

Hidden Markov Models Based Dynamic Hand Gesture Recognition with Incremental Learning Method

Meng Hu, Furao Shen*, and Jinxi Zhao

Abstract—This paper proposes a real-time dynamic hand gesture recognition system based on Hidden Markov Models with incremental learning method (IL-HMMs) to provide natural human-computer interaction. The system is divided into four parts: hand detecting and tracking, feature extraction and vector quantization, HMMs training and hand gesture recognition, incremental learning. After quantized hand gesture vector being recognized by HMMs, incremental learning method is adopted to modify the parameters of corresponding recognized model to make itself more adaptable to the coming new gestures. Experiment results show that comparing with traditional one, the proposed system can obtain better recognition rates.

I. INTRODUCTION

HUMAN-Computer Interaction (HCI) has always been one of the most attractive research fields in computer science. In the past, we use joystick, keyboard and mouse to communicate with computer. Although these icy devices play a dominated role in HCI, they require us to operate them directly, which may hinder the naturalness of interaction. Therefore, they are not supposed to be the ideal media to interact with computer. To overcome this shortage, researchers have been diligent for finding more natural HCI media. In these media, with the advantages of simpleness, convenience and easy to operate, hand gesture turns to be a perfect alternative to replace these devices[1].

Generally, hand gesture has two categories in definition[2]: static hand gesture and dynamic hand gesture. In static hand gesture, hand shape, outline and contour imply gesture meaning, while in dynamic hand gesture, gesture meaning is conveyed in hand velocity, trajectory, orientation, etc. Based on the equipment hand applies, we classify hand gesture recognition into three categories: device-based, touch-screen based and vision-based. In device-based pattern, gesturers are required to wear cumbersome devices, like glove[3], sensor[4], to interact with computer. Touch-screen based pattern is more commonly used in mobile services, like iphone, ipad, etc. Nowadays, with the development of computer vision technology, gesturers can stand in front of a camera by using their bare hands to interact with computer, which is much more convenient for both ordinary and disabled people.

Recently, there have been an increasing number of hand gesture recognition applications and researches based on vision-based methods. They use different models, such as

Fuzzy System[5], Neural Network (NN)[6], Hidden Markov Models (HMMs)[7], to recognize hand gestures.

HMM is a statistical model which has been widely used in speech recognition, part-of-speech tagging and fault diagnosis, etc. After being successfully applied in speech recognition by LR. Rabiner[7], people discover its advantages in dealing with spatial-temporal time-series data. The first authors who introduce HMMs to gesture recognition are J. Yamato et al.[8]. In this paper, researchers use HMMs to recognize six different tennis-playing gestures and obtain 96% recognition rate in average. Lee et al.[9] propose an ergodic model based on adaptive threshold to classify the meaningful gestures by combining all states from all trained gesture models using HMM. L. Nianjun et al.[10] propose a method to recognize 26 English letters from A to Z by using different HMM topologies with different states. Elmezain et al.[11] develop a system to recognize the English letters from A to Z and Arabic numbers from 0 to 9 in real-time from stereo color image sequences by using LRB (left-right banded) topology of HMM with 9 states.

In dynamic hand gesture recognition, the biggest challenge is the spatial-temporal variability of hand gesture, where the same gesture may differ in location, velocity, duration during operating. These characteristics make model itself more difficult to recognize dynamic hand gestures than to recognize static ones. In order to make the trained HMMs more adaptable to the unpredictable environment, we need to adjust the trained HMMs' parameters during recognition phrase. That is the concept of incremental learning we proposed in our system.

Incremental learning[12] is a machine learning paradigm where the learning process takes place whenever new example(s) emerge, and the machine learning model adjusts what has been learned according to the new example(s). The prominent difference of incremental learning from traditional machine learning is it does not assume the availability of a sufficient training set before the learning process, but modifies the models during this process.

In this paper, we firstly develop a system to capture gesturer's moving hand and use traditional left-right banded HMMs to recognize the vectors generated from these moving hand image sequences. Secondly, we design an incremental learning algorithm for the traditional HMMs (IL-HMMs) to make the models more adaptable to the unpredictable environment. In our system, the recognized hand gesture vector with high likelihood probability will be used to modify the parameters of the corresponding HMM to help model improve recognition rate.

Meng Hu (humengnju@sina.com), Furao Shen (frshen@nju.edu.cn) and Jinxi Zhao (jxzhao@nju.edu.cn) are with Robotic Intelligence and Neural Computing Laboratory (RINC.Lab), the State Key Laboratory for Novel Software Technology at School of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu, 210046, P.R.China. Corresponding author is Furao Shen.

II. FRAMEWORK OF IL-HMMs BASED HAND GESTURE RECOGNITION SYSTEM

The IL-HMMs based hand gesture recognition system proposed in this paper can recognize the motion trajectory of a single hand in real time. The system is divided into the following four parts (Fig. 1):

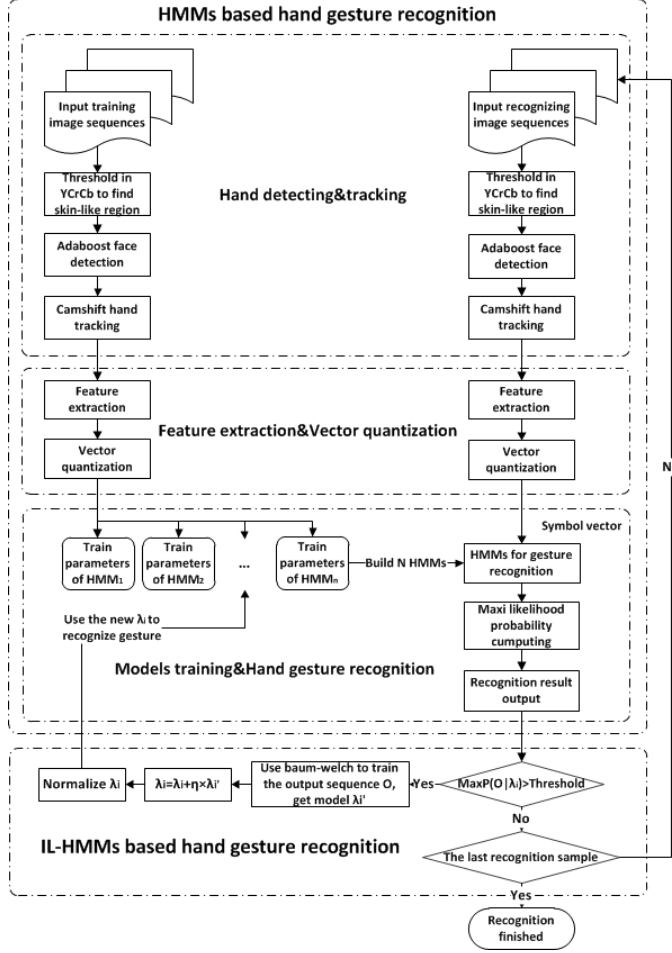


Fig. 1. The frame of IL-HMM based hand gesture recognition system

- *Hand detecting and tracking*: the gesture-operating hand which conveys motion information is detected, segmented, tracked and finally localized to generate its motion trajectory by using the combination of YCrCb color space with predefined threshold detecting, Adaboost face detection algorithm, Camshift motion tracking methods.
- *Feature extraction and vector quantization*: we select orientation information as features to model the hand gesture operating trajectory. The orientation features are determined between two adjacent centroid points of hand region from spatial-temporal motion trajectory. Then vector quantization is adopted to convert these high-dimension data into low-dimension ones that are used as input vectors for HMMs training and recognition.
- *Models training and hand gesture recognition*: before recognizing unclassified hand gestures, HMMs should

be built in training phrase. These different category models are trained by maximize the parameters of them to best describe corresponding input discrete vector sets. Testing data score against different trained HMMs, the model with highest output likelihood probability is determined as the recognized gesture category.

- *Incremental learning*: if the highest probability calculated in gesture recognition phrase is higher than the threshold predefined in our system, the recognized vector will be adopted to train a new HMM λ' for this vector. Then this new HMM λ' will merge into the corresponding category HMM λ by the predefined learning rate η to modify the parameters of λ . After that procedure, the modified HMM λ will be served as recognition model for the next gesture testing data.

III. HMMs BASED HAND GESTURE RECOGNITION SYSTEM

In this section, we depict the HMMs based hand gesture recognition system which contains the following three parts in Fig. 1.

A. Hand detecting and tracking

The proposed system uses an ordinary notebook camera to accomplish hand tracking and recognition, which means the color images only contain 2D information with X-axis and Y-axis coordinates, and do not carry depth information. We propose a combination of hand detecting and tracking algorithms, which can not only locate moving hand effectively, but also consider computing complexity.

1) *Predefined threshold in YCrCb color space for skin-like region detection*: Generally, human skin color is different from background, so the simplest way to find operating hand in color image is to set a threshold range which contains the human skin color scope. The frames we obtained from camera are in RGB color space, and we need to transfer them into YCrCb color space. The reason we select YCrCb as detecting color space[11][13] is human skin color has good clustering property in YCrCb, besides, we can easily split the lightness component interference in this space.

In YCrCb color space, Cb and Cr channels represent chrominance and Y channel refers to brightness. We ignore Y channel in order to reduce the effect of brightness variation and only use the chrominance channels which fully represents the skin color. The threshold range we applied is in [11] by GMM modeling. As a result, if the pixel color value lies in skin color range, it will be classified as skin pixel ($value=1$) or non skin pixel ($value=0$) otherwise. That is:

$$D_i(x, y) = \begin{cases} 1, & F_i(x, y) \text{ in threshold range} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Usually, skin color area is larger than noise one, and we need to set another area size threshold to segment the skin region. The threshold we applied is $\sigma \times S_f$, where S_f means the frame area and we set σ to 0.02 in our system. By finding and locating these skin regions, we can calculate the centroid point and draw the bounding box of them (Fig. 2).

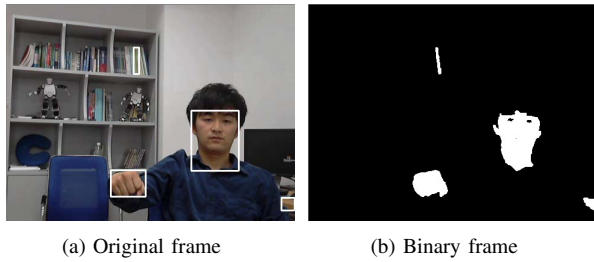


Fig. 2. Skin-like region detecting by predefined threshold in YCrCb

2) *Adaboost face detection for distinguishing hand and face*: Adaboost[14] being used in face detection is introduced by Paul Viola et al.[15]. After that, it has been widely used in industrial area, such as smiling face photographing, and becomes a quite mature application.

In our system, we use face features to build cascade classifiers to help us distinguish face region from images. The reason why do not choose hand features directly to build corresponding classifiers is even though the face has different facial expressions, it turns out to be a rigid, shape-unchangeable object. Consequently, the training of face detecting classifier is much easier than hand, which may change to different shapes and have various contours.

After detecting face successfully by Adaboost based classifier, we can delete face region from the skin-like region, thus, hand and face has been distinguished. If hand and face are overlapped, this situation will be divides into two cases:

- face can still be detected. Therefore, just detect and delete the face region;
- face can not be detected. It means the degree of hand and face overlap is very high, as a result, just regard this region as the hand region.

The result after using Adaboost algorithm for face detection is shown in Fig. 3.

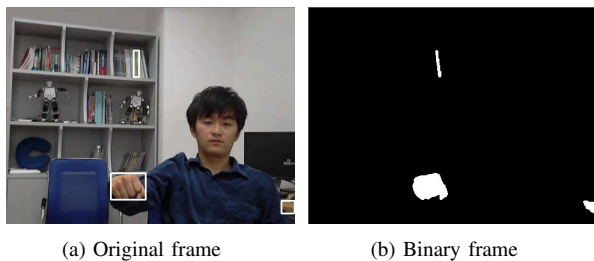


Fig. 3. Adaboost face detection for distinguishing hand and face

3) *Camshift auto-tracking for moving hand tracking*: In order to track the moving hand and reduce calculation, Camshift algorithm[16] is applied to determine a region of interest (ROI) surrounding the hand in each successive frame. The Camshift algorithm adjusts the search window size in the course of its operation. For each frame, the color probability distribution image is analyzed to determine the center and size of the ROI. The current size and location of the tracked object are reported and used to set the size and location of the search window in the next video image. The process is then repeated for continuous tracking.

The result of Camshift moving hand tracking is presented in Fig. 4.

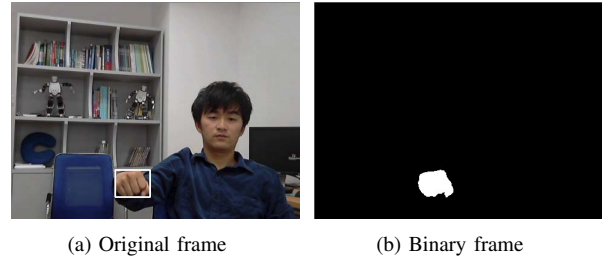


Fig. 4. Camshift auto-tracking for tracking moving hand

B. Feature extraction and vector quantization

Selecting effective features as the representation of hand gesture plays a significant role in models training and gesture recognition stage. There are three features: velocity, location and orientation[17], to be considered as the most important features in hand gesture recognition. In dynamic hand gesture, the gesture operating purpose is determined by the whole trajectory of hand movement. Besides, for the diversity and complexity of hand gesture, different people have different habits to operate them. If we take velocity and location features into consideration, we should set more restrictive conditions during operating. The orientation feature has proved to be the best local representation in terms of accuracy results, therefore, we will rely upon it as the main feature in our system.

As we have detected hand region in section III-A, the central point of hand region can be calculated by its information. The orientation feature of hand movement is computed between two consecutive centroid points of hand region in two adjacent frames from hand gesture trajectory:

$$\theta_t = \arctan\left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right), t = 1, 2, \dots, T - 1 \quad (2)$$

where T represents the length of hand gesture operating sequence. The feature vector will be determined by converting the orientation to directional codewords by a vector quantizer. In our system, the orientation is quantized to generate the codewords from 0 to 17 by dividing it by 20 degree (Fig. 5).

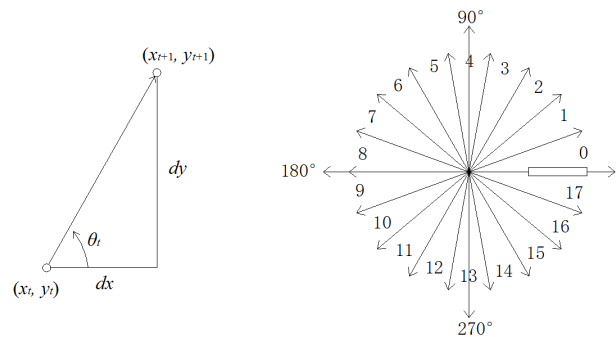


Fig. 5. Illustration of vector quantization

After above phrase, the extracted moving hand regions in the continuous image sequences are transformed into discrete symbol vectors. These quantized vectors will be served as training and recognition materials in next stage.

C. Models training and hand gesture recognition

HMM[7] is a statistical model which is widely used in hand writing recognition, part-of-speech tagging and gesture recognition applications for its capability of modeling spatial-temporal time-series data. An HMM λ is a doubly (observable and hidden state) stochastic model which contains a collection of states connected by transition matrices. Each HMM has a pair of transition matrices: state transition probability matrix A (which provides the probability for hidden state to hidden state transition) and an output probability matrix B (which defines the conditional probability of emitting an output symbol from a given hidden state). Besides, HMM has an initial probability matrix π for each hidden state to serve as initial state.

An HMM has three main functions:

- To compute the vector output likelihood probability by a given HMM and vector;
- To find the optimal hidden state path to present the output vector by a given HMM and vector;
- To modify the parameters of model that can best describe the given vector.

Generally, an HMM has two main topologies to present the model. In ergodic model, the state can reach every other state in a single transition. The other one is left-right banded model where states have no backward path. We select left-right banded model as our HMM topology, not only for the reason of the left-right banded model is good for modeling order-constrained time-series data, but also previous research[10] has proved the recommended model can reach the highest recognition rate.

1) *Hidden markov models training with multiple input vectors*: HMMs use baum-welch algorithm to modify the parameters of model to best describe the training vector. Baum-welch algorithm is a typical application of EM algorithm:

- The E-step: to compute the output likelihood probability $P(O|\lambda)$ by the training vector;
- The M-step: to re-estimate the parameters of hidden markov model $\lambda = (\pi, A, B)$ by using E-step probability.

The termination condition of baum-welch algorithm is $|P(O|\lambda') - P(O|\lambda)| < \varepsilon$, which means the subtraction of two output probabilities computed by two consecutive modified models is less than a predefined threshold value (usually very small). That implies the training procedure has already reached its best state to describe the vector.

For multiple vectors training ($O_i, i = 1, \dots, N$), LR. Rabiner[7] proposes a multi-sequence training algorithm by using the N observation vectors at each stage of the baum-welch re-estimation to iteratively update a single HMM parameter set. While R.I.A. Davis et al.[19] propose a class of new estimation methods where the baum-welch re-estimation

procedure is running separately on the N observation vectors, which means combining the trained HMMs directly and taking a weighted average of the models parameters. In our system, we simply assume that each training vector has the same weight in training phrase. Therefore, after N observation vectors being separately trained to generate N HMMs, we set an empty HMM λ_0 to store the sum of these HMMs corresponding parameters. And finally, the normalized λ_0 will be regarded as the presentation of these N observation vectors in training phrase.

2) *Hidden markov models recognition with input vectors*: We have two algorithms to recognize symbol vectors in HMMs: viterbi algorithm and forward algorithm. In our system, we apply forward algorithm to recognize these vectors. The difference between these two algorithms is :

- Viterbi algorithm is trying to find out the optimal hidden state path to generate the recognized vector. It is a recursive procedure by comparing the output probabilities to get the max value at vector length T , and recursively finding the hidden state path from T to 0 . Then the recognition module computes the probabilities of these optimal hidden state paths to generate output vector by multiplying corresponding factor of transfer matrix A, B in each HMM. The model with largest output probability will be regarded as recognition result.
- The other recognition method is to compute the probability sum of all the possible hidden state transfer paths to generate the output vector from vector length 0 to T . Fortunately, we can get the result much easier by using forward algorithm to avoid computing complexity.

From the probability calculation procedure, we come to conclude that viterbi algorithm is to find optimistic $1 - path$ hidden state sequence to generate the output vector, while forward algorithm is to enumerate $N^T - paths$ to get the probability (Fig. 6).

IV. IL-HMMs BASED HAND GESTURE RECOGNITION SYSTEM

In traditional HMMs algorithm, after the model being trained, the parameters of model are unchangeable during recognition. If the coming new gestures have some differences from the learned gestures, the recognition rate will decrease significantly. Modifying parameters to adjust to new gestures needs all vectors to be re-trained, which is quite time-consuming. An incremental learning method is promising to overcome the shortcomings found in traditional machine learning approaches. Incremental learning method consists of techniques to enable classifiers to gather more information from unseen data but do not forget old information, without having to access previously learned data[18].

In this paper, we add incremental learning thought into HMMs, and design a specific incremental learning algorithm for the traditional HMMs based hand gesture recognition system.

The incremental learning method we applied in our system is ensemble training (ET) and ensemble learning (EL)[18].

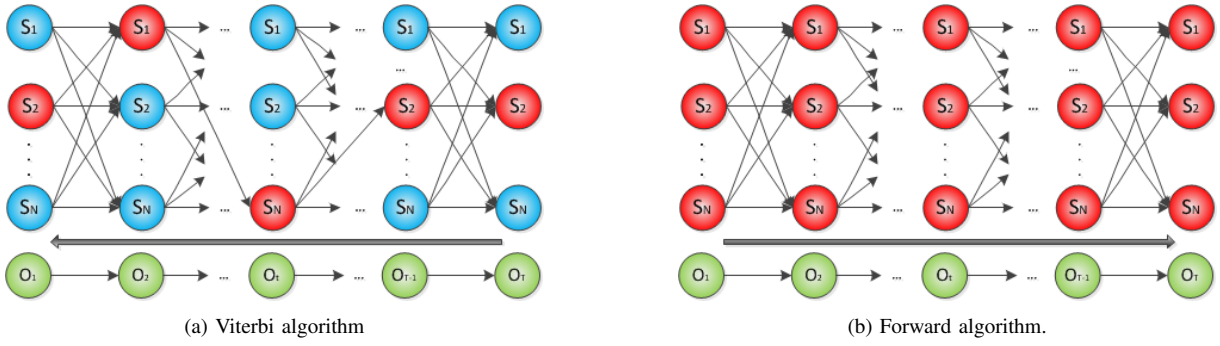


Fig. 6. Difference between viterbi and forward algorithms in hand gesture recognition. Red color state sequences infer the selected paths.

ET consists of independently do the learning of each of the observation sequences from the training set so that each sequence generates an HMM. After all the sequences are learned, the corresponding models are merged to generate a single model representing the whole training data. This training method has been described in the HMMs training phrase depicted in section III-C.

EL is adopted after the recognition of each vector. After recognizing an input vector, the vector will be assigned to train a new HMM if the recognized likelihood probability is smaller than the predefined threshold (Fig. 7). Given a old model $\lambda_{t-1} = (A_{t-1}, B_{t-1}, \pi_{t-1})$, which corresponds to all data up to the time step $t-1$, the new trained model $\lambda'_t = (A'_t, B'_t, \pi'_t)$, thus, we compute the ensemble incremental learning model $\lambda_t = (A_t, B_t, \pi_t)$ by using the following equations:

$$\bar{a}_{ij}^t = \frac{W_{t-1}a_{ij}^{t-1} + W'_t a'_{ij}{}^t}{W_{t-1} + W'_t} \quad (3)$$

$$\bar{b}_{ij}^t = \frac{W_{t-1}b_{ij}^{t-1} + W'_t b'_{ij}{}^t}{W_{t-1} + W'_t} \quad (4)$$

$$\bar{\pi}_i^t = \frac{W_{t-1}\pi_i^{t-1} + W'_t \pi'_i{}^t}{W_{t-1} + W'_t} \quad (5)$$

$$W_t = W_{t-1} + W'_t \quad (6)$$

In our system, we simplify the weight W_t and assume all the weights during EL is unchanged. Thus the above equations can be simplified as fellows:

$$\lambda = \lambda + \eta \times \lambda' \quad (7)$$

where η is the weight or the scale factors, we define it learning rate in our system, λ is the old model, λ' is the new model trained by the recognized vector. The learning rate η depicts the tradeoff between learning new knowledge and forgetting old knowledge. The larger η means the model is forgettable to old trained knowledge, and capable to adapt to the new environment quickly. The smaller η means that the model has the tendency to reserve the old trained knowledge more.

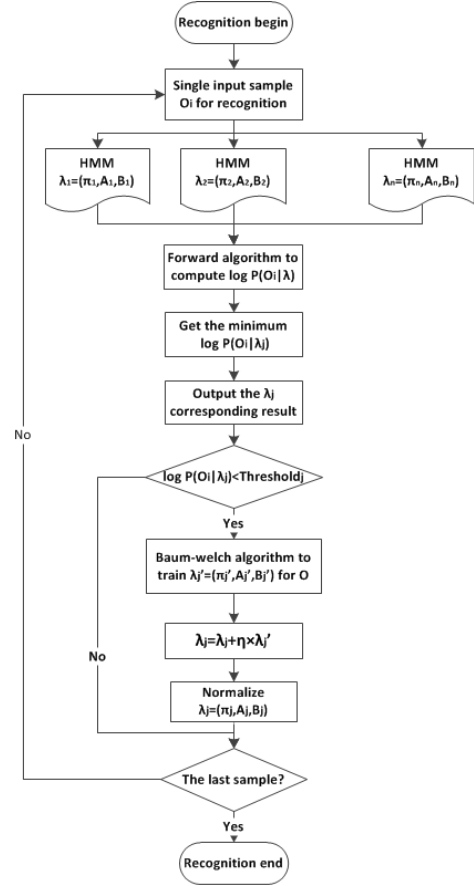


Fig. 7. IL-HMMs based hand gesture recognition in recognition phrase

The rest section we come to discuss about the threshold we predefined. On the one hand, by using the input training vectors to complete the models training, we use forward algorithm to compute the likelihood probabilities of the training vectors by the corresponding model. Actually, since the elements in matrix A and B is usually less than 1, the output probability computed by multiplying the elements dramatically decreases to exceed the precise range of computer. In order to handle the calculation, we modify forward algorithm and use scaling factor array $Scale[i]$ during computation to magnify the probability[7]. Finally,

the probability is calculated by using:

$$\log P(O|\lambda) = - \sum_{i=1}^N \log Scale[i] \quad (8)$$

After scaling, the output probability is within the range of computer, and can be used to generate the threshold in IL-HMMs. As we calculate the probabilities by using scaling forward algorithm in formula (8), the counting maximal probability is transformed into counting minimum value (Fig. 7, Algorithm 1). Besides, the judgement condition also transforms to the minimum probability not larger (Fig. 1), but smaller (Fig. 7) than threshold.

On the other hand, since even the same gesture may have different operating durations for gesturers' operating habit, the length of input vectors also differ from each other. Consequently, the likelihood probabilities may have order of magnitude difference after calculating. In order to measure the probabilities in a standard range, we need to normalize these probabilities. The normalize method we use is:

$$Threshold(i) = \kappa \times \left(\sum_{j=1}^N \log P(O_{ij}|\lambda_i) / length(O_{ij}) \right) / N \quad (9)$$

We divide the likelihood probability $P(O_{ij}|\lambda_i)$ by the length of the corresponding input vector symbols. After getting the sum of these results, we divide the sum by the number of these vectors. κ means the scaling factor, we set its value 0.8 in our experiment. That is, if the likelihood probability computed by the recognition phrase is smaller than the corresponding models threshold, we make the conclusion that the recognized vector have the reliability to modify the model to make itself more adaptive to the coming gesture. The judgment formula is:

$$\log P(O_{ij}|\lambda_i) < Threshold(i) \times length(O_{ij}) \quad (10)$$

The IL-HMMs procedure works according to Algorithm 1.

V. EXPERIMENT AND RESULT ANALYSIS

In our experiment, the gesturer, who uses a single hand to operate hand gesture, is standing before any stationary background with normal lighting. The gestures we recognized are the trajectories of Arabic numbers from 0 to 9, totally 10 gestures. We only use an ordinary notebook camera to capture gesturer's moving hand, and do not carry depth information during tracking. The size of each 2D color image frame is 480×360 , the video frame rate is 30 frames/sec, and each gesture takes from 1 to 3 seconds basing on the complexity of each gesture.

In HMMs training phrase, each gesture obtains 40 training image sequences to train corresponding model. The amounts of testing image sequences ranging from 71 to 171. The gesture samples "1" are used to verify the correctness of our system, we hence obtain much more samples comparing with other gestures (Table I). Besides, we intentionally change the

Algorithm 1 IL-HMMs based hand gesture recognition

Input: Vector set $\mathcal{D} = \{O_i, i = 1 \dots N\}$ for IL-HMMs based hand gesture recognition.

Output: Modified HMM λ .

- 1: Input a single vector O_i for recognition.
- 2: **while** O_i not the last sample in \mathcal{D} **do**
- 3: Compute $\log P(O_i|\lambda_j)$ by scaling forward algorithm, $j = 1 \dots M$.
- 4: Find the minimum $\log P(O_i|\lambda_j)$, get the corresponding model λ_j .
- 5: Return recognition result λ_j .
- 6: **if** $\log P(O_i|\lambda_j) < Threshold_i$ **then**
- 7: Train an new HMM λ'_j for O_i by baum-welch algorithm.
- 8: $\lambda_j = \lambda_j + \eta \times \lambda'_j$.
- 9: Normalize λ_j .
- 10: **end if**
- 11: **end while**

trajectory of gesture "0", "3", "6", "7", "8" gradually during operating in those testing samples. The changing trajectory gestures we selected have different operating complexity, so they can objectively prove the effectiveness and robustness of our proposed IL-HMMs based system.

TABLE I
AMOUNTS OF GESTURE IN TRAINING AND RECOGNITION PHRASE.

Gesture Type	0	1	2	3	4	5	6	7	8	9
Training Number	40	40	40	40	40	40	40	40	40	40
Testing Number	87	171	77	80	85	71	90	86	78	93

We define the recognition rate ψ as the amount of right recognition gesture divides total amount of the corresponding gesture. By using the traditional HMMs based hand gesture recognition system without incremental learning method, the recognition results are shown in Table II.

TABLE II
HMMs BASED HAND GESTURE RECOGNITION - RECOGNITION RATE.

Gesture Type	0	1	2	3	4	5	6	7	8	9
Testing Number	87	171	77	80	85	71	90	86	78	93
Recognized Number	80	156	75	62	82	71	78	75	72	92
Recognition Rate	0.91	0.91	0.97	0.78	0.96	1	0.87	0.87	0.93	0.99

We can find that, by using traditional HMMs based method, the recognition rate of each gesture can reach up to 80% (except gesture "3"). For the trajectory unchanged gesture "1", "2", "4", "5", "9", the recognition rate still remains a very high level, but for trajectory gradually changed gesture, the recognition rate is much lower comparing with the above trajectory unchanging gestures. That means the traditional HMMs based system is not robust in incremental situation.

Based on the different learning rates we apply, we list the recognition rate in Table III. From Table III we can find

TABLE III
IL-HMMs BASED HAND GESTURE RECOGNITION - RECOGNITION RATES VIA DIFFERENT LEARNING RATES.

Gesture Type η	Gesture Type									
	0	1	2	3	4	5	6	7	8	9
1.0	0.460	0.345	0.714	0.500	0.824	0.944	0.533	0.581	0.551	0.763
0.8	0.506	0.427	0.818	0.600	0.859	0.958	0.544	0.651	0.667	0.860
0.5	0.609	0.649	0.870	0.675	0.918	1.000	0.656	0.826	0.808	0.914
0.3	0.759	0.760	0.935	0.813	0.941	1.000	0.878	0.872	0.821	0.978
0.2	0.759	0.760	0.935	0.813	0.941	1.000	0.878	0.872	0.821	0.978
0.1	0.908	0.901	0.961	0.875	0.976	1.000	0.967	0.884	0.897	1.000
0.08	0.943	0.901	0.974	0.875	0.976	1.000	0.967	0.884	0.897	1.000
0.06	0.966	0.912	0.974	0.875	0.976	1.000	0.967	0.884	0.923	1.000
0.05	0.966	0.906	0.974	0.875	0.976	1.000	0.967	0.884	0.936	0.989
0.03	0.966	0.924	0.974	0.875	0.976	1.000	0.967	0.884	0.936	0.989
0.02	0.954	0.918	0.974	0.850	0.976	1.000	0.922	0.884	0.936	0.989
0.01	0.943	0.912	0.974	0.850	0.976	1.000	0.911	0.884	0.936	0.989
0.005	0.943	0.912	0.974	0.825	0.976	1.000	0.900	0.884	0.936	0.989
0.002	0.931	0.912	0.974	0.825	0.976	1.000	0.878	0.884	0.936	0.989
0.001	0.931	0.912	0.974	0.813	0.976	1.000	0.867	0.884	0.936	0.989

that different gestures can reach different recognition rates by using different learning rates. For trajectory unchanged gestures, since the trajectory remains stable while operating, we can just apply low learning rate to maintain the high recognition rate. For trajectory changed gestures, especially gesture “0”, “3”, “6”, if we set the learning rate to 0.03, the recognition rate can reach their highest level. What’s more, if we continue to increase the learning rate, the recognition rate turns to be lower, presenting the state of over-learning.

By using our proposed IL-HMMs based system, we can find that the trajectory unchanging gestures can improve their recognition rates slightly, or at least, remain their recognition rates in HMMs based system. For those trajectory changed gestures, the recognition rate improve significantly, proving that our proposed IL-HMMs based system is capable in facing gesture trajectory gradually changing situation.

Table IV and Table V depict the output probabilities of typical trajectory changed gesture “6” and trajectory unchanged gesture “9” when applying different learning rates. The output probability 67.9566 is the smallest value among all the HMMs output probabilities except gesture “6” (SPGE.6). And the output probability 102.04 is the smallest value among all the HMMs output probabilities except gesture “9” (SPGE.9). The second line is the output probabilities of the corresponding gesture “6” and “9” (PG.6, PG.9). According to Algorithm 1, the recognition result is the model corresponding to minimum output probability, therefore it would vary with different learning rates. We find that by applying different learning rates, PG.6, PG.9 gradually decrease and finally are smaller than SPGE.6, SPGE.9. Hence, the output results will change based on the specific learning rates. At last, we pick up the best recognition results based on proposed IL-HMMs based system which are shown in Table VI.

By comparing the recognition rate of HMMs and IL-HMMs based system (Fig. 8), we can finally conclude that the proposed IL-HMMs based hand gesture recognition can output the right result and reach better recognition rate.

TABLE VI
HMMs BASED HAND GESTURE RECOGNITION - RECOGNITION RATE.

Gesture Type	0	1	2	3	4	5	6	7	8	9
Testing Sample	87	171	77	80	85	71	90	86	78	93
Recognized Number	84	158	75	70	83	71	87	76	73	93
Recognition Rate	0.97	0.92	0.97	0.88	0.98	1	0.97	0.88	0.94	1

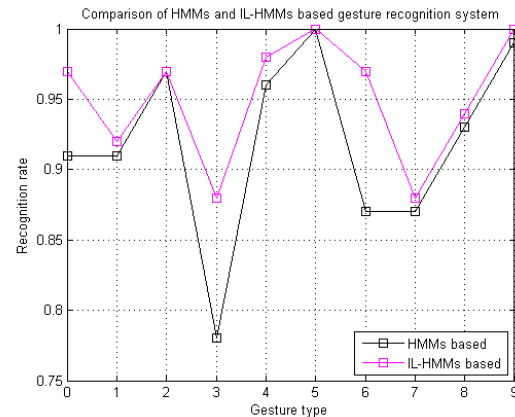


Fig. 8. Comparison of HMMs and IL-HMMs based gesture recognition system

VI. CONCLUSION AND FUTURE WORK

This paper proposes a hidden markov models based hand gesture recognition system with incremental learning method to recognize 10 different digit gestures (from 0 to 9) by the motion trajectory of a single hand. In the experiment, the proposed hand detecting algorithm is still very sensitive to light and needs some restrictive conditions in laboratory to ensure the tracking accuracy. As image processing methods combining with electronic sensors to detect moving hand are widely used in game industry nowadays, not only depth information, but also infrared sensor will be concerned to improve hand tracking accuracy in future work.

In this paper, incremental learning idea is applied to modify trained model parameters to make model adaptive to coming new gestures. This is the so called sample in-

TABLE IV
TRAJECTORY CHANGED GESTURE 6.

Learning Rate	0.002	0.005	0.01	0.03	0.05	0.06	0.08	0.1	0.2	0.3	0.5
SPGE.6	67.96	67.96	67.96	67.96	67.96	67.96	67.96	67.96	67.96	67.96	67.96
PG.6	70.62	69.63	68.83	67.55	67.02	66.84	66.52	66.33	81.66	83.81	87.58

TABLE V
TRAJECTORY UNCHANGED GESTURE 9.

Learning Rate	0.002	0.005	0.01	0.03	0.05	0.06	0.08	0.1	0.2	0.3	0.5
SPGE.9	102.04	102.04	102.04	102.04	102.04	102.04	102.04	102.04	102.04	102.04	102.04
PG.9	105.08	104.82	104.43	103.17	102.07	101.71	101.14	100.74	100.15	101.31	106.97

cremental learning, which means just modifying the existed models parameters, but not creating new models automatically. Another idea is the automatical class incremental learning by the existing samples. In our system, since the models' relationships are independent between each other, when it is necessary to add a new gesture into the system, we need to manually train another HMM for the new category gesture. In our future work, we will consider class incremental learning idea to automatically generate new model for the new hand gesture category.

ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation of China under Grant Nos. (61375064, 61373001, 61321491), 973 program (2010CB327903), and Jiangsu NSF grant (BK20131279, BK2011567).

REFERENCES

- [1] T. Baudel and M. Baudouin-Lafon, "Charade: remote control of objects using free-hand gestures," *Comm. ACM*, vol. 36, no. 7, pp. 28-35, 1993.
- [2] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on System, Man, and Cybernetics*, vol. 37, no. 3, pp. 311-324, 2007.
- [3] D.J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics and Applications*, vol. 14, pp. 30-39, 1994.
- [4] X. Zhang, X. Chen and Y. Li, "A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, iss. 6, pp. 1064-1076, 2011.
- [5] O. Al-Jarrah and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems Artificial Intelligence," *Artificial Intelligence*, vol. 133, iss. 1-2, pp. 117-138, 2001.
- [6] X. Deyou, "A neural network approach for hand gesture recognition in virtual reality driving training system of SPG," *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 519-522, 2006.
- [7] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc.of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.
- [8] J. Yamato, J. Ohya and K. Ishii, "Recognizing Human Action in Time-Sequential Images using Hidden Markov Model," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379-385, 1992.
- [9] H. Lee, and J. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21(10), pp. 961-973, 1999.
- [10] L. Nianjun, C.L. Brian, J.K. Peter and A.D. Richard, "Model Structure Selection and Training Algorithms for a HMM Gesture Recognition System," *International Workshop in Frontiers of Handwriting Recognition*, pp. 100-106, 2004.
- [11] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences," *The Journal of WSCG*, vol. 16(1), pp. 65-72, 2008.
- [12] R. Polikar, L. Udpa, S. Udpa, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 497-508, 2001.
- [13] S.L. Phung, A.Bouzerdoun, and D. Chai, "Skin segmentation using color pixel classification: analysis and comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, iss. 1, pp. 148-154, 2005.
- [14] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148-156, 1996.
- [15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 511-518, 2001.
- [16] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, iss. 5, pp. 603-619, 2002.
- [17] H. Yoon, J. Soh, Y.J. Bae, and H.S. Yang, "Hand Gesture Recognition using Combined Features of Location, Angle and Velocity," *Pattern Recognition*, vol. 37(7), pp. 1491-1501, 2001.
- [18] P.R. Cavalin, R. Sabourin, and C.Y. Suen, "Evaluation of incremental learning algorithms for HMM in the recognition of alphanumeric characters," *Pattern Recognition*, vol. 42, no. 12, pp. 3241-3253, 2009.
- [19] R.I.A. Davis, B.C. Lovell, and T. Caelli, "Improved estimation of hidden Markov model parameters from multiple observation sequences," *Proceedings of the International Conference on Pattern Recognition*, vol. 2, pp. 168-171, 2002.