Boosted multi-resolution spatiotemporal descriptors for facial expression recognition

Guoying Zhao *, Matti Pietikäinen

Machine Vision Group, Infotech Oulu and Department of Electrical and Information Engineering, P.O. Box 4500, University of Oulu, FI-90014 Oulu, Finland

ARTICLE INFO

Article history:
Available online 5 April 2009

Keywords:
Principal appearance and motion
Spatiotemporal descriptors
Facial expression recognition
AdaBoost

ABSTRACT

Recently, a spatiotemporal local binary pattern operator from three orthogonal planes (LBP-TOP) was proposed for describing and recognizing dynamic textures and applied to facial expression recognition. In this paper, we extend the LBP-TOP features to multi-resolution spatiotemporal space and use them for describing facial expressions. AdaBoost is utilized to learn the principal appearance and motion, for selecting the most important expression-related features for all the classes, or between every pair of expressions. Finally, a support vector machine (SVM) classifier is applied to the selected features for final recognition.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

A goal of facial expression recognition is to determine the emotional state of the face, e.g. happiness, sadness, surprise, neutral, anger, fear, and disgust, regardless of the identity of the face. The face can express emotions sooner than people verbalize or even realize their feelings (Tian et al., 2001), and research in social psychology has shown that facial expressions form the major modality in human communication (Ekman and Davidson, 1994). So facial expression is one of the most powerful, natural and immediate means for human beings to communicate their emotions and intentions (Shan et al., 2005a). The recognition of facial expressions is very important for interactive human–computer interfaces. Even though much work has been done, recognizing facial expression with a high accuracy remains to be difficult due to the complexity and variety of facial expressions (Shan et al., 2005a).

Pantic and Rothkrantz (2000) gave an overview of automatic expression recognition, presenting the main system components and some research challenges. In another survey by Fasel and Luettin (2003), the most prominent automatic facial expression analysis methods and systems were introduced. They also discussed some facial motion and deformation extraction approaches as well as classification methods.

According to psychologists (Bassili, 1979), analysis of sequences of images produces more accurate and robust recognition of facial expressions than using only single frames. Psychological studies have suggested that the facial motion is fundamental to the recognition of facial expressions. Experiments conducted by Bassili (1979) demonstrate that the humans do better job in recognizing expressions from dynamic images as opposed to mug shot.

For using dynamic information to analyze facial expressions, several systems attempt to recognize fine-grained changes in facial expression based on the Facial Action Coding System (FACS) which was developed by Ekman and Friesen (1978) for describing facial expressions by action units (AUs), for instance see Refs. (Bartlett et al., 1999; Donato et al., 1999; Kanade et al., 2000; Tian et al., 2001). Some other papers attempt to recognize a small set of prototypic emotional expressions, i.e. joy, surprise, anger, sadness, fear, and disgust. Our work focuses on the latter. Yeasin et al. (2004) applied the horizontal and vertical components of the optic flow as features. At the frame level, a k-NN rule was used to derive characteristic temporal signature for every video sequence. At the sequence level, discrete HMMs were trained to recognize the temporal signatures associated with each of the basic expressions. But this method cannot deal with the illumination variation. Manglik et al. (2004) present a method for extracting position of the eyes, eyebrows and mouth, then determining the cheek and forehead regions. The optical flow procedure is applied to these regions and the resulting vertical optical flow values are fed to the discrete Hopfield network. Their dataset only included 20 samples, obtaining a result of 79.8%. Aleksic and Katsaggelos (2006) exploit Facial Animation Parameters as features describing facial expressions, and utilize multi-stream Hidden Markov Models for recognition. The system is complex, thus difficult to perform in real-time. Cohen et al. (2002) introduce a Tree-Augmented-Naive Bayes classifier for recognition. But they only experimented on a set of five people, and accuracy is only around 65% for person-independent evaluation. Tian (2004) applied Gabor filters to extract appearance

* Corresponding author. Tel.: +358 8 553 7564; fax: +358 8 553 2612.
E-mail addresses: gyzhao@ee.oulu.fi (G. Zhao), mkp@ee.oulu.fi (M. Pietikäinen).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.03.018
features and a three-layer neural network to recognize expressions. The results for low resolution images were quite poor, however.

Recently, a block-based approach based on local binary patterns (LBP) originally developed for single face images (Ahonen et al., 2006) was extended for the recognition of specific dynamic events such as facial expressions using spatiotemporal information (Zhao and Pietikäinen, 2007). Local binary patterns from three orthogonal planes or slices (LBP-TOP) were proposed. They can effectively describe appearance, horizontal motion and vertical motion from the video sequence. The block-based LBP-TOP has been successfully used for facial expression recognition. But it used all the block features which makes the feature vector too long and thus the recognition cannot be done in real-time. In this paper, we propose multiresolution features, computed from different sized blocks, different neighboring samplings and different sampling scales, and utilize AdaBoost to select the slice features for all the expression classes or every class pair, to improve the performance with short feature vectors. Finally, on the basis of selected slices, we work out the location and feature types of most discriminative features for every class pair. The preliminary results of this work were reported in (Zhao and Pietikäinen, 2008).

2. Spatiotemporal local binary patterns

The local binary pattern (LBP) operator is a gray-scale invariant texture primitive statistic, which has shown excellent performance in the classification of various kinds of textures. For each pixel in an image, a binary code is produced by thresholding its neighborhood with the value of the center pixel (Fig. 1a and Eq. (1)).

\[
LBP_{p,r} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}
\]

where \(g_c\) corresponds to the gray value of the center pixel \((x_c, y_c)\) of the local neighborhood and \(g_p\) to the gray values of \(P\) equally spaced pixels on a circle of radius \(R\). By considering simply the signs of the differences between the values of neighborhood and the center pixel instead of their exact values, LBP achieves invariance with respect to the scaling of the gray-scale.

![Fig. 1.](image)

(a) Basic LBP operator. (b) The circular (8,2) neighborhood.

A histogram is created to collect up the occurrences of different binary patterns. The definition of neighbors can be extended to include circular neighborhoods with any number of pixels, as shown in Fig. 1b. In this way, one can collect larger-scale texture primitives.

Local texture descriptors have gained increasing attention in facial image analysis due to their robustness to challenges such as pose and illumination changes. Ahonen et al. (2006) proposed LBP-based facial representation for face recognition from static images. In this approach, the face image is divided into several regions (blocks) from which the LBP features are extracted and concatenated into an enhanced feature vector. This approach is proving to be a growing success. It has been adopted and further developed by many research groups, and has been successfully used for face recognition, face detection and facial expression recognition (Ahonen et al., 2006). All of these have applied LBP-based descriptors only for static images, i.e. they do not utilize temporal information as used in this paper.

Recently, two spatiotemporal local patterns: Volume Local Binary Patterns (VLBP) and LBP from Three Orthogonal Planes (LBP-TOP) were proposed (Zhao and Pietikäinen, 2007). The traditional LBP for static images was extended to spatiotemporal domain. More generally for LBP-TOP, the radii in axes \(X, Y\) and \(T\), and the number of neighboring points in the \(XY\), \(XT\) and \(YT\) planes can also be different, which can be marked as \(R_x, R_y\) and \(R_T\), \(P_{XY}, P_{XT}\) and \(P_{YT}\), the corresponding LBP-TOP feature is denoted as LBP-TOP \(P_{XY}, P_{XT}, P_{YT}, R_x, R_y, R_T\) series. Sometimes, the radius in three axes are same and so do the number of neighboring points in \(XY\), \(XT\) and \(YT\) planes. In that case, we use LBP-TOP \(P_8\) for abbreviation where \(P = P_{XY} = P_{XT} = P_{YT}\) and \(R = R_x = R_y = R_T\).

For its properties of describing the spatiotemporal signals, robustness to illumination variation and local characteristics, LBP-TOP can be utilized to represent the facial expressions (Zhao and Pietikäinen, 2007). Considering the motion of face region, the descriptors are obtained by concatenating local binary patterns on three orthogonal planes or slices from expression sequence: \(XY\), \(XT\) and \(YT\), calculating only the co-occurrence statistics in these three directions. An LBP description computed over the whole face sequence encodes only the occurrences of the micro-patterns without any indication about their locations. To overcome this effect, a representation which consists of dividing the face image into several overlapping blocks was introduced, as shown in Fig. 2, from which the local binary pattern histograms from three planes in each block are computed and concatenated into a single histogram (Zhao and Pietikäinen, 2007). All features extracted from each block volume are connected to represent the appearance and motion of the facial expression sequence.

A LBP-TOP histogram of the facial expression can be defined as

\[
H_{b,c,t} = \sum_{i=0}^{n_t-1} \sum_{j=1}^{2} I(f_i(x,y,t) = i), \quad i = 0, \ldots, n_t - 1; \quad j = 0, 1, 2.
\]

![Fig. 2.](image)

Features in each block volume. (a) Block volumes; (b) LBP features from three orthogonal planes; (c) concatenated features for one block volume with the appearance and motion.
in which \( n_j \) is the number of different labels produced by the LBP operator in the \( j \)th plane (\( j = 0 : XY, 1 : XT \) and \( 2 : YT \)), \( f_j(x, y, t) \) expresses the LBP code of central pixel \((x, y, t)\) in the \( j \)th plane, \( x \in \{ R_v, \ldots, X - 1 - R_v \} \), \( y \in \{ R_v, \ldots, Y - 1 - R_v \} \), \( t \in \{ R_l, \ldots, T - 1 - R_l \} \) (\( X \) and \( Y \) are width and height of image and \( T \) is the utterance length). \( b \) is the index of rows, and \( c \) is of columns

\[
I(A) = \begin{cases} 
1, & \text{if } A \text{ is true}; \\
0, & \text{if } A \text{ is false}. 
\end{cases} \tag{3}
\]

The histograms must be normalized to get a coherent description:

\[
N_{b,c,j} = \frac{H_{b,c,j}}{\sum_{k=0}^{l-1} H_{b,c,k}}. \tag{4}
\]

3. Boosted multi-resolution features

3.1. Multi-resolution features

From the previous work on facial expressions, we can conclude that using a combination of different features could improve the performance (Zhao and Pietikäinen, 2007). So we propose to use features from different resolutions. The generic Block-based LBP-TOP can be denoted as LBP-TOP\(_{\text{TOP-v}}\) \(P_{XY}, P_{XT}, P_{YT}\), \(R_v, R_l, R_T\), where, super-script \(\text{TOP-v}\) means the feature type: \(XY\) (appearance), \(XT\) (horizontal motion) and \(YT\) (vertical motion) from the \(v\)th block volume, where \(0 \leq v < r \times l\). It is related to feature types, but has nothing to do with resolutions, so \(\text{TOP-v}\) keeps the same in multi-resolution analysis. The subscripts mean the parameters of the features, which can be divided into three groups. The first is \(\{P_{XY}, P_{XT}, P_{YT}\}\), which are the number of sampled neighboring points in \(XY, XT, YT\) slices, respectively. The second is \(\{R_v, R_l, R_T\}\), which are the radii of circles along three axes: \(X, Y, T\). The third one is \(r \times l\), which represents the number of rows \(r\) and columns \(l\).

The whole video sequence can be divided into \(r \times l\) sub-volumes, and inside each sub-volume the LBP-TOP features are computed to describe the characteristic of the sub-volume, and finally are connected together to represent the videos. With changing these three groups of parameters, three different types of spatiotemporal resolution are presented: (1) Use of different number of neighboring points when computing the features in \(XY, XT\) and \(YT\) slices, as shown in Fig. 3; (2) Use of different radii which can catch the occurrences in different space and time scales, as demonstrated in Fig. 4; (3) Use of blocks of different sizes to have global and local statistical features, as Fig. 5 shows. The first two resolutions focus on the pixel-level in feature computation, providing different local spatiotemporal information, while the third one focuses on the block or volume level, giving more global information in space and time dimensions. In some methods developed for static face image analysis, features are extracted from regions obtained by shifting and scaling a sub-window (Shan et al., 2005b; Izquierdo et al., 2004). Those methods just consider the location issue, but miss the multi-resolution of the feature itself. Moreover, not only their features, but also their windows do not take the time factor into account. So these methods are limited in their ability to describe motion. Intuitively, using a combination of features from different resolutions should provide more information in various spatiotemporal scales to improve the performance of the

![Fig. 4. Different radii in one slice.](image)

![Fig. 5. Different block sizes.](image)
basic spatiotemporal approach. In this paper, we propose to use features from different spatial and temporal resolutions to describe dynamic events.

3.2. Feature selection

If features from different resolutions were concatenated directly, the feature vector could be very long, which would make the computational complexity very high and the same contribution from different features would decrease the performance. Normally speaking, all spatial temporal features do not contribute equally. In what location, what features, appearance or motion are more important, is the question we will address. In order to find the most distinguishable features, AdaBoost is utilized to learn those features.

Boosting algorithm (Schapire, 2002) is a modern statistics method. Theoretically speaking, it could improve the performance of any learning algorithm. AdaBoost (Freund and Schapire, 1995, 1996) is a sort of self-adaptive Boosting method, which is mainly used to integrate some weak classifiers into a strong classifier. The computational complexity of weak classifiers is much smaller, which makes them much faster to execute.

The set of intra/extra features generated by extracting LBP-TOP from different block sizes and neighboring points is an over-complete set for facial expression patterns and contains much redundant information. We propose to use AdaBoost to select the most significant intra/extra features from a large feature set. Therefore, AdaBoost is adopted to learn the effective features: location, resolution, appearance or motion from a large feature set.

The basic form of (discrete) AdaBoost is for two-class problems. A set of \( N \) labeled training examples is given as \((x_1, y_1), \ldots, (x_N, y_N)\), where \( y_i \in \{+1, -1\} \) is the class label for the example \( x_i \in \mathbb{R}^n \). AdaBoost learns a sequence of weak classifiers, with respect to the distributions of training samples. Here, we adopt AdaBoost learning to learn effective features from a large feature set, including location, resolution, and appearance or motion types of features. Here a training sample \( x \) is defined as the Chi Square distances of LBP-TOP histograms from all the blocks with different resolutions, each block including three Chi Square distances. So a training sample is made of \( 3V \) scalars \((V \) is block numbers with different resolutions).

Previous works show that some facial areas contain more discriminative information than others in terms of distinguishing between subjects or expressions. To take advantage of these cues, a weight can be set to each area based on its contribution to classification (Ahonen et al., 2006) or some areas or features (Hadid et al., 2007; Niu et al., 2006; Shan et al., 2005b; Yang et al., 2007; Zhang et al., 2004) are selected. But in those works, they have following limitations: (1) do selection for all single features (Niu et al., 2006; Yang et al., 2007), which would lose the global property; or do that in blocks (block-based method) (Hadid et al., 2007; Zhang et al., 2004), just considering the location importance. (2) Train for all the expressions, losing the most important distinguishable features between expressions, such as sadness and anger, which are hard pairs for classification.

So we propose firstly to boost the slice features in order to find out at what location and which slices (xy slice, xt slice or yt slice) are more important, this is called slice-based selection; secondly, we train AdaBoost between not only all expression classes (All–All), but also every class pair (One–One), so that the discrimination between every two classes would be obtained.

First strategy is using all the expressions to make feature selection together which is commonly used to handle the multi-class cases in feature selection (Hadid et al., 2007; Zhang et al., 2004). We put the features from same classes as positive samples, and from different class pairs as negative samples. The goal is to find out the general variations between classes. But in this way, all classes would learn the same features, losing the most discriminative features between classes. For example, in the facial expression recognition, sadness and anger are quite similar and hard to discriminate even for human. If we could figure out the specific differences between this pair, it would be helpful to improve the further analysis.

Second strategy is using every expression class pair to make feature selection. That means that the learners are designed for every pair of two classes and the aim is to learn more specific and discriminative features for each pair. For the training sample \( a \) and \( b \) in class \( C(l) \) and class \( C(j) \), compute \( \chi_i = \{\chi_{XY}, \chi_{XT}, \chi_{YT}\} \), if \( a \), \( b \) belong to same classes, \( y_i \) is labeled as +1, otherwise, labeled as −1. Labeled training examples are given as \((x_1, y_1), \ldots, (x_N, y_N)\), value of \( y_i \) is +1 or −1, and they are fed into learners. AdaBoost selects features for discriminating classes \( C(l) \) and \( C(j) \). In same way, the features are learned for each class pair. Selected features are different depending on class pairs and are more related to intra and extra class variations of the two specific classes in question.

In this histogram, we effectively have a description of the face on two different levels of locality: the labels for the histogram contain information about the patterns in pixel-level, and the labels are summed over a small region slice to produce information on regional level (Cohen et al., 2002). Several possible dissimilarity measures are available. In this work, Chi square statistic \( \chi^2 \) in Eq. (2) is adopted after the comparison to log statistics and histogram intersection:

\[
\chi^2(S, M) = \sum_{i=0}^{n-1} \frac{(S_i - M_i)^2}{(S_i + M_i)}
\]

where \( S \) and \( M \) are two slice histograms and \( n \) is the bin number of the histogram.

3.3. SVM classification

After AdaBoost learning, discriminative features are selected and the dimensionality is reduced. A support vector machine (SVM) classifier is explored on the features selected by AdaBoost, since it is well founded in statistical learning theory and has been successfully applied to various object detection tasks in computer vision. SVM is only used for separating two sets of points, the six-expression classification problem is decomposed into six one-against-rest or 15 two-class problems (happiness–surprise, anger–fear, sadness–disgust, etc.), then a voting scheme is used to accomplish recognition. We compared the one-against-rest and one-against-one, the latter got better performance. Sometimes more than one class gets the highest number of votes. In this case, 1–NN template matching is applied to these classes to reach the final result. This means that in training, the spatiotemporal LBP histograms of face sequences belonging to a given class are averaged out the general variations between classes. But in this way, all classes would learn the same features, losing the most discriminative features between classes. For example, in the facial expression recognition, sadness and anger are quite similar and hard to discriminate even for human. If we could figure out the specific differences between this pair, it would be helpful to improve the further analysis.

Here, after the comparison of linear, polynomial and RBF kernels in experiments, we use the second degree polynomial kernel function \( K(\cdot, \cdot) \) defined by: \( K(x, t) = (1 + x \cdot t)^2 \), which provided the best results.
4. Experiments

The proposed method was tested on the Cohn–Kanade database. Boosted Haar features (Viola and Jones, 2001) are used for automatic coarse face detection and 2D Cascaded AdaBoost (Niu et al., 2006) is applied for localizing eyes in the detected faces. Because the face images in the database are of good quality and almost all images are frontal faces, detection of faces and eyes is quite easy, providing almost 100% detection accuracy. The positions of the two eyes in the first frame of each sequence were determined and then these positions were used to define the facial area for the whole sequence. We used this because the positions of the heads in the database are near frontal and keep stable. Though there may be some translation – in-plane rotation and out-of-plane rotation of the head – no further alignment of facial features, such as alignment of the mouth, was performed in our algorithm. The whole sequence was used to extract the proposed spatiotemporal LBP features.

In the experiments presented in (Zhao and Pietikäinen, 2007), the results with radius three were much better than with the other radii. So for the multi-resolution features, the radius is kept as three in our study. The following multi-resolution features with different block volume sizes and different neighboring points are used for feature selection, as shown in Table 1. If all these features were used separately, the best result 91.44% is for LBP-TOP\(3, 3, 8\).

### Table 1

<table>
<thead>
<tr>
<th>Features</th>
<th>Slices</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP-TOP(3, 3, 3)</td>
<td>192</td>
<td>91.44</td>
</tr>
<tr>
<td>LBP-TOP(3, 4, 4)</td>
<td>48</td>
<td>89.04</td>
</tr>
<tr>
<td>LBP-TOP(3, 2, 2)</td>
<td>12</td>
<td>76.20</td>
</tr>
<tr>
<td>LBP-TOP(3, 1, 1)</td>
<td>3</td>
<td>60.70</td>
</tr>
<tr>
<td>LBP-TOP(3, 2, 8)</td>
<td>192</td>
<td>90.91</td>
</tr>
<tr>
<td>LBP-TOP(4, 4, 4)</td>
<td>48</td>
<td>88.24</td>
</tr>
<tr>
<td>LBP-TOP(4, 2, 2)</td>
<td>12</td>
<td>77.01</td>
</tr>
<tr>
<td>LBP-TOP(4, 1, 1)</td>
<td>3</td>
<td>55.35</td>
</tr>
</tbody>
</table>

4.1. Adaboost vs. Fisher linear discriminant

Fisher linear discriminant (Duda et al., 2001) can achieve the purpose of high separability between different patterns. It is usually used to select features or assign the weights. It is generally believed that the similarities of different samples from the same class are higher than those from the different classes. Based on this intuition, two distinct and mutually exclusive classes are defined: \(\Omega_b\) representing the inter-class similarities (corresponding to the similarity between two local region LBP-TOP histograms extracted from two video sequences of different expressions) and \(\Omega_w\) representing the intra-class similarities (corresponding to the similarity between two local region LBP-TOP histograms extracted from two videos of same expression). The mean \(m_{w(x,y)}\) and variance \(S_{w(x,y)}^2\) of intra-class similarities \(\Omega_w\) can be defined as:

\[
m_{w(x,y)} = \frac{1}{N_w} \sum r \sum c \sum v (H^r_{x,y}, H^c_{x,y}),
\]

\[
S_{w(x,y)}^2 = \frac{1}{N_w} \sum r \sum c \sum v (H^r_{x,y}, H^c_{x,y}) - m_{w(x,y)}^2,
\]

where \(r\) is the index of rows, \(c\) is of columns, and \(H^r_{x,y}\) expresses the LBP-TOP histogram in the \(r\)th row and \(c\)th column for \(x\)th video sequence. \(N_w\) is the number of intra-class samples.

The mean \(m_{b(x,y)}\) and variance \(S_{b(x,y)}^2\) of inter-class similarities can be defined as:

\[
m_{b(x,y)} = \frac{1}{N_b} \sum r \sum c \sum v (H^r_{x,y}, H^c_{x,y}),
\]

\[
S_{b(x,y)}^2 = \frac{1}{N_b} \sum r \sum c \sum v (H^r_{x,y}, H^c_{x,y}) - m_{b(x,y)}^2,
\]

where \(H^r_{x,y}\) and \(H^c_{x,y}\) denote the \((r, c, v)\)th local region histograms from two samples of the different expressions, \(N_b\) is the number of samples in inter-class. The slice with higher separability should
have higher $S^2_{b(r,c,v)}$ and lower $S^2_{w(r,c,v)}$. Therefore, the weight of each region can be obtained by Fisher linear discriminant:
\[
W_{r,c,v} = \frac{S^2_{b(r,c,v)}}{S^2_{w(r,c,v)}}.
\]

A higher value of $W_{r,c,v}$ represents a better class separability of the region. The $N$ histograms with the highest $N W_{r,c,v}$ could be selected.

The total number of slices is $(8 \times 8 \times 3 + 4 \times 4 \times 3 + 2 \times 2 \times 3 + 1 \times 1 \times 3) \times 2 = 510$. In Table 2 and Fig. 6, the same number of slices is selected for AdaBoost and Fisher discriminant algorithms but the results from AdaBoost are better, especially when the number of slices reduces. With just 45 slices, the recognition result for AdaBoost is almost 93%, much better than that for Fisher discriminant, 86.9%.

4.2. Block-based vs. slice-based

Figs. 7 and 8 give a visual impression of which features are selected. We can see that using the block-based selection (Fig. 8) only
the location is learned, so the selected features include all the three slices. But in the same location, the different kinds of features do not contribute equally. The slice-based selection (Fig. 7) can obtain not only the location, but also the feature type: appearance, $XT$ motion or $YT$ motion, which makes the features more suitable and adaptive for classification. Furthermore, as shown in Fig. 7, the expression-related features are concentrated in the corners of eyes, mouth corner, sides of the nose, and the eyebrows. These are the important areas for making expressions, not like identity-related features which are in the eyes, mouth and nose regions.

Table 3 lists the results for slice-based and block-based methods. Due to its flexibility, the slice-based method provides better results with the same number of features.

4.3. One-against-one vs. all-against-all

Fig. 9 shows the selected features for two expression pairs (For clear presentation, we just give the selected slices for the $8 \times 8$ block size with eight neighboring points. The experimental results given in Table 4 are from the feature selection using all multi-resolution features). They are different and specific depending on the expressions; not like in Fig. 7 where the selected slices are for all expressions with the specificity missing for discriminating different two expressions. As seen from the left part of Fig. 9, the more discriminative features for sadness and anger focus on the horizontal motion (seven $XT$ slices from total 15) which are in the side of the nose and the mouth corner, and vertical motion (six $YT$ slices) which are in the eyebrows. For happiness and fear (two images on the right in Fig. 9), the learned features focus on the horizontal motion (seven $XT$ slices) which are in forehead, the area between eyes, and between the nose and the mouth, and the appearance (five $XY$ slices) on the forehead and the mouth corner.

Table 4 gives the results for one-against-one and all-against-all strategies. All-against-all focuses on learning the more global expression-related features and identity-related features. One-against-one learns the most discriminative features between two specific expressions, which makes the expressions more distinguishable locally. And because one-against-one SVM classifiers are exploited, the selected features from one-against-one AdaBoost learning can make the SVM more adaptive and effective for the classification. The accuracy for one-against-one is better than that for all-against-all, as demonstrated in Table 4. Fig. 10 illustrates the performance for different strategies with different number of selected slices. For block-based method, if one block is selected, all

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Results from slice-based and block-based feature selection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected features</td>
<td>Slice-based (%)</td>
</tr>
<tr>
<td>90 slices (30 blocks)</td>
<td>92.25</td>
</tr>
<tr>
<td>60 slices (20 blocks)</td>
<td>91.98</td>
</tr>
<tr>
<td>45 slices (15 blocks)</td>
<td>92.78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Results from One–One and All–All feature selection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected features</td>
<td>One–One (%)</td>
</tr>
<tr>
<td>90 slices</td>
<td>93.58</td>
</tr>
<tr>
<td>60 slices</td>
<td>93.85</td>
</tr>
<tr>
<td>45 slices</td>
<td>93.05</td>
</tr>
</tbody>
</table>

Fig. 9. Selected 15 slices for expression Sadness vs. Anger (left) and Happiness vs. Fear (right).

Fig. 10. Performance comparison with selected features from different learning strategies.
three slices inside are selected. So the number of its selected slices are always the multiples of three. The learned multi-resolution features work much better with few slices (60 slices) than the result given by separate features LBP-TOP$^{8,8,8,3,3,3}$ with 192 slices (93.85% vs. 91.44%). Especially when the number of selected slices is very small, e.g. less than 10, One–One obviously outperforms All–All method. We have also tried one-against-rest strategy, but the performance is much worse than that from the One–One learning. That is perhaps because that the negative samples for the learners are from one class against all other classes. The variations could not be that specific and discriminating like one-against-one, which affects the learning to a large extent. Table 5 gives the

<table>
<thead>
<tr>
<th></th>
<th>Surprise</th>
<th>Happiness</th>
<th>Sadness</th>
<th>Fear</th>
<th>Anger</th>
<th>Disgust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surprise</td>
<td></td>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Happiness</td>
<td>1</td>
<td></td>
<td>2(4)</td>
<td>3(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sadness</td>
<td>1(1)</td>
<td></td>
<td>2(2)</td>
<td>(1)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Fear</td>
<td>(2)</td>
<td></td>
<td>3(3)</td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anger</td>
<td></td>
<td></td>
<td>(1)</td>
<td>1(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disgust</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>(1)</td>
</tr>
</tbody>
</table>

Table 5
Confusion matrix with selected 60 slices from One–One and All–All strategies, respectively (inside the parentheses is given the number of misclassified video sequences for All–All strategy).

Fig. 11. Demonstration of selected slices for each pair of expressions. Red ‘‘j” in the blocks means the YT slice (vertical motion) is selected, and ‘‘–” the XT slice (horizontal motion), ‘’/” means the appearance XY slice (appearance). (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)
confusion matrix with 60 selected slices from One–One and All–All strategies, respectively. It can be seen that by using the specific features for different class pairs, the number of misclassified samples reduces, especially for Sadness vs. Anger and Happiness vs. Fear. Regarding the selected principal appearance and motion for the other pairs of expression, see example images in Fig. 11, giving a visual impression of selected slices for each pair of expressions.

4.4. Comparison of classifiers

Neural network architectures, for example Radial Basis Network (RBN) and Generalized Regression Neural Networks (GRNN), have been successfully exploited in various pattern recognition problems (Sarshar et al., 2001; Rubiyah et al., 2002). In this subsection, we utilized those classifiers after slices are selected for every class pair, and compare their performance to that obtained with SVM. Fig. 12 shows the results.

It can be seen SVM outperforms the other classifiers in our problem. RBN works similarly to GRNN, but a bit better. Especially when the number of selected slices increases, SVM provides stable results, while RBN and GRNN do worse. This might be caused by the variation of data, like illumination and skin color changes. The GRNN characteristic that it does not optimize the separating hyperplane (Principe et al., 1999) makes it hard to deal with the classifications of very diverse data. Moreover, when the number of slices is increasing, the larger number of variables (longer feature vectors) makes the function approximation of GRNN less accurate.

4.5. Comparison with other methods

So far most of the research on facial expression analysis has considered static images. However, there is a growing interest in exploiting video sequences for this purpose. For example, Yeasin et al. (2004) used the horizontal and vertical components of the flow as features. At the frame level, the k-NN rule was used to derive a characteristic temporal signature for every video sequence. At the sequence level, discrete Hidden Markov Models (HMMs) were trained to recognize the temporal signatures associated with each of the basic expressions. This approach cannot deal with illumination variations, however. Aleksic and Katsaggelos (2006) proposed facial animation parameters as features describing facial expressions, and utilized multi-stream HMMs for recognition. The system is complex, making it difficult to perform in real-time. Buenaposada et al. (2008) proposed to use the deformation parameters provided by a dense and efficient appearance-based face tracker. A facial expression is represented by a set of samples that model a low dimensional manifold in the space of deformations generated by the tracker parameters.

Table 6 compares our method with other static and dynamic analysis methods in terms of the number of people (PN), the number of sequences (SN) and expression classes (CN), with different measures, providing the overall results obtained with the Cohn–Kanade facial expression database. We can see that with the experiments on the greatest number of people and sequences, our results are the best ones compared not only to the dynamic methods but also to the static ones. Please note that in our earlier work (Zhao and Pietikäinen, 2007), a combination of VLBP and LBP-TOP features provided a very good performance of 95.19%, but in that study the faces were localized using manually detected eyes. In Table 6, the second last row gives the result obtained with the original LBP-TOP features when faces were localized with automatically detected eyes. We can see from the last row that the approach proposed in this paper provides the best performance.

4.6. Experiments for videos taken from a web camera

To evaluate the performance in real-world environments, a simple web camera is used to capture videos with a resolution of 320 by 240 pixels in an office environment. Subjects were about 0.5–1 meters in front of the camera. The face was detected automatically using boosted Haar-like features and the size of the cropped faces was around 40 by 50. Fig. 13 demonstrates results taken from an example image sequence, showing the detected faces and classified expressions for each case.

5. Conclusion

In this paper, we extend the LBP-TOP features to multi-resolution spatiotemporal space and use them for describing facial

---

**Table 6**

Comparison with different approaches on Cohn–Kanade database.

<table>
<thead>
<tr>
<th></th>
<th>PN</th>
<th>SN</th>
<th>CN</th>
<th>Dynamic</th>
<th>Measure</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shan et al. (2005a)</td>
<td>96</td>
<td>320</td>
<td>7(6)</td>
<td>N</td>
<td>10-Fold</td>
<td>88.4 (92.1)</td>
</tr>
<tr>
<td>Bartlett et al. (2003)</td>
<td>90</td>
<td>313</td>
<td>7</td>
<td>N</td>
<td>10-Fold</td>
<td>86.9</td>
</tr>
<tr>
<td>Littlewort et al. (2004)</td>
<td>90</td>
<td>313</td>
<td>7</td>
<td>N</td>
<td>Leave-one-subject-out</td>
<td>91.8</td>
</tr>
<tr>
<td>Chen et al. (2008)</td>
<td>--</td>
<td>--</td>
<td>7</td>
<td>N</td>
<td>--</td>
<td>93.1</td>
</tr>
<tr>
<td>Tian (2004)</td>
<td>97</td>
<td>375</td>
<td>6</td>
<td>N</td>
<td>--</td>
<td>93.8</td>
</tr>
<tr>
<td>Yeasin et al. (2004)</td>
<td>97</td>
<td>--</td>
<td>6</td>
<td>Y</td>
<td>Five-fold</td>
<td>90.9</td>
</tr>
<tr>
<td>Aleksic and Katsaggelos (2006)</td>
<td>90</td>
<td>284</td>
<td>6</td>
<td>Y</td>
<td>Leave-one-subject-out</td>
<td>93.66</td>
</tr>
<tr>
<td>Buenaposada et al. (2008)</td>
<td>92</td>
<td>333</td>
<td>6</td>
<td>Y</td>
<td>Two-fold</td>
<td>91.44</td>
</tr>
<tr>
<td>LBP-TOP$_{8,4,2}$</td>
<td>97</td>
<td>374</td>
<td>6</td>
<td>Y</td>
<td>Two-fold</td>
<td>93.85</td>
</tr>
<tr>
<td>Ours</td>
<td>97</td>
<td>374</td>
<td>6</td>
<td>Y</td>
<td>Two-fold</td>
<td>93.85</td>
</tr>
</tbody>
</table>
expressions. AdaBoost is utilized to learn the most discriminative slices not just blocks from different resolutions and the new pair-boosting strategy is presented to learn what are the discriminative spatial and temporal features between a specific class pair. Our approach makes the feature selection flexible and can determine not only the location (obtained from block-based learning) and resolution of those discriminative features, but also the appearance and type of motion. This could also give some clues for psychologists to analyze the expressions accordingly. Moreover, compared to the method based on boosting for all classes, a class pair learning approach can learn more specific features, improving especially the discrimination ability of relatively similar pairs.

In the experiments, the proposed method works better than commonly used block-based method. Using one-against-one expression learning, the selected features emphasize more on expression-related features for specific two expression classes. Those can help the psychologists to analyze the expressions. For example, the more distinguishable features for sadness and anger focus on the horizontal motion of the side of the nose and the mouth corner, and vertical motion of the eyebrows. For happiness and fear, learned features focus on the horizontal motion in forehead, the area between eyes and between nose the mouth, and the appearance on the forehead and the mouth corner. Evaluation with the videos taken from a simple web camera in an office environment shows the effectiveness of the proposed method in real-time facial expression recognition.

Facial expression is just one example of possible applications where the proposed algorithm could be used. Some other potential applications, such as visual speech recognition and activity recognition, are under investigation.

Acknowledgements

This work was supported by the Academy of Finland. The authors would like to thank Dr. Jeffrey Cohn for providing the Cohn–Kanade facial expression database used in experiments.

References


