Facial Expression Recognition Using Histogram Variances Faces

Ruo Du[1], Qiang Wu[1], Xiangjian He[1,2], Wenjing Jia[1] and Daming Wei[2]
[1]University of Technology, Sydney, 15 Broadway, Ultimo, NSW 2007, Australia
[2]University of Aizu, Japan
{ruodu, wuq, sean, wejia}@it.uts.edu.au
dm-wei@u-aizu.ac.jp

Abstract

In human's expression recognition, the representation of expression features is essential for the recognition accuracy. In this work we propose a novel approach for extracting expression dynamic features from facial expression videos. Rather than utilising statistical models e.g. Hidden Markov Model (HMM), our approach integrates expression dynamic features into a static image, the Histogram Variances Face (HVF)*, by fusing histogram variances among the frames in a video. The HVFs can be automatically obtained from videos with different frame rates and immune to illumination interference. In our experiments, for the videos picturing the same facial expression, e.g., surprise, happy and sadness etc., their corresponding HVFs are similar, even though the performers and frame rates are different. Therefore the static facial recognition approaches can be utilised for the dynamic expression recognition. We have applied this approach on the well-known Cohn-Kanade AU-Coded Facial Expression database then classified HVFs using PCA and Support Vector Machine (SVMs), and found the accuracy of HVFs classification is very encouraging.*

1. Introduction

Human facial expression is able to disclose human's emotions, moods, attitudes and feelings etc.. Recognising expressions can help computer learn more about human's mental activities and react more sophisticatedly, therefore it has enormous potentials in human-computer interaction (HCI). Explicitly, the expressions are some facial muscular movements comparing to neutral face. Six basic emotions (happiness, sadness, fear, disgust, surprise and anger) were defined in the Facial Action Coding System (FACS) [3]. Each of these six basic emotions has a uniquely corresponding facial expression. FACS consists of 46 action units (AU) which depict basic facial muscular movements. Basically, how to capture the expression features precisely is vital for expression recognition. Getting expression features can be divided into two categories: spatial and spatio-temporal approaches. In spatial approaches, expression features extracted from a static face image are utilised for expression classification. Spatio-temporal models dynamic features and computes the models' parameters through statistics of observations.

For spatial approaches, Feng et al. [4][21] used the face Local Binary Patterns (LBP) [10][9] textures as expression features and recognised facial expressions using linear programming (LP). Littlewort et al. [8] utilised facial features based on static images for classification. They detected faces in face pictures and rescaled them to 48×48 pixels. The rescaled facial images were convolved with a bank of Gabor filters in order to obtain their Gabor magnitude representation. Then they performed feature selection using AdaBoost [5] and classification using Support Vector Machines (SVMs) [2][16][17]. For spatio-temporal approaches, Maja et al. [14][15][19] detected AUs by using individual feature GentleBoost [20] templates built from Gabor wavelet features and tracked temporal AUs based on Particle Filter (PF). Then the SVMs [2][16][17] was applied for classification. Petar et al. [1] treated the facial expression as a dynamic process and proposed that the performance of an automatic facial expression recognition system could be improved by modeling the reliability of different streams of facial expression information utilising multistream Hidden Markov Models (HMMs) [1]. The spatial approaches do not model the dynamics of facial expressions and are often disturbed by difference of facial appearances. Spatio-temporal approaches take into account modeling dynamic features but the model parameters are often hard to be obtained accurately.

Our novel approach of expression feature extraction saves the dynamic features into a *Histogram Variances Face(HVF)* image by computing the texture histogram variances among the frames of a face video. The frame rates of the videos do not have to be the same. The Local Binary Pattern (LBP) [10][9] is employed to extract the face texture

for making the histogram variances be immune to illumination interference. The Earth Movers's Distance (EMD) [11] is used to measure the histogram distance for ensuring that the histogram variances are consistent with human's vision. The HVF images are similar if they belong to the same expression with similar durations, so that static facial recognition methods can be utilised for the dynamic expression recognition. We test the HVFs classification by Support Vector Machines (SVMs) [2][16][17] after Principal Component Analysis (PCA) [12][13] dimensionality reduction. The accuracy of HVFs classification is very encouraging and highly matches human's perception on original videos.

The rest of the paper is organised as follows. Section 2 describes the procedures of generating a HVF image from an input video based on *LBP* operator and *EMD*. Section 3 presents the dimensionality reduction using *PCA* and uses *SVMs* for training and recognition. Experimental results and discussion are presented in Section 4. Section 5 comes up with conclusions and discussion.

2. Histogram Variances Face (HVF)

The HVF image is a novel representation of the dynamic features in a face video. In general, the procedures of generating a HVF from a video can be summarised as follows, and the related techniques will be described in later subsections:

1. Automatically align faces in temporal direction by detecting fiducial points (the eyes) per frame.
2. Preprocess and texturise face images.
3. Break down each texturised image into $M \times N$ blocks and compute the histogram variance for each block in temporal direction.
4. Create a new $M \times N$ 8-bit grayscale image, i.e. a *Histogram Variances Face (HVF)*. Each pixel value corresponds to a block histogram variance.

2.1. Fiducial points detection and faces alignment

For different expression videos, normally the scales and locations of human faces in frames are various. To make all the HVFs have the same scale and location, it is necessary to detect the face fiducial points and cut the faces out in terms of fiducial points. Meanwhile, bilinear interpolation is used to make sure all the face images have the same size. To detect the fiducial points, we make use of a real-time face detection scheme based on Haar-Like feature classifier cascade and AdaBoost learning [18], called Viola-Jones face detector. It consists of a cascade of classifiers trained by the AdaBoost algorithm. Each classifier uses integral image filters, bases on Haar Basis functions and can be computed very fast at any location and scale. For each stage in the cascade, a subset of features is chosen using a feature selection procedure based on the AdaBoost. This face detection scheme detects and locates eyes positions on the Cohn-Kanade expression database [7] precisely and fast. And in our system, the eyes are the fiducial points used to cut and align the faces because in the frontal face image sequences, the positions of human's eyes determine the face position accurately. The faces in videos are cutted out according to eyes' position and are normalised to a fixed size in proportion with distance between eyes.

2.2. Preprocessing and LBP texturising

After getting aligned faces using Viola-Jones face detector[18], we then mask the areas outside an ellipse around each face and leave only the face area as the region of interest (ROI). Histogram equalisation in ROI is also applied to reinforce the gradient. Furthermore, the illumination variety in a video is another issue that could interfere with the histogram variance in the temporal direction. To overcome this, we employ the LBP operator shown in [10][9] to extract the texture of the masked faces, and hence eliminate the illumination interference.

Local Binary Pattern (LBP) describes the surroundings of a pixel by generating a bit-code from the binary derivatives of a pixel. The operator is usually applied to grayscale images and the derivative of the intensities. A typical form of the LBP operator takes the 3×3 surrounding of a pixel and generates a binary 1 if the neighbor of the centre pixel has larger value than the centre pixel. The operator generates a binary 0 if the neighbor is less than the centre. The eight neighbours of the centre can then be represented with an 8-bit unsigned integer. The LBP value is calculated using Equation 1 [10][9]:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad (1)$$

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0 \end{cases}$$

where *P* is the number of neighbors, *R* is the radius and g_c corresponds to the gray value of the center pixel of a local neighborhood. $g_p(p = 0, \ldots, P-1)$ correspond to the gray values of *P* equally spaced pixels on a circle of radius *R* that form a circularly symmetric set of neighbors. Figure 1 shows an example of an LBP operator.

2.3. Earth Mover's Distance for calculation of histogram variances

There are a number of approaches to compute the similarity between two histograms. Normally these approaches are divided into two categories: bin-to-bin and cross-bin. In

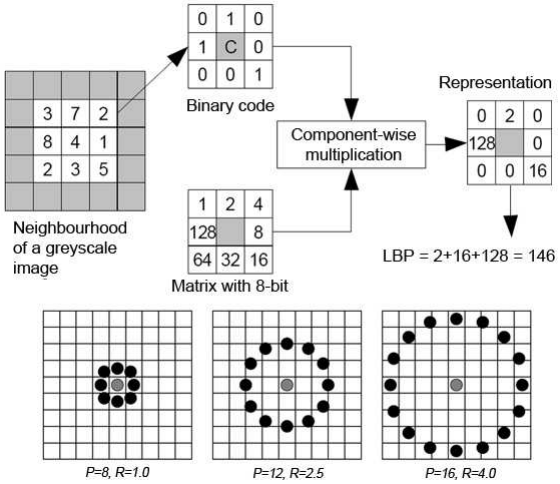

Figure 1. An example of computing LBP in a 3×3 neighborhood

our case, although a block may not have texture change during a video, its corresponding histograms in different frames are unlikely to keep the same because of the noise. It is quite often that the block histograms shift slightly according to Gaussian distributions. So the bin-to-bin approaches will not work well here because they are sensitive to the slight histogram shifting. Note that the histogram shifts caused by noise are invisible for human vision, so we should ignore these kinds of shifts. The Earth Mover's Distance (EMD) is a cross-bin approach and able to address the shift problem caused by noise because slight histogram shifts do not affect the EMD much. And the EMD is consistent with the human's vision because that two histograms will have greater EMD value if they look more differently in most cases. Another good cross-bin choice can be the *Quadratic Form Distance*, however it needs a positively definite parameter matrix which must be pre-defined. Our experiments prove that the EMD has the best performance.

Earth Mover's Distance (EMD) is a method to evaluate dissimilarity between two multi-dimensional distributions in some feature space where a distance measure between single features (called the ground distance) is given. It has the excellent capability of matching human's vision on histogram distribution differences. Basically, the EMD was formalized as the following linear programming problem. Let $P = \{(p_1, w_{p1}), \ldots, (p_m, w_{pm})\}$ be the first signature with *m* clusters, where p_i is the cluster representative and w_{pi} is the weight of the cluster; $Q = \{(q_1, w_{q1}), \ldots, (q_n, w_{qn})\}$ the second signature with *n* clusters; and $D = [d_{ij}]$ is the ground distance matrix where d_{ij} is the ground distance between clusters p_i and q_j. EMD is to find a flow $F = [f_{ij}]$, where f_{ij} is the flow between p_i and q_j, that minimizes the overall cost [11]

$$WORK(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij} \quad (2)$$

subject to the following constraints [11]:

1. $f_{ij} \geq 0; i \in [1, m], j \in [1, n]$,
2. $\sum_{j=1}^{n} f_{ij} \leq w_{pi}; i \in [1, m]$,
3. $\sum_{i=1}^{m} f_{ij} \leq w_{qj}; j \in [1, n]$ and
4. $\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = min\left(\sum_{i=1}^{m} w_{pi}, \sum_{j=1}^{n} w_{qj}\right)$

Constraint *1* allows moving "supplies" from *P* to *Q* and not vice versa. Constraint *2* limits the amount of supplies that can be sent by the clusters in *P* to their weights. Constraint *3* limits the clusters in *Q* to receive no more supplies than their weights; and constraint *4* forces to move the maximum amount of supplies possible. This amount is called the total flow. Once the transportation problem is solved, and the optimal flow *F* has been found, the EMD is defined as the resulting work normalized by the total flow [11]:

$$EMD(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \quad (3)$$

The normalization factor is the total weight of the smaller signature because of constraint *4*. This factor is needed when the two signatures have different total weights, in order to avoid favoring smaller signatures. In general, the ground distance d_{ij} can be any distance and will be chosen according to the problem in question.

We employ EMD to measure the distance between two histograms when calculating histogram variances in the temporal direction. In our case, p_i and q_j are the grayscale pixel values, which are in $[0, 255]$. w_{pi} and w_{qj} are the pixel distributions at p_i and q_j respectively. The ground distance d_{ij} that we choose is the square of euclidean distance between p_i and q_j, i.e., $d_{ij} = (p_i - q_j)^2$.

2.3.1 Procedures of calculating histogram variances

1. Suppose a sequence consists of *P* face texture images, firstly break down each image evenly into $M \times N$ blocks, denoted by $B_{x,y;k}$, where *x* is row index, *y* is column index and *k* is the *k-th* frame in the sequence. Calculate every gray-value histogram of $B_{x,y;k}$, denoted by $H_{x,y;k}$, where $x = 0, 1, \ldots, M-1; y = 0, 1, \ldots, N-1; k = 0, 1, \ldots, P-1$.

2. Calculate the histogram variance $var(x, y)$:

$$var(x, y) = \frac{1}{P} \sum_{k=0}^{P-1} EMD(H_{x,y;k}, \mu_{x,y}), \quad (4)$$

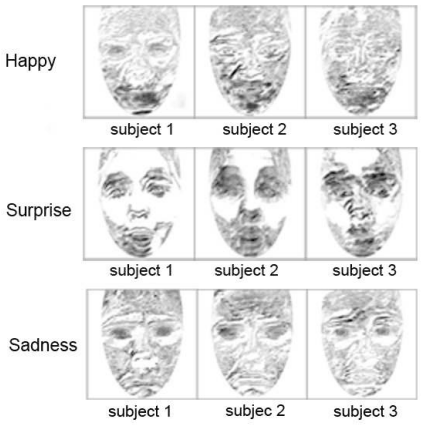

Figure 2. Examples of HVF images

where $\mu_{x,y}$ is the mean histogram

$$\mu_{x,y} = \frac{1}{P}\sum_{k=0}^{P-1} H_{x,y;k} \tag{5}$$

and $EMD(H_{x,y;k}, \mu_{x,y})$ is the Earth Mover's Distance between $H_{x,y;k}$ and $\mu_{x,y}$.

3. Construct an $M \times N$ 8-bit grayscale image as our HVF. Suppose that $hvf(x,y)$ denotes the pixel value at coordinate (x,y) in an HVF image:

$$hvf(x,y) = 255 - \left\lfloor \frac{255 * var(x,y)}{MAX(var(x,y))} \right\rfloor \tag{6}$$

And $hvf(x,y) = 255$ i.i.f $hvf(x,y) > threshold$. Figure 2 shows some HVF examples extracted from happiness, surprise and sadness videos respectively.

2.3.2 Computing histograms of various-size blocks

Whether the different block sizes affect our HVFs recognition is one of the questions we are going to answer in this paper. Hence we must get the histograms for various-size blocks. To make the histogram computation more efficient, we get the bigger-size histograms by adding small-size ones.

Suppose $H(\alpha)$ denotes the histogram vector with respect to image area α, there is

$$H(\alpha) + H(\beta) = H(\alpha \cup \beta), \tag{7}$$

so if $H_{x,y;k}(\gamma,\eta)$ denoted the histograms of $\gamma \times \eta$ pixels block at *x-th* row and *y-th* column in frame *k*, then

$$H_{x,y;k}(a\gamma, b\eta) = \sum_{i=0}^{a-1}\sum_{j=0}^{b-1} H_{ax+i,by+j;k}(\gamma,\eta), \tag{8}$$

where $a, b, \gamma, \eta \in N^+$. We only obtain all histograms with size 3×3 in our experiments, then the size 6×6 and 12×12 histograms can be computed fast and easily through Equation 8.

3. Classifying HVF images using PCA+SVMs

HVF records the dynamic features of the expression. As we can see in Figure 2, for the expressions of happiness, surprise and sadness, the homogeneous HVFs look similar and HVFs belonging to different expressions have their own unique features. To verify the performance of HVF image's features, we just utilise the typical facial recognition technologies PCA+SVMs, which have proven to be very well suitable for classification tasks such as facial recognition.

3.1. PCA dimensionality reduction

In experiments, the all pixel values of an HVF image construct an $n \times 1$ column vector $z_i \in R^n$, and an *n* by *l* matrix $Z = \{z_1, z_2, \ldots, z_l\}$ denotes the training set which consists of *l* sample HVF images. The PCA algorithm finds a linear transformation orthonormal matrix $W_{n\times r}(n >> r)$, projecting the original high *n*-dimensional feature space into a much lower *r*-dimensional feature subspace. x_i denotes the new feature vector:

$$x_i = W^T \cdot z_i \ \ (i = 1, 2, \ldots, l). \tag{9}$$

The columns of matrix *W* called eigenfaces [13], which are the r eigenvectors corresponding to the r largest eigenvalues of the scatter matrix *S*:

$$S = \sum_{i=1}^{l}(z_i - \mu)(z_i - \mu)^T \tag{10}$$

where μ is the mean image of all HVF samples and $\mu = \frac{1}{l}\sum_{i=1}^{1} z_i$.

3.2. SVMs training and recognition

SVMs [2][16][17] is an effective supervised classification algorithm and its essence is to find a hyperplane that separates the postive and negative feature points with maximum margin in the feature space. Very likely the real-world problems are not linearly separable, in that case SVMs map the original input space using 'kernel' functions into a higher dimenional space where the feature points are linearly separable.

Suppose α denotes the Lagrange parameters that describe the separating hyperplane ω in SVM. Finding the hyperplane that maximises the margin between positive and negative data set involves getting the nonzero solutions α_i of a Lagrangian dual problem, which is a quadratic programming problem and is solvable. Once we find all α_i and

given a labeled training set $\langle x, y\rangle$, the decision function can be as follow:

$$f(x) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(x, x_i) + b\right) \quad (11)$$

Where *b* is the bias of the hyperplane and *l* is the number of training samples, y_i is the label of train data, x_i is the vector of PCA projection coefficients of HVFs, $K(x, x_i)$ is the 'kernel mapping' and here as we use linear SVMs, therefore

$$K(x, x_i) = \langle x, x_i\rangle \quad (12)$$

$\langle x, x_i\rangle$ means the dot product of *x* and x_i.

Since the SVM is basically a two-class classification algorithm, here we adopt the pairwise classification (one-versus-one) for multi-class. In pairwise classification there is a two-class SVM for each pair of classes to separate members of one class from members of the other. Specifically, there are maximum $C_6^2 = 15$ two-class SVM classifiers are trained for the classification of six sorts of expressions. When recognising a new HVF image, all the $C_6^2 = 15$ two-class classifiers are applied to the testing HVF and the winner class is the one that takes the most votes.

4. Experiments

4.1. Dataset

Our experiments adopted the Cohn-Kanade AU-Coded Facial Expression Database [7]. This database consists of 97 university students ranging in age from 18 to 30 years. 65% were female, 15% were African-American, and 3% were Asian or Latino. Videos in this database were recoded using a camera located directly in front of the subject. Subjects were instructed by an experimenter to perform a series of 23 facial expressions. Subjects began and ended each display with a neutral face. Before performing each display, an experimenter described and modeled the desired display. Image sequences from neutral to expression apex were digitized into 640 by 480 pixel arrays with 8-bit precision for grayscale values.

We selected 31 subjects randomly from the database, each subject has up to 6 expressions (image sequences), the total number of sequences is 169, which means 169 HVFs were generated. The image sequences belong to the same expression have the similar duration but their frame rates are different. And for a certain expression, we fed around 80% HVFs to PCA+SVMs training process and the classifiers were later applied to all HVFs.

4.2. Preferences for HVFs generation

The faces of selected subjects were detected and cut out. Then these faces were resized to 300×300 pixels and aligned. To eliminate illumination interference, we used a 3×3 neighborhood with radius 1 for the LBP operator. As to the ground distance for EMD, we adopted the square of *Euclidean* distance between two pixel values. The reason for choosing *Euclidean* distance here is because for human's vision, the more difference of pixel value distribution between two image histograms causes the more distinction of the two images. The final data dimensions were reduced to 95 after PCA operation, and for the linear SVMs, our penalty parameter *C* is 8.

Moreover, to check the influence of different block segmentations, we chose the block's sizes as 3×3, 6×6 and 12×12 pixels , namely each texture image was broken down into 100×100, 50×50 and 25×25 blocks respectively. Because the blocks' histogram variance in the temporal direction becomes a pixel value in HVF, thus our HVF sizes are 100×100, 50×50 and 25×25 pixels as well.

	100^2 blocks		50×50 blocks		25×25 blocks	
	Reco.	FPR	Reco.	FPR	Reco.	FPR
HA	96.6%	3.3%	100%	3.3%	100%	3.3%
SU	96.7%	3.4%	96.7%	0.0%	96.7%	0.0%

Table 1. Recognition rates of happy and surprise HVFs.

4.3. Training and recognition

For supervised learning, the training data (HVFs) need to be labeled with specific classifications before training. Since Cohn-Kanade database only contains the AU-Coded combinations for image sequences instead of expression definitions (i.e. surprise, happy, anger etc.), we need to label each HVF with an expression definition manually according to FACS before feeding it to SVMs. In terms of human perception, we are quite confident to recognise original image sequences of happiness and surprise. Therefore the training data for these two classes can be labeled with high correction. This implies that these two expressions have evidently unique features. Our experimental results (Table 1) testifies this point with high HVFs recognition rate, where *FPR* is the *false positive rate*.

When we were labeling HVFs of *anger*, *disgust*, *fear* and *sadness*, nearly half of them were very challenging to be attached the convincing classifications, according to neither AU-Coded combinations nor human's perception on original image sequences, especially for *anger* and *sadness*. From the AU-Code of FACS perspective, AU-Coded prototypes in FACS (2002 version) [7] are overlapping for these expressions. And from human's vision perspective, one expression of a person may be reflected by several different sequences of images and one sequence of images is also often interpreted as various expressions. An investigation [6] about facial expression recognition by human discloses that

	HA	SU	AN	DI	FE	SA
Recog(%)	97.8	79.3	55.9	60.2	36.8	46.9

Table 2. A recent investigation of facial expression recognition by human in [6].

	100^2 blocks		50×50 blocks		25×25 blocks	
	Reco.	FPR	Reco.	FPR	Reco.	FPR
HA	96.6%	0.0%	100%	0.0%	100%	0.0%
SU	86.7%	3.3%	90.0%	1.7%	90.0%	1.7%
AN	96.8%	6.8%	96.8%	5.1%	96.8%	5.1%
HA	89.7%	0.0%	96.6%	0.0%	96.6%	0.0%
SU	83.3%	7.0%	90.0%	5.3%	90.0%	5.3%
DI	85.7%	13.5%	89.3%	6.8%	89.3%	6.8%
HA	93.1%	1.9%	96.5%	0.0%	96.5%	0.0%
SU	90.0%	7.7%	93.3%	3.8%	90.0%	3.8%
FE	86.9%	10.1%	91.3%	5.1%	86.9%	8.5%
HA	93.1%	1.7%	96.5%	0.0%	96.5%	0.0%
SU	90.0%	3.5%	93.3%	3.5%	93.3%	3.5%
SA	89.2%	8.4%	92.8%	5.1%	89.2%	6.7%

Table 3. Recognition rates of happy and surprise versus other sorts of HVFs.

compare to the expressions of *happy* and *surprise*, the expressions of *anger*, *fear*, *disgust* and *sadness* are much more difficult to be recognised by people. (see Table 2).

After trying our best to manually label the expressions under above circumstances, we conducted the following experiments:

1. Feed *happy*, *surprise* and *anger* HVFs into the SVMs. For these three sorts of expressions, we trained $C_3^2 = 3$ two-class classifiers (i.e. happy-surprise, surprise-anger and anger-happy classifiers) and test HVFs using majority voting. Likewise, we keep *surprise* and *anger* unchanged but substitute *anger* with *disgust*, *fear* and *sadness* respectively, and then conduct the same training and testing. The results are displayed in Table 3.

2. Put *anger*, *disgust*, *fear* and *sadness* in one group. For these four tough expressions, we train a set of classifiers which has $C_4^2 = 6$ two-class classifiers and test new HVFs using majority voting. Table 4 shows our results.

3. Put all of the HVFs together, train $C_6^2 = 15$ two-class classifiers. Use this set of classifier to recognise all of the HVFs by voting. We obtain the experimental results as shown in Table 5.

4.4. Discussion

1. From Table 1 we can see that both happy and surprise HVFs have very high recognition rates, e.g., happy HVFs reach amazing 100% recognition rate with only 3.3% false positive rate (FPR). They are also quite distinguishable from the rest HVFs according to Table 3. These results coincide with our observations on the original image sequences, as human can also easily identify the original happy and surprise sequences from Cohn-Kanade database. This fact confirms that the HVFs preserve the dynamic features well.

2. From Table 4, the recognition rates for anger, fear, disgust and sadness HVFs are much lower. This reflects the challenges that we have encountered when labeling the training data (nearly half of the training data in these four expressions are not convincing for us to label a class because of expression features entanglement). An investigation of facial expression recognition by human [6] also indicates that human is not sensitive to recognise the anger, fear, disgust and sadness expressions. This fact is exactly embodied in our HVFs recognition results.

3. Table 5 shows the recognition results when all six expressions were fed to SVMs for training, we can see that happy and surprise HVFs still stand out and the rest ones are hampered by the entanglement of features. Taking into account our difficulties for labeling the training data, the recognition rates in Table 5 make sense.

4. The frame rate of videos and the faces location in frames do not affect our experimental results evidently, but the durations of the expressions have to be similar, e.g. from neutral to apex. Moreover, the size of block

	100^2 blocks		50×50 blocks		25×25 blocks	
	Reco.	FPR	Reco.	FPR	Reco.	FPR
AN	74.1%	12.6%	77.4%	12.6%	70.9%	13.9%
DI	78.6%	12.1%	78.6%	10.9%	75.0%	13.4%
FE	69.5%	8.0%	73.9%	8.0%	69.5%	8.0%
SA	67.8%	3.6%	67.8%	2.4%	67.8%	3.6%

Table 4. Recognition rates of anger, disgust, surprise and sadness HVFs

	100^2 blocks		50×50 blocks		25×25 blocks	
	Reco.	FPR	Reco.	FPR	Reco.	FPR
HA	93.1%	0.0%	96.5%	0.0%	93.1%	0.0%
SU	90.0%	0.0%	93.3%	0.0%	90.0%	0.0%
AN	80.6%	10.1%	80.6%	8.6%	80.6%	10.1%
DI	75.0%	8.5%	82.1%	7.8%	75.0%	8.5%
FE	78.2%	3.4%	78.2%	2.7%	73.9%	3.4%
SA	75.0%	0.0%	71.4%	0.0%	71.4%	1.4%

Table 5. Recognition rates of all sorts of HVFs

is not critical to our results, but generally, the 50×50 blocks segmentation has the best performance in our experiments.

5. Conclusion

Our experiments demonstrate HVF is an effective representation of the dynamic and internal features of a face video or image sequence. HVF is able to integrate well the dynamic features of a certain duration of expression into a static image through which the static facial recognition approaches can be utlised to recognise the dynamic expressions. The application of HVFs fills the gap between the expression recognition and facial recognition.

Acknowledgements. Funding for this work was provided by UTS ECRG Research Grant (No. 2006000775).

References

[1] P. S. Aleksic and A. K. Katsaggelos. Automatic facial expression recognition using facial animation parameters and multistream hmms. In *INFORMATION FORENSICS AND SECURITY*, volume 1, pages 3–11, 2006.

[2] B. B., G. I., and V. Vapnik. An training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[3] P. Ekman and W. Friesen. Facial action coding system. In *Palo Alto, CA: Consulting Psychologists Press*, 1978.

[4] P. M. Feng X and H. A. Facial expession recognition with local binary partterns and linear programming. In *Pattern Recognition and Image Analysis*, volume 15(2), pages 546–548, 2005.

[5] Y. Freund and R. E.Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Journal of Computer and System Sciences*, volume 55, 1997.

[6] T. Jinghai, Y. Zilu, and Z. Youwei. The contrast analysis of facial expression recognition by human and computer. In *ICSP*, pages 1649–1653, 2006.

[7] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France*, pages 46–53, 2000.

[8] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. In *CVPR*, 2004.

[9] T. Ojala, P. M, and T. M$\ddot{a}$enp$\ddot{a}\ddot{a}$. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. In *PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, volume 24, pages 971–987, 2002.

[10] T. Ojala, M. Pietik$\ddot{a}$inen, and T. M$\ddot{a}$enp$\ddot{a}\ddot{a}$. Gray scale and rotation invariant texture classification with local binary patterns. In *ECCV*, pages 404–420, 2000.

[11] Y. RUBNER, C. TOMASI, and L. J. GUIBAS. The earth mover's distance as a metric for image retrieval. In *International Journal of Computer Vision*, volume 40(2), pages 99–121, 2000.

[12] M. Turk and A. Pentland. Eigenfaces for recognition. In *Journal of Cognitive Neuroscience 3*, volume 1, pages 71–86, 1991.

[13] M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[14] M. Valstar and M. Pantic. Fully automatic facial action unit detection and temporal analysis. In *Computer Vision and Pattern Recognition Workshop*, volume 17-22 June, page 149, 2006.

[15] M. Valstar, I. Patras, and M. Pantic. Facial action unit detection using probabilistic actively learned support vector machines on tracked facial point data. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[16] V. Vapnik. The nature of statistical learning theory. In *Springer-Verlag*, 1995, ISBN 0-387-98780-0.

[17] V. N. Vapnik. Statistical learning theory. wiley interscience. In *Wiley Interscience*, 1998.

[18] P. Viola and M. J. Jones. Robust real-time object detection. In *ICCV*, 2001.

[19] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *Systems, Man and Cybernetics(ICSMC)*, volume 2, pages 1692–1698, 2005.

[20] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *SMC*, pages 1692–1698, 2005.

[21] Y. Zilu and F. Xieyan. Combining lbp and adaboost for facial expression recognition. In *ICSP*, volume 26-29 Oct., pages 1461–1464, 2008.