International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.6 (2015), pp.91-106
http://dx.doi.org/10.14257/ijsip.2015.8.6.11

A Dynamic Gesture Trajectory Recognition Based on Key Frame Extraction and HMM

Zhang Qiu-yu, Lv Lu, Zhang Mo-yi, Duan Hong-xiang and Lu Jun-chi

School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, P. R. China
zhangqylz@163.com, lvlu901219@126.com

Abstract

Aiming at changing high computational complexity, underdeveloped real time, low recognition rate of dynamic gesture recognition algorithms, this paper present a real-time dynamic gesture trajectory recognition method based on key frame extraction and HMM. Key frames are selected without keeping track of all the details of one dynamic gesture, which is based on difference degree between frames. The trajectory data stream, sorted by the time-warping algorithm, is used to construct the Hidden Markov Method model of dynamic gesture. Finally, optimal transition probabilities are employed to implement dynamic gesture recognition. The result of this experiment implies that this method has high robustness and real time. The average recognition rate of dynamic gesture (0~9) is up to 87.67%, and average time efficiency is 0.46s.

Keywords: *Dynamic gesture recognition, Hand gesture trajectory, Key frame, HMM, Inter-frame difference degree*

1. Introduction

In recent years gesture recognition is a crucial means of Human-Computer Interaction [1, 2]. As a simple and robust motion feature, gesture trajectory is generally used in behavioral action recognition. Compared with static gesture recognition, dynamic gesture recognition is widely applied in real-world. However, due to the fact that dynamic gesture is a gesture sequence consisting of a series of different hand postures whose recognition takes time, the speed of dynamic gesture recognition is not satisfactory [3-4].

Currently, vision-based gesture recognition research falls to the promising research field. Bao, *et al.*,[5] proposed tracking gesture trajectory method by SURF feature, which established model by dynamic time warping algorithm. The data stream clustering method based on correlation analysis was developed to recognize a dynamic hand gesture. The recognition rate of training and testing set were 87.1% and 84.6% respectively. Ren, *et al.*, [6] proposed a model of spatio-temporal appearance based on hierarchical fusion of multi-modal information and an algorithm of dynamic space-time warping for dynamic gesture recognition under complex background information. The average recognition rate was quite well, however, the average time efficiency of segmentation, parameter extraction and identification were 1.1s, 0.9s, 0.07s respectively. Zhang, *et al.*, [7] classified length, position, velocity, and acceleration of the hand as the dynamic characteristics of the hand to produce HMM sign language models, which were used during video retrieval. David, *et al.*, [8] used tensor voting to filter the trajectory, which was obtained from hand detector, and found the orientation in Radon space to create gesture model. Thus it allowed a large number of meaningful gestures to be defined, whose rate was close to the method of Ref. [6]. In addition, common gesture feature extraction methods are also based on color information [9], gradient histogram [10] *etc.*. But these above gesture recognition methods are poor real-time, and susceptible to quite a few factors, including complex background, lighting, complexity of algorithm.

ISSN: 2005-4254 IJSIP

With the rapid development of hardware devices in recent years, most researchers have used Kinect [11], which was launched by Microsoft Company as the latest equipment in 2010, to identify gesture. Because it is not affected by illumination change and complex background [12], Kinect captures a multitude of attention of researchers both at home and abroad [13-15]. At the same time, dynamic hand gestures could generate a large amount of redundant information; in consequence, some researchers began to apply key frame extraction method to dynamic gesture recognition system. Wang [16] used Affinity Propagation Clustering to extract key-frame of the video sequence adaptively, which was used to ensure a higher recognition rate and also meet real-time requirements. Ramakrishnan, *et al.*, [17] put forward motion capture data method to split gestures sequence automatically, which could accurately describe the change of hand.

In conclusion, this paper proposes a real-time dynamic gesture trajectory recognition method based on key frame extraction and HMM to improve real-time capability and efficiency of the recognition. The algorithm does not keep track of all the details of dynamic hand gestures and only selects some key frames in the process of gesture movement .It employs key frame extraction method based on difference degree between frames. These images are used to segment gesture and calculate hand centers. The dynamic signal models are established through HMM after trajectory data streams are received by time warping. Finally, optimal transition probability matrices of models are utilized to recognize dynamic hand gestures.

2. Related Work

Currently, key frame extraction methods can be summarized as the following four kinds:

1) Key frame extraction method based on shot [18]. This method puts the first frame, the middle or the last one of each shot as the key frame of this shot. It is simple in design with low computational complexity and suitable for simple content or fixed scene switching. However, for complex scenarios or variety of camera lens transformations the key frames extracted above cannot often represent information of lens accurately.

2) Key frame extraction method based on motion analysis [19]. Generally, method based on optical flow computation selects a number of key frames according to the structure of the lens, but this method has a huge amount of calculation. It also has poor real time while the local minimum calculated is not necessarily accurate.

3) Key frame extraction method based on visual content [20]. Changes of each frame's color, texture and other visual information are developed to extract key frames, which will take the current frame as the key frame when the information has significant changes. Firstly, the first frame of camera is one key frame. Secondly, difference degree between the previous key frame and this one is calculated. Finally, the frame is chosen as a key frame if the difference is greater than a certain threshold. It can select key frames according to the different degree of information, which is not necessarily representative. What is more, the number of frames is easily excessive.

4) Key frame extraction method based on cluster analysis [21]. Clustering method is of high computation efficiency and effectively access to significant change of video shot visual content. Yet it can't effectively preserve time sequence and dynamic information of original camera image frames. This method is the most popular method of current key frame extraction methods. An initial class center is determined at first. Then the current frame is judged to classify as a new class according to the distance between current frame and the class center. At last the nearest distance frame of each class is selected to represent the class after classification, which is one of key frame sequence in camera lens.

Whereby, the key frame extraction process can be summarized in two steps:

1) Seeking quantization parameters of images feature;
2) Judging whether the quantitative characteristics of parameters is key feature values.

3. Dynamic Gesture Trajectory Recognition Based on Key Frame Extraction and HMM

3.1. Dynamic Gesture Trajectory Recognition Process

As shown in Figure 1, this paper studies two aspects of dynamic gesture trajectory recognition process:

1) Key frame extraction based on difference degree of inter-frame. Initial key frame sequence is obtained by the algorithm of selecting maximum in inter-frame difference degree, which is calculated between frames of video. Subsequently, final key frame sequence is gotten by sorting key frames double using time warp algorithm;

2) Trajectory feature extraction. Gesture segmentation algorithm based on skin color is used to get area of hand for sequence obtained above. Then distance sequence to center of trajectory sequence is used as characteristics to input HMM algorithm.

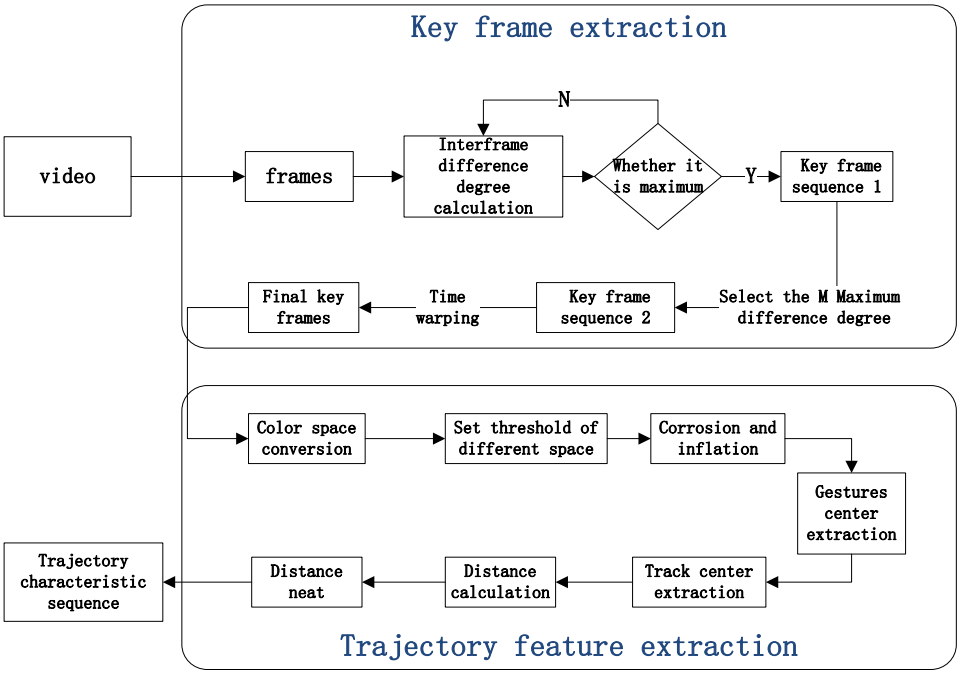


Figure 1. Flow Chart of Dynamic Gesture Trajectory Recognition

3.2. Key Frame Extraction Method Based on Inter-frame Different Degrees

Video is composed of a series of frames, expressed as $S = \{f_1, f_2, \ldots, f_N\}$. The key frame sequence is a paramount part in video, which contains a set of discrete image frames. The method of key frame extraction is: one or more key frames are extracted according to the content of shot; thereby the essence of video information is represented by a small number of frames. Namely M ($M<<N$) frames are chosen from N frames as key frames for subsequent operations. This paper selects key frames using key frame extraction method based on inter-frame different degree. Inter-frame difference degree is calculated with Euclidean distance, of which the largest M maximums are key frames. Finally the final key frame sequence is gotten by two time-warping for frames obtained above.

This paper chooses Euclidean distance as the difference degree between frames to select key frames. The greater the value of *diff* is, the larger the difference degree between the frame and its adjacent frames is. Therefore, frame is chosen, whose value of *diff* is large.

Step 1: Calculate Euclidean distance between frames as difference degree between adjacent frames. The formula is

$$diff_n = \sqrt{\sum_i \sum_j (G_{n+1}(i, j) - G_n(i, j))} \quad (1)$$

where n=2, 3, …, N-1. After improving (1), get

$$diff_n = \sqrt{\sum_i \sum_j ((G_{n+1}(i, j) - G_n(i, j)) - (G_n(i, j) - G_{n-1}(i, j)))} \quad (2)$$

$$G_n(i, j) = R_1 \times r_n(i, j) + G_1 \times g_n(i, j) + B_1 \times b_n(i, j) \quad (3)$$

where, R_1=0.299, G_1=0.587, B_1=0.114,and $r_n(i, j)$, $g_n(i, j)$ and $b_n(i, j)$ respect the red, green, and blue components of nth frame image in (i, j).

Step 2: Select the local maximum values of $diff_n(n = 2, 3, \ldots, N\text{-}1)$ as the key frame sequence, showing as the points identified with blue "○" in Figure 2. Judgment formula is

$$e(i) = \begin{cases} 1 & diff''(i) < 0 \cap diff'(i) = 0 \\ 0 & ohers \end{cases} \quad (4)$$

But the value of $diff_n$ is not necessarily large when it is local maximum values (Figure 2). Consequently other means are also needed to filter some local maximum points.

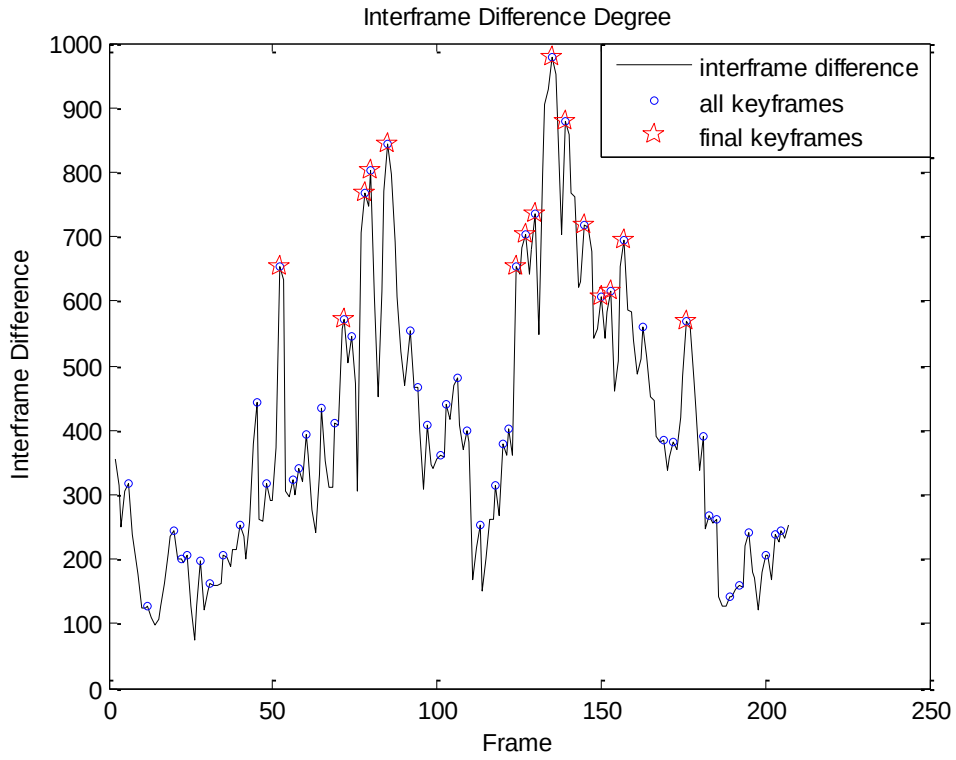


Figure 2. Key Frame Extraction according to Different Degree Value

Step 3: Sort all local maximum points to select the M biggest points as key frames, Show as the points identified by red "☆" in Figure 2. Set M=15.

Step 4: Ensure the order of the M frames by time-warping algorithm so that final key frames of video image sequence is strictly chronological.

After getting a series of key frames by the use of key frame extraction algorithm above, skin color is required to segment gesture from image to extract the gesture area so that these images can be used for HMM training and recognition.

3.3. Gestures Area Segmentation

RGB color space should be transformed to color space in which brightness and chromaticity is separated, because of subtle difference of chromaticity, on the contrary, significant changes of brightness in skin color. This paper takes advantage of YCbCr [22] and HSV [19] color space to extract gesture area.

Step 1: Transform images from RGB color space to one color space in which brightness and chromaticity is separated.

Step 2: Set the threshold. Skin color clustering of YCbCr color space likes a two head spindle, that is to say, when the value of Y is larger or smaller, the skin color clustering region will be contracted. Therefore, transformation of color format is necessary in segmentation. The conversion formula of YCbCr space to YCb'Cr' space is

$$C_1' = \begin{cases} (C_1(Y) - C_1'(Y))\dfrac{Wc_1}{Wc_1(Y)} + C_1'(Y) & Y \notin [K_l, K_h] \\ C_1(Y) & Y \in [K_l, K_h] \end{cases} \tag{5}$$

$$Wc_1(Y) = \begin{cases} WLc_1 + \dfrac{(Y - Y_{\min})(Wc_1 - WLc_1)}{K_l - Y_{\min}} & Y < K_l \\ WHc_1 + \dfrac{(Y_{\max} - Y)(Wc_1 - WHc_1)}{Y_{\max} - K_h} & Y > K_h \end{cases} \tag{6}$$

where sets K_l=125, K_h=188 (segmentation domain of nonlinear color transformation), Y_{min}=16, Y_{max}=235 (the minimum and maximum values of Y component in skin color clustering), Wc_b=46.79, WLc_b=23, WHc_b=14, Wc_r=38.76, WLc_r=20, WHc_r=10. For YCb'Cr' color space, threshold is

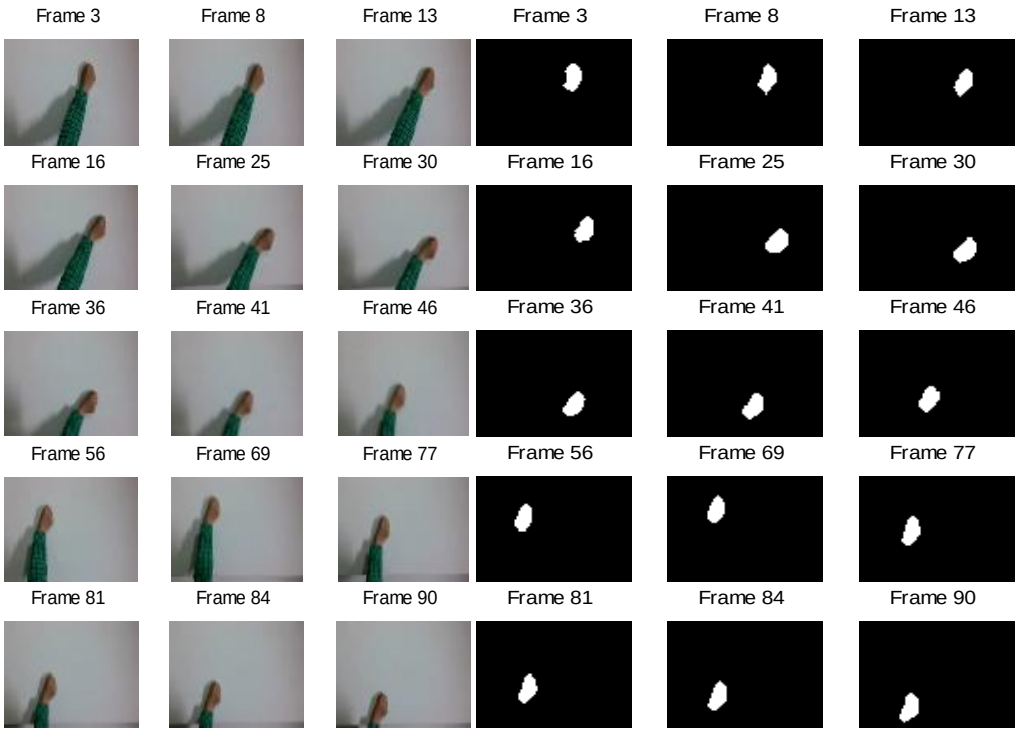
$$b(i, j) = \begin{cases} 1 & \dfrac{(x - x_p)^2}{x_q^2} + \dfrac{(y - y_p)^2}{y_q^2} \le 1 \\ 0 & others \end{cases} \tag{7}$$

$$[x \quad y] = [Cb'(i, j) - C_1 \quad Cr'(i, j) - C_2] \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \tag{8}$$

where x_p=1.60, y_p=2.41, x_q=25.39, y_q=14.03, C_1=109.38, C_2=152.02, γ=2.35(rad), $b(i, j)$ is the value after image segmentation in (i, j), Cb', Cr' represent the chromaticity of blue and red space in YCb'Cr' respectively.

Skin segmentation based on HSV sets H threshold for 0.03~0.128.

Figure 3 is skin color segmentation outcome of one dynamic hand gesture “2”. Figure 3(a) manifests M original color images after key frame extraction of one sample, Figure 3(b) shows binary image of gesture area, which is segmented based on skin color for images in Figure 3(a). It can be seen that the method is not affected by shadow and accurately can segment the gesture area.



(a) Original images (b) Binary images

Figure 3. Gesture Segmentation Effect based on the Color of Skin

Step 3: Erosion and dilation for images segmented. In view of the influence of light conditions, there will be a few noise points in the image after being segmented. Therefore, image should be eroded and expanded to eliminate noise points. Computational formula is expressed as vectors:

$$X \Theta B = \left\{ p \in \varepsilon^2, p + b \in X, \forall b \in B \right\} \tag{9}$$

$$X \oplus B = \left\{ p \in \varepsilon^2, p = x + b, x \in X \cap b \in B \right\} \tag{10}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

That is B={(0,3), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (2,4), (2,5), (3,0), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,4), (4,5), (5,2), (5,3) ,(5,4), (6,3)}.

The intention of erosion and expansion is to eliminate some small noise points, smooth the edge of the gesture area, at the same time, and not alter their area significantly. Figure 4 demonstrates the effect of erosion and expansion. Figure 4(a) manifests an original image frame. Figure 4(b) reveals binary image of the original image through segmentation based on skin color. Figure 4(c) is the image after Figure 4(b) is eroded and expanded. It can be seen that noise in Figure 4(b) is eliminated and gesture area border has become relatively smooth. Meanwhile the area of hand is not changed.

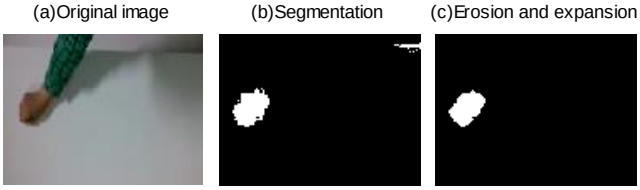


Figure 4. Noise, Erosion and Expansion Processing of Images

3.4. Trajectory Feature Extraction

Step 1: Calculate gesture center in each image, namely the centroid (x_n, y_n). The computational formula is

$$(x_n, y_n) = \left(\frac{\sum_i \sum_j x \cdot b_n(i,j)}{\sum_i \sum_j b_n(i,j)}, \frac{\sum_i \sum_j y \cdot b_n(i,j)}{\sum_i \sum_j b_n(i,j)} \right) \tag{12}$$

Figure 5 is the distribution of trajectory points of gesture (0~9), which is calculated by formula (12) for key frames above. The gesture center of one key frame sample in some dynamic gestures is indicative of red "+" in Figure 5, namely trajectory. The blue line represents contours of trajectory.

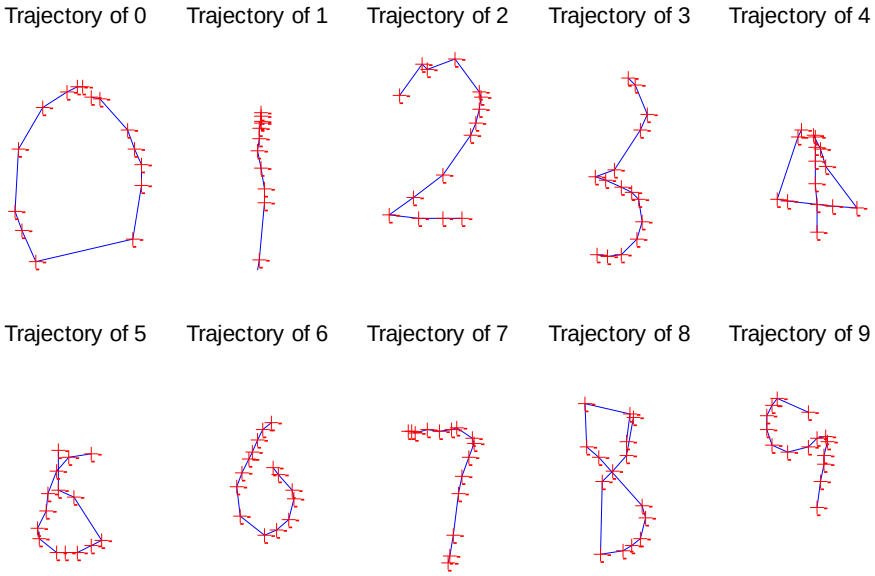


Figure 5. Trajectory Profiles of Dynamic Gesture (0~9)

Step 2: Compute the trajectory center (x_0, y_0) of trajectory point {(x_1, y_1), (x_2, y_2), …, (x_M, y_M)}. The formula is

$$(x_0, y_0) = \left(\frac{1}{T} \sum_{t=1}^{M} x_t, \frac{1}{T} \sum_{t=1}^{M} y_t \right) \tag{13}$$

Step 3: Calculating distances between each centroid point (x_n, y_n) and the trajectory center (x_0, y_0) is described as follows.

$$r_n = \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2} \tag{14}$$

Step 4: Use (15) for distance neat for $G = (r_1, r_2, \ldots, r_M)$. Distance neat figure is shown in Figure 6, neat length $L = 5$. The formula is

$$l_n = \frac{r_n}{L} + 1 \tag{15}$$

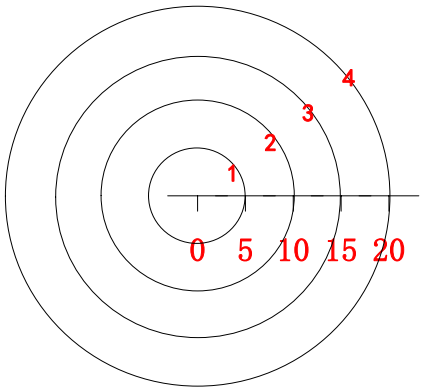


Figure 6. Distance Neat Figure

Finally an observation sequence vector $D = (l_1, l_2, \ldots, l_M)$ is gotten, which is input HMM as characteristic of training models.

3.5. Gesture Recognition

The composition of the HMM:

1) The first-order Markov process, namely description of states transferring, which is described by state transition probability matrix A and initial state distribution π.

2) Random observation process, namely description of corresponding relationship between state and observed sequence, which is described by state output probability distribution matrix B.

Evaluation, decoding and learning are three main problems of the HMM model, which can be solved by forward-backward algorithm, Viterbi algorithm [23] and Baum Welch algorithm [24] respectively. This paper adopts pattern from left to right as the basis of model, and the more model states are, the better effect in theory.

3.5.1. Evaluation: The evaluation is identification. All output probabilities $P(O / \lambda)$ of observation sequences are produced by HMM under given model $\lambda = \{\pi, A, B\}$ and observation sequence O. Compared with the output probability of model, the highest probability is recognition result. The observation sequence here is characteristic vector extracted above. $P(O / \lambda)$ is computed by forward algorithm, whose process is as follows:

Step 1: Initializing $\alpha_1(i) = \pi_i b_i(o_1)$, i=1, 2, …, N;

Step 2: Recurring computation formula is

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), (1 \le t = 1,2,\ldots,T-1), (j = 1,2,\ldots,N) \tag{16}$$

Step 3: At last, $P(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i)$.

where, $\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = s_i \mid \lambda)$ represents probability of state s_i at the moment t when observation sequence generated is $o_1, o_2, \ldots, o_t$ for given model $\lambda=\{\pi, A, B\}$.

The backward algorithm process is as follows:

Step 1: Initializing $\beta_t(i) = 1$, i=1, 2, …, N;

Step 2: Recurring computation formula is

$$\beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(o_{t+1})\beta_{t+1}(j),(t=T-1,T-2,\ldots,1),(i=1,2,\ldots,N) \tag{17}$$

where $\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T, q_t = s_i \mid \lambda)$ represents the probability of state s_i at moment t when the observation sequence is $o_1, o_2, \ldots, o_t$ at the moment $t+1$ for given model $\lambda=\{\pi, A, B\}$.

3.5.2. Decoding: Decoding is looking for the most suitable implicit state sequence, which will produce observation sequence O under condition of given model parameter $\lambda=\{\pi, A, B\}$ and observation sequence O. That is to say, search its best state sequence $Q^*=q_1^* q_2^* \ldots q_T^*$ to explain the observation sequence O. The Viterbi algorithm is frequently-used, which is based on dynamic programming. The specific steps of Viterbi algorithm is as follows:

Step 1: Initialize $\delta_1(i)=\pi_i b_i(o_1)$, $(1\le i\le N)$, $\psi_1(i) = 0$, $(1\le i\le N)$;

Step 2: Recurrence formula is

$$\delta_t(j)=\max_i\left[\delta_{t-1}(i)\cdot a_{ij}\right]b_j(o_t),(2\le t\le T,1\le j\le N) \tag{18}$$

$$\psi_t(j)=\arg\max_i\left[\delta_{t-1}(i)a_{ij}\right],(2\le t\le T,1\le j\le N) \tag{19}$$

Step 3: Terminate,

$$P^*=\max_i\left[\delta_T(i)\right],\quad q_T^*=\arg\max_i\left[\delta_T(i)\right] \tag{20}$$

The best state sequence back is: $q_t^*=\psi_{t+1}(q_{t+1}^*)$, $(t=T-1, T-2,\ldots, 1)$, where $\delta_t(i)$ represents the maximum probability of state s_i at the moment t under given model parameter $\lambda=\{\pi, A, B\}$ and observation sequence $o_1, o_2, \ldots, o_t$.

3.5.3. Learning: Learning is training HMM models, which is procession of constant reassessment for the model parameters under a given observation sequence of sample. After HMM model parameter $\lambda=\{\pi, A, B\}$ is adjusted constantly through iterative computation, the probability $P(O / \lambda)$ of observation sequence O reach maximum value, whose model is trained to be the most suitable for sample set. The specific steps to calculate optimal parameters λ^* of HMM are as follows:

Step 1: Establish an initialization model of HMM $\lambda=\{\pi, A, B\}$.

Step 2: Estimate a new model parameter $\overline{\lambda}=\left(\overline{\pi}, \overline{A}, \overline{B}\right)$ according to given model parameter $\lambda=\{\pi, A, B\}$ and observation sequence O. Revaluation formula is

$$\overline{\pi_i}=P(q_1=s_i)=\gamma_1(i),\quad \overline{a_{ij}}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)},\quad \overline{b_j(k)}=\frac{\sum_{t=1,o_t=v_k}^{T-1}\gamma_t(j)}{\sum_{t=1}^{T-1}\gamma_t(j)} \tag{21}$$

where $\gamma_t(i) = P(q_t = s_i \mid O, \lambda)$ represents probability of state s_i at moment t under given observation sequence O and HMM model parameter λ. By forward-backward variables the formula is expressed as follows.

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(O/\lambda)} = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)}, (1 \le i \le N) \tag{22}$$

where $\xi_t(i, j) = P(q_t = s_i, q_{t+1} = s_i \mid O, \lambda)$ represents the probability of state s_i at moment t and s_i at moment t+1 under given observation sequence O and HMM model parameter λ. By forward-backward variables the formula is expressed as follows.

$$\xi_t(i, j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \beta_{t+1}(j)} \tag{23}$$

Step 3: Calculate the probability $P(O/\lambda)$ under model λ and probability $P(O/\bar{\lambda})$ under model λ by forward-backward algorithm for observation sequence O.

Step 4: ε is the threshold of convergence. When formula (24) is met, $P(O/\bar{\lambda})$ is convergent. At this time, λ is the most suitable parameter got by training the HMM which can represent gesture. Whereas, continue to implement **step 2** until formula (24) is met. Formula (24) is

$$\left| \log P(O|\bar{\lambda}) - \log P(O|\lambda) \right| < \varepsilon \tag{24}$$

4. Experiment Results and Analysis

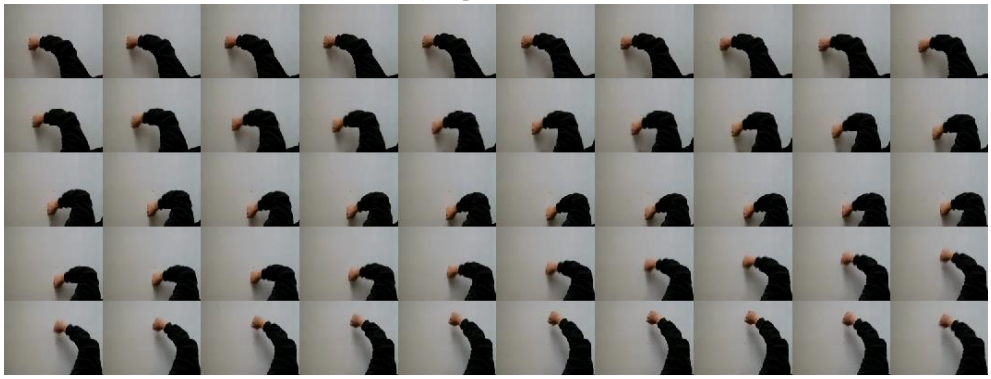
Experiment platform: the experiment is performed using a Windows 7 OS in MATLAB R2013a using Intel core i3-2120, 3.3GHz and 4G.

The experiment data set of dynamic hand gestures consists of ten dynamic hand gestures "0~9". Three experimenters are invited to perform 10 times every gesture, in consequence, 300 gesture samples (3×10×10) are gotten. The continuous time of every gesture is about 3*s*~12*s* (frame rate δ=25*fps*). For simulating close gesture recognition applications, the distance between gesture and camera is about 30~ 80cm and every image is 160×120 with 24-bit true color.

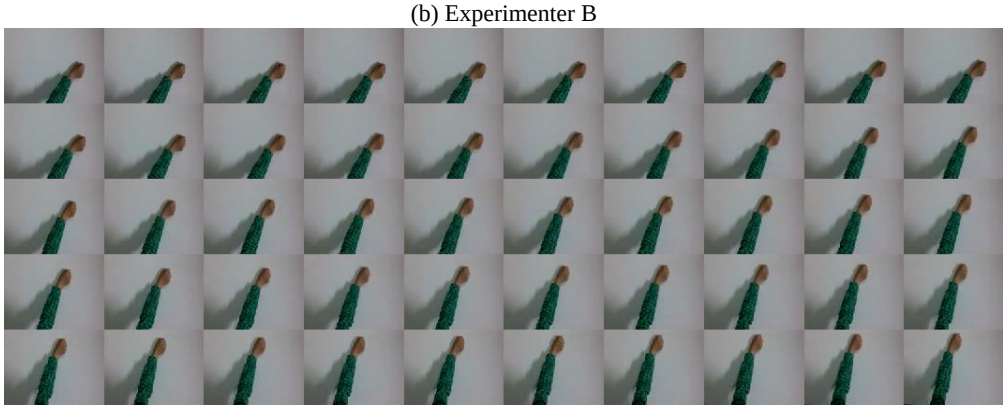
Figure 7 shows 1~50 original image frames of dynamical gesture sample "7" in gesture video data set.

Figure 7(a), Figure 7(b) and Figure 7(c) demonstrate 1 to 50 frames of original videos regarding experimenter A, B and C respectively.

(a) Experimenter A



(b) Experimenter B



(c) Experimenter C

Figure 7. The 1~50 Frame Original Image of Dynamical Gesture "7"

4.1. Recognition Effect Comparison of YCbCr and HSV

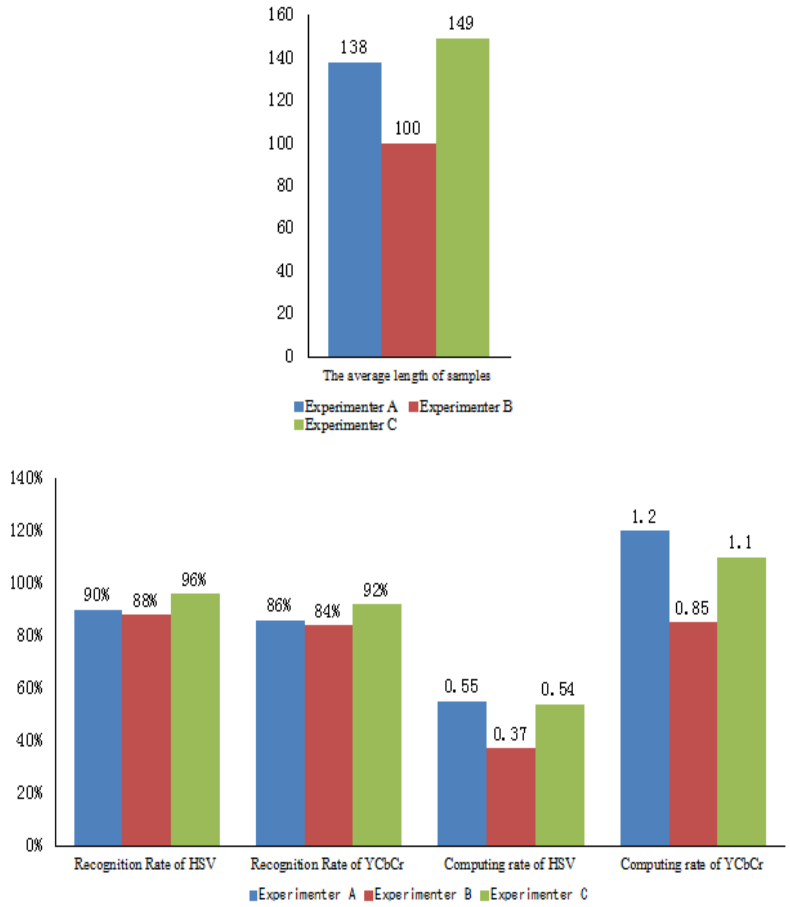
In the experiment, train set of gesture is composed of six samples of gesture in every experimenter respectively, namely that every train set has 60 (6×10) samples. For one experimenter all the samples constitute test set, in which every test set is made of 100 samples. Every sample is trained and recognized in dynamical gesture sets based on YCbCr and HSV respectively. The results are expressed in Figure 8. Figure 8(a) manifests average sequence lengths of three experimenters. Figure 8(b) shows respective recognition rate and average computing rate for experimenters based on YCbCr and HSV during extraction of gesture area.

For Figure 8, "velocity" represents average time of every sample for one experimenter. The computational equation of recognition rate is

$$\eta = \frac{TR}{TR+FR} \tag{25}$$

where *TR* represents the number of correct recognition samples, *FR* represents the number of error recognition samples, η is recognition rate.

In Figure 8, the average sequence length of experimenter B can be seen is the lowest, meanwhile, the recognition and the test rate are lowest. The average sequence length of A and C are the same in general. However, the recognition rate of A is much lower than that of C under the condition of close average sequence length. From Figure 8, recognition rate and velocity of different experimenters is different because accuracy and speed of sample are different when performing the same sample for different experimenter.



(a) Average sequence length (b) Recognition rate and average computing rate

Figure 8. Recognition Results for Different Experimenters based on HSV and YCbCr

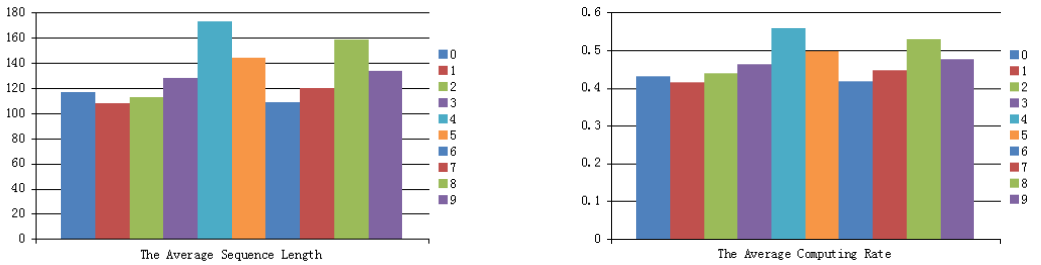
1) The velocity of HSV and YCbCr is proportional to the average sequence length. In condition of certain frame rate σ, the longer time of sample is, the more inter-frame difference degree needs to compute. Accordingly the processing time is longer under certain number of key frames. However, too little time of sample means that fast speed of hand movement would result in loss of detailed information in dynamical gesture

2) With same recognition rate, adopting time of HSV is much less than that of YCbCr for segmenting gesture area. Therefore, gesture segmentation based on HSV can better meet requirements of real-time experiment. YCb'Cr' for segmentation of gesture area must be switched because skin color clustering YCbCr color space is nonlinear.

However, the recognition rate is not only influenced by sequence length, which still needs to be studied in the future.

4.2. Mixed Sample Recognition Rate

The experiment selects all samples of A, B and C to conduct experiment. Each of ten dynamic gestures has 30 samples. Six samples of every gesture of each experimenter are chosen to train, that is to say number of train samples is 180 (3×6×10). And 300 samples are recognized. The experiment results are presented in Figure 9 and Table 1.



(a) Average sequence length (b) Average computing rate

Figure 9. Comparison of Average Sequence Length and Computing Rate between 0~9

Figure 9 illustrates average length of sequence and average computing speed concerning gesture. Figure 9(a) demonstrates the comparison of the average sequence length between gesture 0~9. Figure 9(b) demonstrates the comparison of the average computing rate of gesture 0~9. The longer of the average sequence length, the bigger of the computing rate. To sum up, the computing rate is proportional to the sequence length. Figure 9 manifests the same result with Figure 8.

Table 1 demonstrates total number of recognition error sample is 37 and the recognition rate is 87.67%. The recognition rate is lower compared with Figure 8. The total running time of experiment is 135.257129s and the average running time is 0.457524*s*.

Table 1. Recognition Rate and Real Time of Gesture "0~9"

Dynamic Gesture	Correct identification number	Error identification number	Recognition rate
0	27	3	90%
1	30	0	100%
2	25	5	83.33%
3	25	5	83.33%
4	26	4	86.67%
5	23	7	76.67%
6	27	3	90%
7	30	0	100%
8	24	6	80%
9	26	4	86.67%

From Table 1, three reasons of algorithm leading to errors can be gotten:

1) The trajectories of different experimenter performing a same gesture are different. Therefore, to balance differences among them the model generated by training would result in deviation.

2) The similar trajectory is easy to confuse because of extracting *M* frame images in videos only, such as "0" and "2", "8" and "9".

3) The number of HMM training characteristics is too small, which can't describe the position of tracing point accurately. Meanwhile, recognition is easy to confuse with other dynamical gestures.

Otherwise, inaccurate gesture region extraction is considerate for the reasons that lead to the error. Owing to setting threshold value, which is the simplest method in gesture segmentation, inaccurate gesture region leads to biased centroid point and distance.

4.3. Compared with the Existing Methods

For test recognition rate and speed of the algorithm, the method is compared with algorithm proposed in the Ref. [5] and Ref. [6].

Table 2. Comparative Analysis of Algorithm Results

	This paper	Ref. [5]	Ref. [6]
Frequency of Platform	3.3GHz	2.2GHz	600MHz
Frame Rate(*frames/s*)	25	8~16	10
Recognition rate (%)	87.67	84.6	91.7
operation speed (s)	0.457524	1~3	≥2.07

Under condition that the frame rate is more twice than that in the Ref. [5] and Ref. [6], Table 2 shows that the operation speed of algorithm is twice higher than that of Ref. [5] and Ref. [6], although frequency of the algorithm experimental platform is relatively high. The result suggests that the algorithm does not only ensure the requirements of identification, but also meets real-time capability of dynamic gesture trajectory recognition. The experimental data demonstrates that the method is more reliable which can be widely applied to the field of human-computer interaction.

5. Conclusion

This paper present a real-time dynamic gesture trajectory recognition method based on the combination of key frame extraction and HMM. Considering that key frame extraction can reduce redundancy information of video, the method puts forward key frame extraction algorithm which can extract 15 frames from more than 100 frames of image sequence as key frames of this gesture based on difference degree between frames. In addition, the use of gesture segmentation method based on skin color is to obtain gesture area. Finally the features are extracted from its trajectory in order to establish a HMM model as input information for training and recognition. The experimental results indicate that the method can fully satisfy real-time requirements of dynamic gesture trajectory recognition. The method is effective, reliable and robust in complex illumination.

The further work in this paper will study simple and precise gesture extraction under complex background, at the same time, algorithm based on the HMM will be improved to accurately establish gesture models.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61363078), the Natural Science Foundation of Gansu Province of China (No. 1212RJZA006, No. 1310RJYA004). The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] X. Y. Wang, G. Z. Dai, X. W. Zhang and F. J. Zhang, "Recognition of Complex Dynamic Gesture Based on HMM-FNN Model", Journal of Software(*in Chinese*), vol. 19, no. 9, **(2008)**, pp. 2302−2312.
[2] S. S. Rautaray and A. Agrawal, "Real time multiple hand gesture recognition system for human computer interaction", IJISA, vol. 4, no. 5, **(2012)**, pp.56-64.
[3] H. Li and M. Greenspan, "Model-based segmentation and recognition of dynamic gestures in continuous video streams", Pattern Recognition, vol. 44, no. 8, **(2011)**, pp.1614-1628.

[4] Y. Song, D. Demirdjian and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction", ACM Transactions on Interactive Intelligent Systems(TiiS), vol. 2, no. 1, **(2012)**, Article 5, pp. 1-28.
[5] J. T. Bao, A. G. Song, Y. Guo and H. R. Tang, "Dynamic Hand Gesture Recognition Based on SURF Tracking", Robot(*in Chinese*), vol. 33, no. 4, **(2011)**, pp. 482-489.
[6] H. B. Ren, Y. X. Zhu, G. Y. Xu, X. Y. Lin and X. P. Zhang, " Spatio-Temporal Appearance Modeling and Recognition of Continuous Dynamic Hand Gestures", Chinese Journal of Computers, vol. 23, no. 8, **(2000)**, pp. 824-828.
[7] S. Zhang and B. Zhang, "Using HMM to sign language video retrieval", Proceedings of IEEE 2010 Second International Conference on Computational Intelligence and Natural Computing Proceedings (CINC), **(2010)** Sept 13-14, Wuhan, China.
[8] C. David, V. Gui, P. Nisula and V. Korhonen, "Dynamic hand gesture recognition for human-computer interactions", Proceedings of IEEE 2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), **(2011)** May 19-21, Timiúoara, Romania.
[9] H. Duan and Y. Luo Y, "A Method of Gesture Segmentation Based on Skin Color and Background Difference Method", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering, **(2013)** March 22-23, Hangzhou, China.
[10] A. Misra, T. Abe and K. Deguchi, "Hand Gesture Recognition Using Histogram of Oriented Gradients and Partial Least Squares Regression", Proceedings of the MVA, **(2011)** June 13-15, Nara, Japan.
[11] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications", Sensors, vol. 12, no. 2, **(2012)**, pp. 1437-1454.
[12] J. Sanchez-Riera, J. Čech and R. Horaud, "Action recognition robust to background clutter by using stereo vision", Proceedings of the Computer Vision–ECCV 2012, Workshops and Demonstrations, Springer Berlin Heidelberg, **(2012)**, pp. 332-341.
[13] Y. Zhang, S. Zhang and Y. Luo, "View-Invariant 3D Hand Trajectory-Based Recognition", Journal of University of Electronic Science and Technology of China, vol. 43, no. 1, **(2014)**, pp. 60-65.
[14] A. Hernández-Vela, M. Á. Bautista, X. Perez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol and C. Angulod, "Probability-based Dynamic Time Warping and Bag-of-Visual-and-Depth-Words for Human Gesture Recognition in RGB-D", Pattern Recognition Letters,vol.50, no.1, **(2013)**, pp. 112-121.
[15] J. Yu, G. Tian and J. Yin, "Dynamic Hand Gesture Recognition Algorithm Based on Depth Information", Journal of Shandong University (Engineering Science) (*in Chinese*), vol. 44, no. 3, **(2014)**, pp. 52-56.
[16] Y. F. Wang, "Research on Core technology of Dynamic Gesture Recognition", Sichuan Normal University, Sichuan, China, **(2011)**.
[17] A. S. Ramakrishnan and M. Neff, "Segmentation of hand gestures using motion capture data", Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, **(2013)** May 6-10, Saint Paul, Minnesota, USA.
[18] Y. Rui, T. S. Huang and S. Mehrotra, "Exploring video structure beyond the shots", Proceedings of the IEEE International Conference on Multimedia Computing and Systems, **(1998)** Jun 28-Jul 1, Austin, TX.
[19] W. Wolf, "Key frame selection by motion analysis", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP-96), vol. 2, **(1996)** May 7-10, Atlanta, GA.
[20] H. J. Zhang, J. Wu and D. Zhong, "An integrated system for content-based video retrieval and browsing", Pattern Recognition, vol. 30, no. 4, **(1997)**, pp. 643-658.
[21] A. Hanjalic and H. J. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis", IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 8, **(1999)**, pp. 1280-1289.
[22] X. Jiang and X. B. Lu, "A Dynamic Gesture Recognition Method Based on Computer Vision", Sciencepaper Online, http://www.paper.edu.cn, **(2013)**, pp. 1-8.
[23] M. Elmezain and A. Al-Hamadi, "Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models", Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology, **(2007)** Dec 15-18, Giza.
[24] Y. Zhang, P. Liu and F. K. Soong, "Minimum error discriminative training for radical-based online Chinese handwriting recognition", Proceedings of the IEEE Ninth International Conference on Document Analysis and Recognition(ICDAR 2007), vol. 1, (**2007**) September 23-26; Parana.

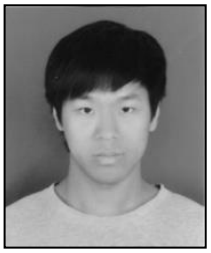
Authors

Zhang Qiu-yu, he is a researcher/PhD supervisor, graduated from Gansu University of Technology in 1986, and then worked at school of computer and communication in Lanzhou University of Technology. He is vice dean of Gansu manufacturing information engineering research center, a CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis analysis, image understanding and recognition, multimedia communication technology.

Lv Lu, she is a master student in Computer Application Technology from Lanzhou University of Technology, Lanzhou, China. She obtained an undergraduate degree in Computer Science and Technology from Lanzhou University of Technology, Lanzhou, China, in 2012. Her research interests include image processing and pattern recognition, dynamic gesture trajectory recognition.

Zhang Mo-yi, she is a PhD student, graduated from Lanzhou University of Technology in 2010, and then worked as the lecturer at the school of computer and communication in Lanzhou University of Technology. Her research interests include Image Processing and Pattern Recognition.

Duan Hong-xiang, she is a PhD student, graduated from Xi'an University of Architecture and Technology with Bachelor of engineering degree in 1999, received master degree from Lanzhou University of Technology in 2011, and then worked at school of computer and communication in Lanzhou University of technology. Her research interests include image processing and pattern recognition, dimensionality reduction for high-dimensional data.



Lu Jun-chi, he graduated from Liaoning Technical University, Liaoning, China, in 2009. He is currently master student in Computer Applications, Lanzhou University of Technology, Lanzhou, China. His research interests include image processing and pattern recognition.