Video-Based Facial Expression Recognition Using Hough Forest

SHIH-CHUNG HSU1, CH-TING HSU1, AND CHUNG-LIN HUANG1,2
1. Department of Electrical Engineering, National Tsing Hua University, Hsin-Chu, Taiwan.
2. Department of Applied Informatics and Multimedia, Asian University, Taichung, Taiwan.
E-mail: d9761817@oz.nthu.edu.tw, clhuang@asia.edu.tw

ABSTRACT
This paper introduces a new video-based facial expression recognition system. Facial expression analysis encounters two major problems: non-rigid shape deformation and person-specific facial expression appearance. Our method analyzes the video sequence to recognize facial expression and locate the temporal apex of the facial expression by using modified Hough forest and minimizing the influence of person-specific facial expression appearance. Our contributions are (1) random sampling 3-D accumulated spatial-temporal motion map to generate video patches, (2) proposing the correlation filtering for more effective Hough voting, and (3) recognizing and locating the apex of the facial expression. The experimental results show that the performance of our method is better than the other face expression recognition methods.

Keywords: Facial Expression Recognition, Hough Forest, Random Forest.

1. INTRODUCTION

Human emotion is usually understood by both of the facial and the vocal clues [1]. Many researchers focus on analyzing the facial clue (or facial expression) to categorize human emotion. Facial expression displays the internal emotion of human being, but it varies from person to person [2]. In [40], an authentic facial expression database is created based on spontaneous emotions. Ekman [3] shows the universality in facial expressions of happiness, sadness, anger, fear, surprise, and disgust. To code facial expressions, Ekman et al. [4] develop the Facial Action Coding System (FACS) in which the facial movement is described by action units (AUs) for encoding the motion of various facial muscles. In [41, 42], they propose facial expression recognition by using AUs detection. Facial Animation Parameters (FAPs) can also be used as features for facial expression classification [37].

Human facial expression understanding has been an indispensable and challenging research topic which can be categorized as the image-based and the video-based methods [5]. The video-based methods [6–9, 37–39, 43] attempt to recognize the facial expressions based on the facial features of facial movement. They consider the facial expression as dynamic, which evolves over time from the onset, the apex, and the offset. Onset is defined as the time from the start of the expression episode to the peak of facial movement. Apex is the amount of time the expression held at the peak, and offset is the time from the fading of the expression until it stops. The image-based methods [10, 11, 44–46] take only one shot to capture the image characteristics at the apex of facial expression. They ignore dynamic features and encounter two main issues: non-rigid morphing and person-specific appearance.

Existing facial expression analysis methods can also be divided into geometric-based and appearance-based approaches. The appearance-based methods detect the edge gradient and generate histogram of gradient orientation (HOG) [4] for facial expression recognition. The other descriptions of texture variation resistant to brightness variation are Local Binary Patterns (LBP) [10, 11], Haar-like [12] and Gabor Wavelet [13–16]. These features are texture variation discriminative which can be used to recognize the AU and facial expression [28]. The geometric-based methods find a number of facial feature points (at eyebrows, eyes, and mouth) as the key points. The variations of these key points are used for the facial expression recognition. The Gabor feature based boosted classifiers [15] is used to detect the facial feature points. By analyzing temporal changes of these feature points, it can recognize the AUs and the temporal facial expression [9]. The other researchers use Active Appearance Model...
Active Shape Model (ASM) [18] to detect the key points and trace these key points to record their displacement which is person independent. However, the locations of these key points are not precise because of head displacement.

In [19], they also categorize the facial expression recognition into template-based, and rule-based, and learning-based methods. The template-based method [20] extracts the differential-AAM features (DAFs) of the face images. To recognize the facial expression, it computes the difference between the input image and reference image by using the directed Hausdorff distance. In [43], template matching and HMM are applied for recognizing facial expression. The rule-based method [21] finds the mapping rules between AUs and expressions. The learning-based method [6, 10, 22] represents the face image by local binary patterns (LBP) [6], pyramid histogram of gradient (PHOG) [24], or local phase quantization (LPQ) [25] features. Then, it applies support vector machine (SVM) [23] or Linear Discriminant Analysis (LDA) for facial expression recognition. Recently, the Sparse Representation based Classifier (SRC) [44] is proposed to identify the facial expression.

This paper presents a video-based facial expression recognition method. First, it locates the face, nose and two eyes for face alignment. Different from previous video-based methods, it extracts the motion field and accumulates the motion to build the 3-D spatial-temporal map which is randomly sampled to generate 3-D video patches as the dynamic features. The 3-D video patch samples can be used to eliminate the influence of person-specific facial effects on expression recognition. Different from conventional Hough forest classifier [26, 27], it proposes the correlation filtering to find effective votes to recognize and locate the apex of the expression. The experimental results illustrate that it can recognize the facial expression and identify the apex of the expression more effectively.

2. FACIAL FEATURE EXTRACTION

Facial expression recognition method encounters two challenges [28]: non-rigid shape deformation and person-specific appearance. Here, we analyze the non-rigid morphing to eliminate the effects of person-dependent facial expressions. For different persons, the facial muscle will make the similar movement during facial expression. To extract the motion feature, we apply facial feature point detection, face registration, tracking, and motion analysis. We apply Viola-Jones face detector [29] to locate human face, and use Gabor-feature-based boosted classifier to locate the facial feature points.

A. Registration

We assume that the face is in a near-frontal pose and the eyes have the same inter-ocular distance in each image frame. Here, we may align the face images based on the facial feature points. The faces may be rotated and scaled so that we need to suppress the inter-sequence and intra-sequence variations of the faces. The registration techniques are applied to find the displacement field that registers each frame to a neutral, expressionless reference frame to ensure that the faces are fully aligned and the facial feature points are located at the same position. The original and the aligned faces are shown in Fig. 1. Similar to [7], we assume that the displacement of the pixels between two image frames is small so that we can suppress the intra-sequence and inter-sequence variations by apply simple affine registration. The head motion can be suppressed by applying the minimizing the squared sum of difference (SSD) between the previous reference frame and the registered current frame.

![Fig. 1. (a) The original face; (b) Aligned face; (c) Tilt face.](image)

B. Motion Extraction
After the registration, we may find the local facial motion field between two consecutive frames. Similar to [7], we use free-form deformation (FFD) model [30] to analyze the facial motion. Based on B-Spline [31], the FFD model is used to estimate the displacement field using a lattice of control points overlaid on the face image. Next, cubic B-splines are used to interpolate the motion field between the control points and generate a smooth continuous face deformation.

The control points are displaced so that a cost function describing the alignment of two consecutive images is minimized based on normalized mutual information [11]. In our implementation, we use the sum of squared difference (SSD) as the image alignment criterion. To find the new position of each point, we use a B-splice interpolation using the 16 closest neighboring control points. The displacements between the corresponding points in two consecutive frames can be found as the motion field. The motion between two consecutive frames is small and may not provide enough information. So, we accumulate the motion information for a temporal duration for pixel \( i \) defined as

\[

t^{\text{mag}}(x, y, t) = \sum_{t=t_i-1}^{t=t_i+D} \sqrt{u(x, y, t)^2 + v(x, y, t)^2}
\]

\[
\tilde{I}^x(x, y, t_i) = \sum_{t=t_i-1}^{t=t_i+D} u(x, y, t)
\]

\[
\tilde{I}^y(x, y, t_i) = \sum_{t=t_i-1}^{t=t_i+D} v(x, y, t)
\]

where \( u(x, y, t) \) and \( v(x, y, t) \) are the horizontal and vertical optical flow motion of pixel \( i \) at frame \( t = t_i \), and \( D \) is the temporal duration (e.g., \( D = 1 \)). The accumulated motion field \( F = \{ \tilde{I}^x, \tilde{I}^y, \tilde{I}^z \} \) consists of magnitude component, vertical component, and horizontal component.

C. Video Patch Sample

As shown in Fig. 3, the 3D video patches (20x20x3) are randomly sampled from the accumulated motion field \( F \). It is encoded as \( p_l(\cdot) = (p^{\text{mag}}, p^x, p^y, c_l, d_l, t_l) \), where \( p^{\text{mag}}, p^x \), and \( p^y \) indicate the three motion components of the patch sample (with center at pixel \( i \)), \( c_l \) is the expression label, \( d_l \) is 2D spatial displacement offset from the patch center to the centroid of image frame, \( t_l \) is temporal displacement offset to the video sequence center.

![Fig. 2. The sample at pixel \( i \) of the accumulated motion \( t^{\text{mag}} \)](image)

![Fig. 3. (a) The video patches selected from the video sequence. (b) The video patches.](image)
3. CLASSIFICATION USING HOUGH TREE

The facial expression recognition is based on the matching between the input 3D video patches and the pre-stored labeled video patches. Due to a large number of video patches, the matching process is very time consuming. To speed up the recognition process, Hough forest [26, 27] originated from the random forest and Hough transform is proposed.

A. Random Forest

Random forest [32] is an ensemble classifier consisting of many decision trees. Decision trees are commonly used to recognize a strategy most likely to reach a goal and then calculate the posteriori probabilities. The randomization consists of random selection of training data and random test selection of the division hypothesis at each non-leaf node. Random forests are constructed by a supervised training which involves tree construction and a best binary hypothesis of the non-leaf node that divides the training data for this node into two subsets. After training, a tree is constructed and a binary test is assigned to each non-lead node. During testing, a test sample passes down one of the trees and reaches a leaf node. The output is calculated by averaging the distributions at the reached leaf node.

B. Hough Forest

Hough forest is developed based on the concept of generalized Hough transform [33] which has been used to find the imperfect instances of objects by a voting procedure. This voting procedure is carried out in a parameter space, from which the object candidates are identified as local maxima in Hough space. The parameters of video patch are used to find the matching between the input video patch and its \( n \)-nearest neighbors which are labeled and pre-stored in the leaf node. For each input patch, the \( n \)-nearest neighbors found in the leaf node will cast the votes in Hough space. After Hough voting, the bucket in Hough space with maximal number of accumulated votes will be found which indicates the class and the temporal apex of the facial expression in the video.

In the training process, 3D video patches are randomly selected from the training videos. The sampled patches will be located at the discriminant portion on the face such as the eyes and mouth. The patches on the nose are less discriminant than the patches on the other places because the motion field in the nose region has little facial expression related information. So, we only sample the video patches around discriminant face region.

C. Hough Tree Training

Hough tree \( T_n \) is trained by a set of labeled 3D video patches. Each video patch is defined as \( p_i(z) = (p_i^{mag}, p_i^x, p_i^y, c_i, d_i, t_i) \). After Hough tree training, in each leaf node \( L \), we have a set of patches with labels and spatial/temporal displacements as \( D^L = \{c_i, d_i, t_i\} \). The leaf node \( L \) provides the probability \( P^L \) for each class \( c \), i.e. \( \sum_c P^L = 1 \). During the training process, each non-leaf node \( B \) is assigned a binary hypothesis \( h \) defined as

\[
  h_{a,r,q}^B(p_i) = \begin{cases} 
    0 & \text{if } p_i^a(r) < p_i^a(q) \\
    1 & \text{otherwise}
  \end{cases}
\]  

(2)

where \( \{a, r, q\} \) is the parameter set of hypothesis \( h \), \( r \) and \( q \) are two selected points in video patch \( p_i(z) \), \( a \) indicates the selected attribute (e.g., \( mag, x \), or \( y \)). The hypothesis \( h \) compares the selected attribute \( a \) of \( p_i(r) \) and \( p_i(q) \) to decide whether the video patch \( p_i(z) \) goes to the left or the right node.

Hough forest training begins at the root by choosing a hypothesis and then splits the training patches based on the selected hypothesis to generate the two child nodes. At each subsequent child node, the same procedure continues recursively, with each node being designated as a non-leaf node until the termination criteria is met. Upon termination, the remaining labeled patches with \( D^L = \{c_i, d_i, t_i\} \), and \( P^L \) are stored in the leaf nodes. The ideal binary hypothesis splits the patches with the minimal uncertainties of the class label and temporal center offsets. Here, we develop two measures to evaluate the uncertainty for a set of patches \( A = \{p_i(z)\} \). The first measure \( U_i \) defines the class uncertainty as
\[ U_1 = -1 \cdot \sum_c P_c \ln(P_c) \]  

where \( P_c \) is the proportion (or probability) of patches with class label \( c \) in the training set \( A \). \( U_1 \) defines the summation of the entropy of all classes which should be minimized. The second measure \( U_2 \) defines the maximal center offset uncertainty of all patches in different classes as

\[ U_2 = \max_c \left\{ \sum ||t^c - \bar{t}^c||^2 \right\} \]

where \( \bar{t}^c \) is the mean temporal offset of the video patches with class label \( c \) in class set \( A \). Using the Minimax algorithm, we need to minimize the maximal offset uncertainty.

At each non-leaf node during training, a pool of binary tests (or hypotheses) for node \( B \), \( \{ h^B \} \), are generated with randomly selected attribute values (i.e., \( a \), \( r \), and \( q \)). The set of training patches arriving at the non-leaf node will be evaluated by all binary hypotheses in the pool. Then, the uncertainty measures \( U_1 \) and \( U_2 \) are made for the training data set, and the best hypothesis for node \( B \) will be selected if either \( U_1 \) or \( U_2 \) is minimized. The \( k^th \) hypothesis satisfying the following minimization objective will be chosen as

\[ \arg\min_k \left\{ U_1(\{ A|h^B_k = 0 \}) + U_2(\{ A|h^B_k = 1 \}) \right\} \]

where \( k = \{a, r, q\} \) indicates the set of parameters to be selected for the hypothesis \( h \) of decision node \( B \), subscript * indicates the chosen uncertainty measure (\( U_1 \) or \( U_2 \)). In the training stage, we randomly select class uncertainty \( U_1 \) or offset uncertainty \( U_2 \) interleave throughout the tree growing.

In the training stage, we try a set of randomly selected hypotheses to separate the training data set into two sub-sets and then find the best hypothesis that makes the best separation. After the best binary decision (or hypothesis) is determined, it can be used to generate two child nodes. Then, we iteratively determine the best binary hypothesis for each child node to further divide the training sub-set in each child node. The child node will stop splitting and become a leaf node. The splitting terminates if one of the following three criteria occurs: (a) the depth of current child node is too deep, (b) the number of patches in each training sub-set is too small, and (c) the uncertainty measures are below certain constraints. The Hough tree training flow chart is shown in Fig. 4.

---

**Fig. 4** Hough tree training flow chart

### 4. EXPRESSION RECOGNITION
To recognize the facial expression, we extract the video patches from the test video. Each input patch through Hough tree will end up a certain leaf node. The pre-stored labeled patches will be retrieved to cast votes. Each patch casts one vote to the designated bucket in Hough space based on its expression label \(c\), spatial/temporal offset displacements \(d\) and \(t\). By comparing the accumulated votes in the buckets, we can recognize the correct facial expression and the expression apex.

### A. Hough Voting

Each input patch, \(p_i(t)\) will end up in leaf \(L\) of Hough tree \(T\) in which there are a set of similar labeled patches as \(D^L_c = \{c_j, d_j, t_j\}\), where \(c_j\) is the class label, \(d_j\) is the spatial offset, and \(t_j\) is the temporal offset. Let \(Q_c(t)\) be the random event corresponding to the possible existence of facial expression class \(c\) and centered at \(t\). The accumulated Hough votes in Hough space can be used to represent the class posterior probability of the input patch as

\[
P(Q_c(t)|p_i(t), T) = \sum_j P(Q_c(t)|c_j, d_j, t_j) P\left(c_j, d_j, t_j | p_i(t)\right) P\left(p_i(t)|p_j(t)\right)
\]

(6)

The summation \(\sum_j\) must satisfy the constraints as \(c_j = c\), \(|d_j - d_i| < R\), and \(t_j = t\). \(R\) indicates the radius of the related region. The class posterior probability of the input patch to Hough tree \(T\) can also be rewritten as

\[
P(Q_c(t)|p_i(t), T) = \sum_j P(Q_c(t)|c_j, d_j, t_j) P(c_j, d_j, t_j | p_i(t))
\]

(7)

The 1st term can be approximated as the Parzen-window estimate of \(D^L_c\) with the spatial offset \(d_j\), temporal offset \(t_j\), and class \(c_j = c\). The 2nd term can be approximated as \(P^L_c\) indicating the probability of the input patch belonging to class \(c\). We can rewrite Equ. (7) as

\[
P(Q_c(t)|p_i(t), T) = \left( \frac{1}{|D^L_c|} \sum_{t_j \in D^L_c} G((t' - t) - t_j) \right) \cdot P^L_c.
\]

(8)

where \(G(\cdot)\) is the 1-D Gaussian Parzen window function.

Before Hough voting, we propose the so-called correlation filter to invalidate some of the votes. Each patch in the leaf node may cast a valid vote only when it is close to the input patch. We apply the correlation filter to invalidate the votes from the similar patches in the lead nodes if they are outside the related region of the input patches as shown in Figure 5. The related region is defined as \(|d_j - d_i| < R\), where \(R\) is the radius of the related region which is determined experimentally. The correlation filtering increases the validity of the accumulated votes. Here, we rewrite Equ. (8) as

\[
P(Q_c(t)|p_i(t), T) = \frac{1}{|D^L_c|} \sum_{t_j \in D^L_c \cap |d_j - d_i| < R} G((t' - t) - t_j) \cdot P^L_c
\]

(9)

where \(G(\cdot)\) is the 1-D Gaussian Parzen window function. After correlation filtering, the accumulated votes in buckets are assumed all valid votes.

![Hough Tree](image)

**Fig. 5.** Correlation filter validates the video patches (in green color) stored in the leave node.
For Hough forest \( T \), we average over all the trees as

\[
P(Q_c(t) \mid p_i(\cdot), T) = \frac{1}{|T|} \sum_{n} P(Q_c(t) \mid p_i(\cdot), T_n)
\]

where \( T = \{ T_n \} \) and \(| T |\) indicates the number of Hough trees. Equ. (9) and Equ. (10) define the probabilistic vote of a single input patch for facial expression class \( c \). Votes from all the input patches selected from the test video sequence, \( S(t) \), are integrated into the accumulators in the temporal axis for different classes as

\[
V(t, c) = \sum_{p(y) \in S(t)} P(Q_c(t) \mid p_i(\cdot), T).
\]

### B. One-vs-one method

We may decompose the classification problem of six facial expressions into 15 binary-classification problems (i.e., happy vs. surprise, anger vs. fear, sad vs. disgust, etc.). By developing 15 binary Hough trees, we can describe the likelihood of six different classes for each input patch as

\[
P(Q_c(t) \mid p_i(\cdot), T) = \sum_{j \neq c} w_{cj} \cdot P(Q_{cej}(t) \mid p_i(\cdot), T_{cej})
\]

where \( T \) is a set of Hough trees as \( T = \{ T_{cej} \}, c = 1 \sim 6 \), and \( j = 1 \sim 6 \), \( T_{cej} \) is the binary-classification Hough tree for differentiating class \( c \) from class \( j \), and \( Q_{cej}(t) \) is the random event centered at \( t \) corresponding to the possible existence of facial expression of class \( c \) or \( j \). The weight \( w_{cj} \) for the likelihood generated from each Hough tree is defined as

\[
w_{cj} = \begin{cases} w_c, & P_c^L < P_j^L \\ w_c, & P_c^L \geq P_j^L \end{cases}
\]

where \( P_c^L \) or \( P_j^L \) indicates the posteriori probability of class \( c \) or \( j \), and the weights for each class \( c \) are normalized as \( \sum_j w_{cj} = 1 \). A positive weight \( w_c \) is assigned if the likelihood (or votes) of the designated class is larger than its counterpart, otherwise a negative weight \( -w_c \) is assigned. Different weight \( w_c \) is assigned to different class which is determined experimentally. For certain pair of ambiguous facial expressions such as angry vs. disgust, the weight will be smaller. Finally, all the weighted likelihoods are added for each class. The one with the largest accumulated weighted likelihood is identified as the correct facial expression \( C_{expression} \) as

\[
C_{expression} = \text{Argmax}_c P(Q_c(t) \mid p_i(\cdot), T).
\]

### C. Expression Validation

After voting, the testing video will be identified as one of the six facial expressions if the accumulated votes of the corresponding binary classifier will be much larger than the others. The number of votes also has to be larger than certain threshold for us to identify the location of the accumulator and recognize the facial expression. However, even if the testing video does not belong to one of the six facial expressions (e.g., neutral or undefined), one of the pre-defined six facial expressions with the largest accumulated weighted likelihood will still be found. Here, we set a threshold by averaging all the accumulated weighted likelihoods of all classifiers. If the largest accumulated weighted likelihood is larger than 1.3 of the average of all classifiers, then the identified expression is valid else it is neutral or undefined. The threshold is defined as

\[
\text{Max} P(Q_c(t) \mid p_i(\cdot), T) > 1.3 \cdot \frac{1}{|C|} \sum_{c \in C} P(Q_c(t) \mid p_i(\cdot), T).
\]

Based on Eq. (15), we may identify the most likely facial expression and the apex of facial expression.

### 5. EXPERIMENTAL RESULTS

Here, we show the experimental results and compare the results of our method with the others
based on the same test dataset. There are two different video datasets for continuous facial expression recognition: Cohn-Kanade+ AU-coded facial expression database (Cohn-Kanade+)\cite{34, 35} and MMI-Facial Expression Database (MMI)\cite{36}. We test our method by using these video data sets and compare our results with the others. Besides, we also create the third facial expression video dataset recorded from 10 individuals in our lab.

### A. Cohn – Kanade+ Dataset

The video dataset \cite{34, 35} is recorded from 210 people with ages ranging from 18 to 50. The gender ratio is that 69% female and 31% male. The racial distribution is 81% Caucasian, 13% African American, and 6% others. The resolution of each frame is 640\(\times\)490 or 640\(\times\)480. Each video starts from a natural expression to onset, and finally to apex. In the dataset, they label all the possible expressions for each video. In the first experiment, we choose the facial expression dataset from Cohn-Kanade+ database which has sufficient information and strongly indicates certain expression. Our purpose is to avoid selecting some videos that provide no indication of certain facial expression. The video dataset of six different facial expressions is shown in Table 1. We randomly selected one-third of the dataset for training. The rest two-thirds of the dataset are for testing.

**TABLE 1. Cohn-Kanade+ hand-labeled database.**

<table>
<thead>
<tr>
<th>Class</th>
<th>Angry</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happy</th>
<th>Sad</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>41</td>
<td>55</td>
<td>25</td>
<td>69</td>
<td>28</td>
<td>80</td>
</tr>
</tbody>
</table>

### B. MMI Dataset

The MMI video dataset \cite{36} is recorded from 19 males and females which includes different AUs and expressions. Each video starts from neutral to onset, and to apex and then come back to natural expression. The resolution of each frame is 720\(\times\)480. Each subject is filmed without prior training or head-motion limitation. The length of video is different for different subject that greatly increases the complexity of analyzing the facial expression of the videos in MMI dataset. Each video in the MMI dataset consists of the transitions: Neutral→Onset→Apex→Offset→Neutral. Fig. 6 shows three subjects, and each subject has his own way to express the angry expression. We may see that the way of showing the anger and the head motion changes from person to person. Table 2 shows the MMI dataset. We randomly select some of the MMI dataset for training which consists of 15 video sequence of each expression. The rest of the MMI data set is used for testing.

![Fig. 6. MMI dataset of different facial expressions.](image)

**TABLE 2. Different expression videos in MMI database**

<table>
<thead>
<tr>
<th>class</th>
<th>angry</th>
<th>disgust</th>
<th>fear</th>
<th>happy</th>
<th>sad</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>quantity</td>
<td>32</td>
<td>30</td>
<td>29</td>
<td>38</td>
<td>32</td>
<td>41</td>
</tr>
</tbody>
</table>

### C. Lab708 Dataset

We collect another facial expression video dataset by taking the facial expression videos of 10 graduate students. They have no priori guiding to make their facial expressions. However, they are
allowed to move their heads very slightly while making facial expressions. The resolution of each frame is 720×480. Each subject has been recorded 13 video sequence for each expression, and each video lasts 6~7 seconds. Totally, we have collected 780 video sequences. One-third of the dataset is used for training and the other two-thirds are for testing.

Fig. 7. The fear expression of four different subjects in our database.

D. Parameter Setting

In the experiments, we need to determine the size of the patch and the number of selected patches per frame. Here, we train 10 Hough trees and choose 10 videos to test different parameter settings. In the experiments, the patch sampling rate varies from 50 to 300 patches per frame and the patch size varies from 5×5 to 90×90. For fixed sampling rate, we can find the best patch sample rate and patch size. First, we fix the sample rate and try the different patch sizes. Then, we fix the patch size and vary the sampling rate. In the 1st experiment, we find that the patch size has a great influence on the recognition rate. The larger patch size will include more temporal information of the facial motion for higher recognition accuracy. However, if the patch size is larger than 20×20, then the recognition accuracy decreases. The recognition accuracy vs. the patch size is shown in Fig. 8.

Fig. 8. The recognition rate for different patch size.

We also illustrate the influence of the number of selected patch samples per frame for facial expression recognition as shown in Fig. 9. Larger number of video patch samples will generate more votes, the statistics will become more un-biased, and the recognition accuracy increases. However, more patch samples indicates more complicated voting process and more computation time. We find that the recognition accuracy will not increase once the number of video patch samples per frame is more than 200. In the 2nd experiment, we find that the higher sampling rate will induce more stable recognition rate, however the recognition time will also be increased. When the patch size is 20×20×3 and the sampling rate is 100 patches/frame, we have the best recognition rate and system performance. In the following experiments, we select 100 patches/frames and 20×20×3 patch size.
E. Experimental Results of Three Datasets

One-third of Cohn-Kanade+ dataset are for training, and the rest two-thirds are for testing. MMI dataset includes 5 video sequences for each expression. Similarly, we select one-third of MMI dataset and Lab 708 dataset for training and the rest for testing. For Cohn-Kanade+ dataset, we do not train our classifier by using the modified multi-voting Hough forest but using the conventional Hough forest. It is because the number of single-person video sequence is not sufficient for single Hough forest training. For MMI and Lab-708 dataset, the multi-voting Hough forest training can be applied. The video patch size is 20×20×3. In testing, the sampling rate is 100 video patches/frame. Each Hough forest consists of 5 or 3 Hough trees. We train an odd number of Hough trees for the final voting. The experimental results of the three datasets are shown in Tables 3, 4, and 5.

TABLE 3. Confusion Matrix for Cohn-Kanade+ dataset.

<table>
<thead>
<tr>
<th></th>
<th>angry</th>
<th>disgust</th>
<th>fear</th>
<th>happy</th>
<th>sad</th>
<th>surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>angry</td>
<td>0.75</td>
<td>0.1</td>
<td>0</td>
<td>0.13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>disgust</td>
<td>0</td>
<td>0.97</td>
<td>0</td>
<td>0.025</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>fear</td>
<td>0</td>
<td>0</td>
<td>0.88</td>
<td>0.055</td>
<td>0</td>
<td>0.055</td>
</tr>
<tr>
<td>happy</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
<td>0.97</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sad</td>
<td>0.10</td>
<td>0</td>
<td>0.10</td>
<td>0.80</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>surprise</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th></th>
<th>angry</th>
<th>disgust</th>
<th>fear</th>
<th>happy</th>
<th>sad</th>
<th>surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>angry</td>
<td>0.59</td>
<td>0.22</td>
<td>0</td>
<td>0</td>
<td>0.18</td>
<td>0</td>
</tr>
<tr>
<td>disgust</td>
<td>0.12</td>
<td>0.80</td>
<td>0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>fear</td>
<td>0</td>
<td>0</td>
<td>0.62</td>
<td>0.04</td>
<td>0</td>
<td>0.33</td>
</tr>
<tr>
<td>happy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>sad</td>
<td>0.11</td>
<td>0.07</td>
<td>0.14</td>
<td>0.66</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>surprise</td>
<td>0</td>
<td>0</td>
<td>0.13</td>
<td>0</td>
<td>0</td>
<td>0.86</td>
</tr>
</tbody>
</table>

TABLE 5. Confusion Matrix for the LAB708 dataset

<table>
<thead>
<tr>
<th></th>
<th>angry</th>
<th>disgust</th>
<th>fear</th>
<th>happy</th>
<th>sad</th>
<th>surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angry</td>
<td>0.6</td>
<td>0.1</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>Disgust</td>
<td>0.1</td>
<td>0.8</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fear</td>
<td>0</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>Happy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sad</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Surprise</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

F. Temporal Likelihood Variation
The video sequence of each facial expression lasts around 6~7 second with 15 frame/sec. Each face expression is dynamic which evolves over time from the onset, the apex, and the offset. Fig. 10 shows the weighted likelihood of the six classifiers for the same testing facial expression video sequence. The weighted likelihoods $P(Q_c(t)|p_i(y), T)$ for six different classifiers at different time instances are illustrated in Fig. 10. In the early stage, the weighted likelihoods of all facial expression classifiers are very small. It is identified as neutral because the maximum likelihood does not satisfy Eq. (15). The likelihoods of the sad expression classifier and fear expression classifier become larger. However, the likelihood of fear expression classifier is a little bit larger. In the later stage, the weighted likelihood of fear expression classifier increase steeply. It shows more confidence of identifying the entire video sequence as fear expression.

![The likelihood of each classifier at different time instances](image)

The likelihood, $P(Q_c(t)|p_i(y), T)$ of six different classifiers for the input video sequence.

G. Comparisons

Here, we compare the performance of our method with the other three methods [37~38] by using Cohn-Kanade+ data set. The features and classifiers used for the other three methods are shown in Table 6.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Features</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aleksic[37]</td>
<td>Facial animation parameters</td>
<td>HMM</td>
</tr>
<tr>
<td>Kotsia[38]</td>
<td>Gabor wavelet</td>
<td>SVM</td>
</tr>
<tr>
<td>Yeasin[39]</td>
<td>PCA optical flow</td>
<td>HMM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recog. rate</th>
<th>Ours</th>
<th>Aleksic [37]</th>
<th>Yeasin [39]</th>
<th>Kotsia[38]</th>
</tr>
</thead>
<tbody>
<tr>
<td>angry</td>
<td>0.75</td>
<td>0.70</td>
<td>1</td>
<td>0.86</td>
</tr>
<tr>
<td>disgust</td>
<td>0.97</td>
<td>0.97</td>
<td>0.62</td>
<td>0.87</td>
</tr>
<tr>
<td>fear</td>
<td>0.88</td>
<td>0.88</td>
<td>0.76</td>
<td><strong>0.92</strong></td>
</tr>
<tr>
<td>happy</td>
<td>0.97</td>
<td><strong>0.98</strong></td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td>sad</td>
<td>0.80</td>
<td>0.96</td>
<td><strong>0.96</strong></td>
<td>0.89</td>
</tr>
<tr>
<td>surprise</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.96</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.89</td>
<td><strong>0.93</strong></td>
<td>0.90</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Finally, we compare our results with the experiments of three other methods [10, 44, 45] which are...
all image-based framework and tested on Cohn-Kanade database. The average precision of our method is 89% which is higher than the other three methods which use the geometric features [45], the sparse coding [44], and LBP [10], are 87.5%, 88% and 79.1% respectively.

The experimental results show that the recognition rate is not good for the three expressions: angry, fear and sad. However, the recognition rate is much better than the previous three methods for the other three expressions. It is due to incomplete 2D feature extraction. For instance, the two images in Fig. 11 cannot fully reflect the difference between “angry” and “sad”.

![Fig. 11. The angry and sad expressions.](image)

We compare the system performance by using three different datasets and find that the recognition rate for the testing data set from MMI is the worst. For the other two datasets, we have better recognition rate. It is because we cannot have a very precise face calibration for MMI facial images. MMI facial images have a larger 3-D head motion and it complicates our face calibration process. However, under the condition of no-limitation for the test subject, our system still can have acceptable facial expression recognition rate because of our multi-voting Hough forest structure.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training</th>
<th>Restricted head motion</th>
<th>Avg. Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cohn-Kanade+</td>
<td>Yes</td>
<td>Yes</td>
<td>0.89</td>
</tr>
<tr>
<td>MMI</td>
<td>No</td>
<td>No</td>
<td>0.73</td>
</tr>
<tr>
<td>Lab708</td>
<td>No</td>
<td>No</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Our system shows a pretty good performance for the dataset with limited head motion. For dataset of larger head motion, the reliability of the extracted spatial-temporal feature reduces and the recognition rate decreases. Our system requires that the input video facial expression evolves over time from the onset, the apex, and the offset. However, the image-based methods have no such limitation. It takes only one shot as the observation which contains sufficient image characteristics at the apex of facial expressions.

### 6. CONCLUSIONS

We have introduced a facial expression recognition system by using 3-D spatial-temporal local feature extraction and Hough forest. We have also applied the ROI filtering to reduce the error during the training process that increases the discriminative capacity of the parameter voting. Normally, people with the same emotion may show different facial expressions. For some facial expression videos, people usually do not make unanimous conclusion of which expression class. Human facial expression identification is still a difficult problem.

### REFERENCE