND A, ERARD F B. LAFTER: a real-time face and lips tracker with facial expression recognition[J]. Pattern Recognition, 2000, 33: 1369-1382.

[17] BARTLETT M S, LITTLEWORT G, FASEL I, MOVELLAN J R. Real time face detection and facial expression recognition: development and applications to human computer interaction[C]//Proceeding of Conference on Computer Vision and Pattern Recognition (CVPR 2003). Madison, Wisconsin, USA: IEEE Computer Society, 2003: 53-58.

[18] COHEN I, SEBE N, GARG S, CHEN L S, HUANGA T S. Facial expression recognition from video sequences: temporal and static modeling[J]. Computer Vision and Image Understanding, 2003, 91(1-2): 160-187.

[19] COOTES T F, TAYLOR C J, COOPER D H, GRAHAM

J. Active shape models-their training and application[J]. Computer Vision and Image Understanding, 1995, 61(1): 38-59.

[20]KOBAYASHI H, HARA F. Facial interaction between animated 3D face robot and human beings[C]//Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. IEEE Computer Society Press, 1997: 3732-3737.

[21]COVER T, HART P. Nearest neighbor pattern classification[J]. IEEE Transactions in Information Theory, 1967, 13: 21-27.

[22]BURGES C J C. A tutorial on support vector machines for pattern recognition[J]. Knowledge Discovery and Data Mining, 1998, 2: 121-167.

[23]KANADE T, COHN J, TIAN Y. Comprehensive database for facial expression analysis[C]//Proceedings of Fourth IEEE International Conference on Face and Gesture Recognition. IEEE Computer Society Press, 2000: 46-53.

[24]PANTIC M, ROTHKRANTZ L J M. Expert system for automatic analysis of facial expressions[J]. Image and Vision Computing, 2000, 18(11): 881-905.

作者简介:

孙正兴,男,1964年生,教授、博士生导师,中国图像图形学会计算机动画与数字艺术专委会常务委员,中国计算机学会计算机辅助设计与图形学专委会委员,中国人工智能学会人工心理与人工情感专委会委员,江苏省微型电脑应用协会副理事长兼多媒体技术专委会主任,江苏省计算机学会计算机辅助设计与图形学专委会主任. 主要研究方向为多媒体计算、计算机视觉和环境智能. 获省部级科技进步三等奖 3次. 已在国内外重要刊物上发表学术论文 90余篇,主编教材 3部、译著 1部.

徐文晖,男,1984年生,硕士研究生,主要研究方向为计算机视觉与智能人机交互.

第 3届中国智能计算大会 ICC2009

The Third Intelligent Computing Conference

为了推动智能计算理论、方法及应用的发展,活跃该领域的研究,交流总结我国智能计算研究工作者的最新成果,由中国运筹学会智能计算分会主办和德州学院承办的第 3届中国智能计算大会将于 2009年 5月 15～19日在美丽的泉城济南召开.

会议主题:

1)智能计算

遗传算法、模拟退火算法、禁忌搜索算法、进化算法、启发式算法、蚁群算法、粒子群算法、混合智能算法、免疫算法、人工智能、神经网络、机器学习、生物计算、DNA计算、量子计算、智能计算与优化、模糊逻辑、模式识别、知识发现、数据挖掘.

2)不确定理论

随机集、模糊集、粗糙集、可信性理论、机会理论、不确定规划、随机规划、模糊规划、混合规划.

3)应用

车辆调度问题、可靠性问题、存贮问题、排序问题、选址问题、分配问题、更新问题、图像处理、电子商务、信息安全、风险分析与控制、智能加工系统、智能调度系统、智能交通系统、智能金融工程、智能信息检索、智能控制与自动化、智能通讯工程等.

会议网站: http://211.64.47.133/web/conf2009.