

MIXED HAND GESTURE RECOGNITION SYSTEM AND ITS APPLICATION

SHUAI JIN^{1,2}, ZHONG YANG^{1,2}, YANG ZHENG^{1,2}, YI LI^{2,*}
WEIDONG CHEN² AND XIAOXIANG ZHENG²

¹College of Computer Science

²Qiushi Academy for Advanced Studies

Zhejiang University

No. 38, Zheda Road, Yuquan Campus, Hangzhou 310027, P. R. China

{ ShineDSK; jason.yang1228; xdzy2426 }@gmail.com

*Corresponding author: liyi9810857@gmail.com

chenwd@zju.edu.cn; zxx@mail.bme.zju.edu.cn

Received August 2012; revised January 2013

ABSTRACT. *Natural gestures are always mixtures of both dynamic and static gestures, so are gesture interfaces for human-computer interaction. This makes recognizing mixed gestures important. Since there has been little work focusing on this issue, this paper is devoted to spotting and recognizing from mixed gesture sequences. The main work of this paper falls into two parts. The first is to distinguish meaningful gestures from random sequence using state-based spotting algorithm. The second is to further classify gestures using different features. Experimental results and practical application show that this system is effective and efficient for low-cost gesture recognition.*

Keywords: Hand gesture recognition, Mixed hand gesture, Spotting algorithm, Data aligning, Human-computer interaction

1. Introduction. Hand gestures provide an important complimentary modality for speech to make ourselves understood in daily communication and interaction with humans. Sometimes, information and ideas like degree, discourse structure, spatial and temporal structures conveyed by hand gestures may not be easily carried using other modes [1]. Especially for people with language disabilities, hand gestures become the most reliable and manageable way for communication. In designing natural and robust human-computer interfaces, expressiveness and naturalness are considered to be the two key elements, but are often missing from interfaces using other modes. According to Wachs et al. [2], gesture interface presents three main advantages over conventional HCI (Human-Computer Interaction) systems, namely accessing information while maintaining total sterility, overcoming physical handicaps and exploring big data. Today, gesture interfaces have been used in a wide range of applications including medical, rehabilitation, entertainment, military, education and so on [3].

Gestures used in daily life are rapid, continuous and highly free-form [4], and can be varied according to the meaning conveyed and the people conduct it. Generally speaking, hand gestures are primarily classified as static and dynamic. Static gestures are usually still hand postures or configurations, while dynamic gestures are made up of continuous hand movements and posture formation. People are able to move fluidly among these two types as needed. However, in most of current human-computer interaction scenarios, only static or dynamic gestures are involved. These systems only make a one-to-one mapping from either hand posture configuration or trajectories to perform some single command, reducing the power of expressiveness and naturalness of gesture interfaces.

Compared with the large number of studies recognizing static or dynamic gestures, there are only a few work of identifying both gestures from sequences for continuous interaction [5]. This mixed style of both static and dynamic gestures in one sequence is very common and important in natural interaction and communication. Considering the action of “shaking hands”, first reach out the hand, open it and keep still for some seconds waiting for another hand, then close it. This sequence is composed of both dynamic and static gestures. Dynamic gestures are continuous hand movements of reaching out the hand and posture formation of both opening and closing hand, and static gesture is keeping the open hand still waiting. Identifying these static and dynamic gesture information from this mixed sequence would make sense for building natural interfaces to increase expressiveness in communicating with computers.

Hand gesture recognition systems aim to tracking human gestures, identifying gestures as input and processing these representations through mapping of commands to output. There are two major types of solutions for gesture interfaces, namely glove-based and vision-based systems. Among them, markless vision-based systems provide a non-intrusive solution for gesture data acquisition at a low cost with passive and stealthy sensing. There are excessive work on static gesture recognition since the early 90’s for its relative simplicity [6], but less work on dynamic gesture recognition with most focus on trajectory [7] and very few on hand posture formation. And there still lacks work of identifying static and dynamic gestures from mixed sequence.

This paper deals with the problem of recognizing mixed static and dynamic gestures from a randomly occurred gesture sequence using markless vision-based method. First, a state-based spotting algorithm are presented to automatically identify static and dynamic gestures from the sequence. Then static gestures and two different types of dynamic gestures are further recognized and classified. Finally, a framework is provided for building low-cost online interaction systems using this mixed gesture sequence, and the application of interacting with virtual apartment is also presented for illustration.

2. Related Work. Markless vision-based gesture recognition systems have found vital applications across a wide range of scenarios, from traditional virtual reality, sign language and robotics to more recent domains as video games, medical environment, augmented reality, etc. However, since static and dynamic hand gestures have different features, they have to be treated using different methods and algorithms.

Static gestures only involve orientation, position and still posture information of the hand without any movement, and they can be detected using general classifier or template-matcher. Birk et al. used principle component analysis and Bayes classifier for constructing a mapping between hand postures and alphabet [8]. Triesch and Malsburg used elastic graph matching for hand posture recognition even under complex environments without segmentation [9]. Zhang et al. used an adaptive complexion model and voting theory for rapid static hand posture recognition [6].

As the information related to static hand gesture is limited, only one-to-one mapping from gestures to control and interaction commands can be built using these information; thus the ability and extent of such gesture control are constrained. Static gestures are often involved in simple control scenarios, as for desktop applications controlling cursor or virtual keyboards [10], for simple sign language translation like single alphabet or digit [11], for multimedia manipulation like turning media player on or off, for game control like moving or shooting, for virtual environment navigation and object manipulation like moving forward and pointing [12].

Unlike static gestures, dynamic gestures have temporal and spatial aspects, with hand body moving or posture formation, and usually are done intentionally for communication. Most often, dynamic gestures can convey emotion or intention and are direct translation of short oral language. These aspects make dynamic gestures recognition more complex than that of static gestures. The temporal or spatial dimension can be handled by modeling through hand gesture representation as done using motion based model. On the other hand, this dimension can also be handled using automata-based learning algorithms as most common approaches have done. These approaches treat dynamic gestures as a set of states and transitions within states. Thus dynamic gestures are considered as a path between an initial and a final state. Lee and Kim propose an approach using Hidden Markov Models (HMM) based threshold for gesture recognition [13]. Patwardhan and Roy use a predictive eigen tracker for recognizing dynamic gestures involving changing shapes and trajectories [14].

With the additional temporal and spatial dimension, the effective commands dynamic gestures can conduct are more and richer than static gestures, and these commands are not limited to one-to-one mappings. In game control, dynamic gestures allow the games to be controlled physically just as we play in the real world [15], in virtual environment, involving dynamic gestures means navigation and interaction with the environments in a more natural way as interacting with real objects [16]. In daily life, gesture sequence is often a mixture of random occurred static and dynamic gestures, without clear boundaries. Though a plenty of work have involved solely static or dynamic gestures as interfaces for HCI, few of them have ever tried using both gestures at the same time. The main reason for this may due to the different features involved in static and dynamic gestures which require different recognition methods. It is also still difficult to spot them separately.

On mixed gestures recognition, Chen et al. raised a two-level approach for gesture category differentiation but with a cumbersome architecture to implement [17]. This mixture of gestures can be used as smart environment controller [18], real-time TV and set-top-box controller [19], virtual object manipulation [20]. Both the quality and quantity of work on mixed gesture recognition are far less than static and dynamic gesture separately. This is mainly due to the fact that splitting and spotting from a continuous gesture sequence still remains a problem for effective and robust recognition using markless vision-based method.

Based on these facts, most of our work is devoted to the spotting algorithm automatically identifying static and dynamic gestures from a mixed sequence. Then both gesture classes are further recognized using separate methods and algorithms.

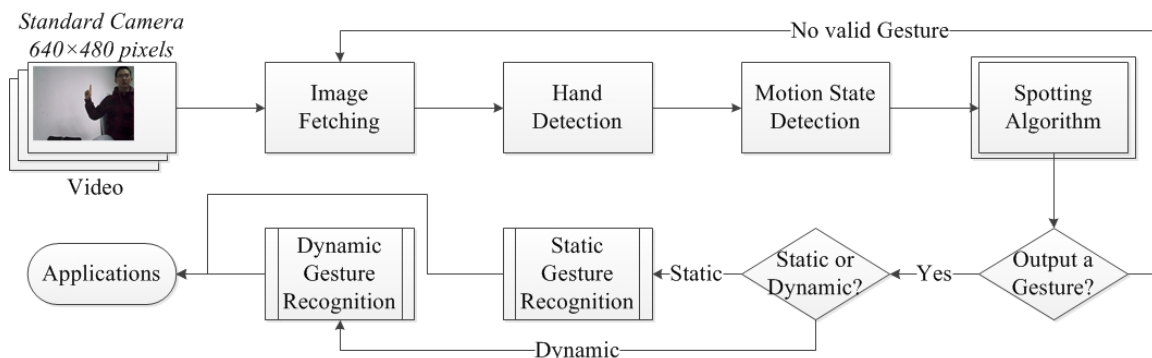


FIGURE 1. Structure overview of recognition system

3. System Design and Implementation. The overall structure of the mixed hand gesture recognition system is shown in Figure 1. The system contains three individual modules, namely, gesture spotting, static gesture recognition and dynamic gesture recognition. Gesture spotting module first detects the valid hand area from gesture image sequence, then a state-based spotting algorithm is used to segment gestures and mark them static or dynamic. Static gestures are recognized using voting theory and relief algorithm in static gesture recognition module, and dynamic gestures are recognized using Hidden Markov Models (HMMs) in dynamic gesture recognition module. The system design makes it distinctive from solely static or dynamic gesture recognition systems in recognizing from a randomly mixed gesture sequence with reasonable performance.

3.1. Gesture spotting. In a mixed gesture image sequence, the static and dynamic gestures always occur randomly, together with other meaningless actions as hands move-in and move-out which should be ignored. Static gestures can be represented using single images while dynamic gestures must be represented using series of images. This module of gesture spotting is thus provided to extract meaningful gestures from this random sequence, and mark them as static, dynamic or meaningless. This is a two-stage process. First, hand area must be extracted from each frame image of the sequence and hand motion state is detected. Second, a stage-based spotting algorithm is applied to determine whether the hand is in static or dynamic gesture state.

Hand detection from gesture image sequence is following the common detection procedure. YCbCr skin-color model is used to extract valid hand area and remove unnecessary background information, and the result is a binary image sequence of hand area as shown in Figure 2(b). The red rectangle denotes the valid hand area segmented from raw image of Figure 2(a).

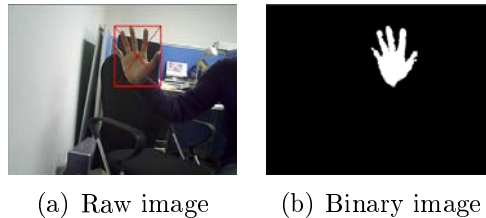


FIGURE 2. Hand detection result

By performing context-sensitive state analysis on the image sequence, each frame image is determined as whether it is in moving state or in still state, and this procedure is called motion state detection. To determine the motion state of each frame image, two factors are considered, namely change of hand location and variation of hand shape.

The change of hand location is calculated using Euclidean distance:

$$\Delta d = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (1)$$

The pair (x_i, y_i) is the hand location in the i th frame.

The variation of hand shape is approximated using variation of the rectangular region:

$$Deltas = |A_i - A_{i-1}| \quad (2)$$

A_i is the area of rectangular region in i th frame.

Motion state of each frame image is then determined as:

$$r = \begin{cases} 1, & \text{if } (\Delta d \geq \delta_1) \text{ or } (\Delta s \geq \delta_2) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where δ_1 and δ_2 are thresholds for filtering jitters in hand motion detection. Here, $r = 1$ means the frame image is in moving state, and $r = 0$ is part of a still one.

Algorithm 1 State-Based Spotting Algorithm

Require:

Define $n_0 = 0$, $n_1 = 0$, $flag_1 = 0$, $flag_2 = 0$; Create an empty list L for storing possible dynamic gestures.

Ensure:

```

1: for all image  $I_t$  do
2:    $f_1 = 0$ ,  $f_2 = 0$ ;
3:   Run hand segmentation and motion detection to get a result  $r$ ;
4:   if  $r = s_0$  then
5:      $n_0 ++$ ;
6:     if  $n_0 \geq N$  then
7:        $f_1 = 1$ ,  $n_0 = 0$ ;
8:     end if
9:   else
10:    if  $r = s_1$  then
11:       $n_1 ++$ ;
12:      if  $n_1 \geq N$  then
13:         $f_1 = 1$ ,  $f_2 = 1$ ,  $n_1 = 0$ ;
14:      end if
15:    end if
16:    else
17:      if  $r = s_2$  then
18:        Insert  $I_t$  to the back of  $L$ ;
19:         $n_0 = 0$ ,  $n_1 = 0$ ;
20:      end if
21:    end if
22:     $n = length(L)$ ;
23:    if  $f_1 = 1$  then
24:      if  $n \geq N$  then
25:        Output  $L$  as a dynamic hand gesture;
26:      end if
27:      Clear  $L$ ;
28:    end if
29:    if  $f_2 = 1$  then
30:      Output  $I_t$  as a static hand gesture;
31:    end if
32:  end for

```

After hand detection and motion state detection, each frame image must be in one of the three states: s_0 still hand, s_1 no hand and s_2 moving hand. It is obvious that a series of continuous frame images with state s_2 must be partial or complete of a dynamic gesture. Some single frame image with state s_0 must be static gesture.

However, spotting based on just states of single frames is not exact enough. There may be careless movement when performing a static gesture making state changing from s_0 to s_2 immediately, causing separate gestures spotted as a single one. If the interval between gestures in a sequence is not long enough, multiple gestures are also probably spotted as a whole. When performing a dynamic gesture, there may be tiny pause causing the gesture

divided into two separate gestures. It is also likely to spot some meaningless gesture as an independent one. Frames of these gestures must be handled to reduce the influence on motion state detection results and increase robustness of the method.

For the sake of enhancing robustness of the spotting algorithm, not to recognize these meaningless gestures, a threshold N is set as in Algorithm 1, which is related to HMMs to separate these undefined gesture frames from defined gestures, and filter these frames. Steps 6, 12 and 25 ensure this process.

3.2. Static gesture recognition. Since static gestures are relatively simple for recognition and there have already been excessive work on it, a simple and rapid algorithm proposed by Zhang et al. [6] is involved for static gesture recognition. This method is based on voting theory and relief algorithm.

This algorithm requires off-line learning process for recognition. After image normalization to a uniform size of 160×120 , each valid image is represented as a column vector V . A mean vector \vec{D}_k is calculated for each gesture category k as:

$$\vec{D}_k = \frac{1}{n} \sum_{i=1}^n \vec{V}_i \quad (4)$$

with n the sample number and the i th column vector \vec{V}_i denoting the corresponding posture.

To recognize an unknown static gesture \vec{G} , a score for each category is counted using the following rule:

$$s_{k,i} = \begin{cases} D_{k,i}, & G_i = 1 \\ 1 - D_{k,i}, & G_i = 0 \end{cases} \quad (5)$$

Here, G_i is the i th element of vector \vec{G} . If $G_i = 1$ means G_i corresponding to a skin-like pixel in raw image. $D_{k,i}$ is the i th element of vector \vec{D}_k , and $s_{k,i}$ is the score of putting G_i in category k . The total score of each category is represented as $s_k = \sum s_{k,i}$. To filter some undefined gestures in similarity computing, a threshold is also set up. The unknown gesture is then classified into the category with the highest score and a gesture score under the threshold would be discarded.

3.3. Dynamic gesture recognition. Different from static gestures usually with single image, dynamic gestures always contain continuous images. Hidden Markov Model (HMM)-based approaches are employed for recognizing such gestures. Feature vector is important for training and learning of such models. Here, a feature vector including position, velocity, size and shape of hand is provided for HMMs training.

3.3.1. Feature extraction. Position and velocity are the two most significant features of hand trajectory tracking and are both described as two-dimensional coordinates. Size and shape information are also important features describing current posture of the hand. All these features can be combined together as a six-dimensional feature vector for HMM learning and training.

- **Hand Position Description.** Hand position feature $(P_{x,t}, P_{y,t})$ measures relative difference between current hand location and centroid of the gesture hand (C_x, C_y) , calculated as:

$$(C_x, C_y) = \left(\frac{1}{n} \sum_{t=1}^n X_t, \frac{1}{n} \sum_{t=1}^n Y_t \right) \quad (6)$$

The hand position feature is then represented as:

$$(P_{x,t}, P_{y,t}) = (X_t - C_x, Y_t - C_y) \quad (7)$$

where (X_t, Y_t) is the hand location in current t th frame as shown in Figure 2(a).

- **Hand Velocity Description.** The hand velocity feature $(V_{x,t}, V_{y,t})$ is the change of speed and direction of hand motion. It measures hand position change between current and previous frames as:

$$(V_{x,t}, V_{y,t}) = (X_t - X_{t-1}, Y_t - Y_{t-1}) \quad (8)$$

- **Hand Size Description.** The hand size feature S_t is defined as area of bounding box of the hand (see Figure 2(b)). This feature is used as an approximate of hand depth, calculated as:

$$S_t = A_t - A_1 \quad (9)$$

where A_t and A_1 correspond to area of bounding box at t th and 1st frames.

- **Hand Shape Description.** The hand shape feature R_t is introduced for posture variation monitoring. Unlike other features, this feature is rapid and convenient to get. This representation helps to express the rich varieties of hand postures with a powerful performance.

The six-dimensional feature vector is then represented as:

$$\vec{F}_t = \langle P_{x,t}, P_{y,t}, V_{x,t}, V_{y,t}, S_t, R_t \rangle \quad (10)$$

with each component normalized to a real number within the range of $[-1.0, 1.0]$ for training.

3.3.2. Data aligning algorithm. HMMs only accept data with the same fixed length as training data set. However, even for one participant, the same dynamic gesture may take different time to complete. This results in varied length of raw gesture data sequences. A solution to normalize the sequence length is data aligning. Algorithm 2 shows the detail of this procedure.

First, the average length of samples in the gesture is calculated. Then, each sample is checked to see whether the length is equal to the average or not. Shorter samples are expanded and longer samples are compressed to average length both using linear interpolation. This method enables participants with a higher degree of freedom to conduct natural gestures with few constraints.

4. Experimental Results. Three experiments and one demo application are designed to evaluate the performance of the proposed system. Six people (two trained, four untrained) are involved. Totally 11 gestures are defined with 3 static gestures and 8 dynamic gestures, as in Figure 1. These dynamic gestures include 4 with two dimensional trajectory changing, 2 with posture formation and 2 with depth changing.

Experiment I is designed to evaluate the overall recognition performance of the system over mixed gesture sequences with randomly occurred gestures. Experiment II specifically deals with the performance of spotting algorithm on different sequences. Experiment III evaluates and analyzes the system performance on a practical application of navigation and interaction in a virtual apartment using gestures.

These experiments are carried out on a computer with Intel® Quad Q6600 2.4GHz CPU, 4GB RAM, NVIDIA® GeForce GTS 450 graphics card. Gesture data were captured using an ANC® web camera. System and demo application are developed using Visual C++ 2008 and OpenCV library under Microsoft® Windows 7 operating system. Experiment data are processed and analyzed using Matlab® with a ready-made HMM toolbox written by Murphy [21].

Algorithm 2 Data Aligning Algorithm

Require:

Calculate the average length \bar{L} for the training gesture:

$$\bar{L} = \frac{1}{n} \sum_{i=1}^n |F_i|$$

where n is the sample count, F_i stands for one sample, $|F_i|$ means the length of F_i .

Ensure:

1: **for all** F_i **do**

2: Define F_i as

$$F = \{\vec{F}_1, \vec{F}_2, \dots, \vec{F}_n\};$$

3: Define an array D to record the distance between adjacent vectors in F , where

$$D[i] = \|\vec{F}_{i+1} - \vec{F}_i\|, \quad 1 \leq i \leq n - 1;$$

4: **while** $length(F) > \bar{L}$ **do**

5: Select the minimum value $D[k]$ from D ;

6: Define \vec{F}' where

$$\vec{F}' = \frac{(\vec{F}_k + \vec{F}_{k+1})}{2};$$

7: Delete \vec{F}_k and \vec{F}_{k+1} ;

8: Insert \vec{F}' to the position k of F ;

9: Update array D ;

10: **end while**

11: **while** $length(F) < \bar{L}$ **do**

12: Select the maximum value $D[k]$ from D ;

13: Define \vec{F}' as Step 6;

14: Insert \vec{F}' to the position $k + 1$ of F ;

15: Update array D ;

16: **end while**

17: **end for**

4.1. Experiment I: Hand gesture recognition. This experiment includes two parts. The first part is to evaluate the applicability of HMMs on 8 dynamic gestures. HMMs need to be trained repeatedly to achieve optimal parameters and good likelihood. The experiment result is shown in Table 2. The column \bar{L} denotes average length of each dynamic gesture, M and Q are parameters for each HMM corresponding to state number and mixture number. The results reveal that HMMs are reliable with high likelihood using these parameters after training.

The second part is the result of hand gesture recognition. This is further divided into two tests. The first test takes samples from 6 people (totally $6 \times 11 = 5280$ samples) as training set, and evaluates the recognition rate on gesture data taken from 6 people (totally $6 \times 50 \times 11 = 3300$ gestures). The second test takes samples from 3 people (totally $3 \times 80 \times 11 = 2640$ samples) as training set, and evaluates the recognition rate on gesture data taken from another 3 people (totally $3 \times 30 \times 11 = 990$ gestures). The results of these two tests are shown in Table 3.

The first three rows in Table 3 correspond to recognition rate of static gestures, with the rest of dynamic gestures. The overall performance of the first test is over 95% while the second is averaged at 85%. As in the second test, the recognition result is based on some generic training set, the overall performance is lower than the first test, but still within

TABLE 1. Gestures defined in the system





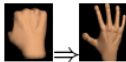










Gesture	Posture Variation	Trajectory	Description	Category
Forward		None	Move forward	static
Backward		None	Move backward	static
Stop		None	Stop moving	static
Grasp		None	Grasp object	deform
Release		None	Release object	deform
Turn Left			Turn left	vertical
Turn Right			Turn right	vertical
Turn On			Turn on	vertical
Turn Off			Turn off	vertical
Push		Forward	Push away	depth
Pull		Backward	Pull back	depth

TABLE 2. \bar{L} , M and Q for dynamic gestures

Gesture	\bar{L}	M	Q
Turn Left	11	4	4
Turn Right	13	2	3
Grasp	5	3	4
Release	4	2	2
Push	12	6	5
Pull	9	4	4
Turn On	18	3	2
Turn Off	19	6	4

\bar{L} : Average length of gestures.

M : The state number.

Q : The mixture number of Gaussians.

a reasonable scope. This shows the applicability of the system. The general recognition rate of static gestures is between the best and worst of dynamic gestures. This reveals the fact that the trajectory feature has a greater influence on recognition results than posture feature. The recognition on dynamic gestures ‘Grasp’ (87.00%) and ‘Release’ (89.00%) are relatively lower than average, and this is mainly due to the fact that these two gestures only have tiny depth variation, with almost no change of hand posture and trajectory.

TABLE 3. Recognition results in Experiments I and II

Gesture	Experiment I				Experiment II			
	Samples	Correct	Error	Recognition (%)	Samples	Correct	Error	Recognition (%)
Forward	300	291	9	97.00%	90	80	10	88.89%
Backward	300	293	7	97.67%	90	82	8	91.11%
Stop	300	297	3	99.00%	90	86	4	95.56%
Grasp	300	261	39	87.00%	90	70	20	77.78%
Release	300	267	33	89.00%	90	68	22	75.56%
Turn Left	300	296	4	98.67%	90	82	8	91.11%
Turn Right	300	299	1	99.67%	90	84	6	93.33%
Turn On	300	299	1	99.67%	90	83	7	92.22%
Turn Off	300	289	11	96.33%	90	78	12	86.67%
Push	300	278	22	92.67%	90	81	9	90.00%
Pull	300	282	18	94.00%	90	77	13	85.56%
total	3300	3152	148	95.52%	990	871	119	87.98%

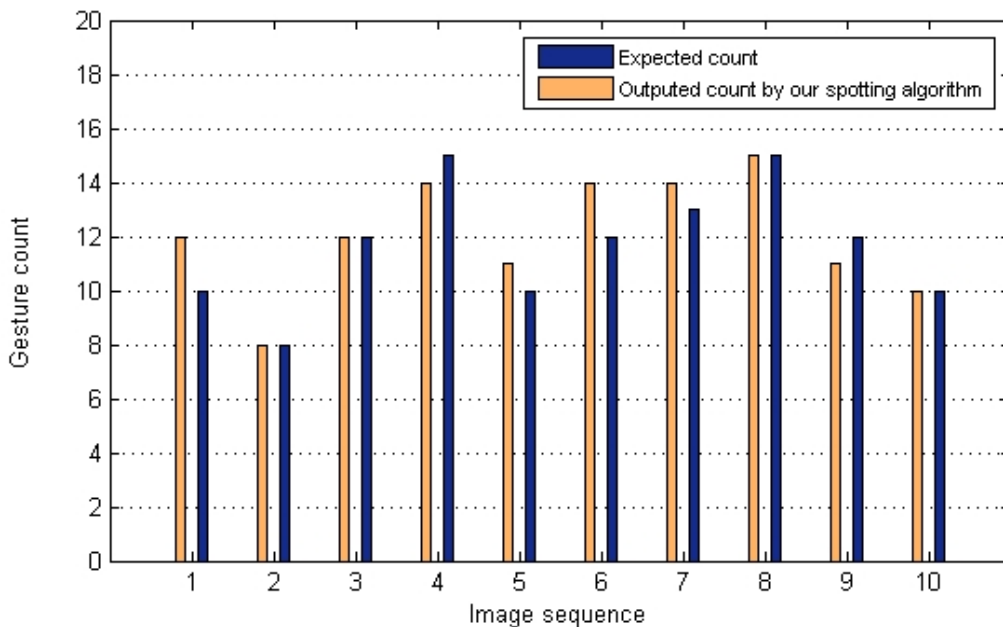


FIGURE 3. Tests on spotting algorithm

4.2. Experiment II: Gesture state spotting. This experiment aims to evaluate the robustness and efficiency of the spotting algorithm. Gesture sequences defined for evaluation consist of alternatively appearing random dynamic and static gestures with different lengths. 10 such sequences are included in experiment with the expectation to ignore meaningless postures and mark meaningful dynamic gestures. The results are shown in Figure 3. The results show that, the spotting algorithm output the same number as expected counts for 4 sequences, output less for another 4 sequences and more for 2 sequences.

With the threshold, recognition results of 6 dynamic gestures are shown in Figure 4. Among the 6 gestures, gesture 1, 2 and 5 are defined and gesture 3, 4 and 6 are undefined.

Through the evaluation of trained HMMs, each gesture has eight loglikelihood scores. A higher score indicates a higher similarity to that gesture. The curves show that defined gestures have excellent similarities to the correct category with a higher score, while the undefined gestures get much lower score to each category. A score to some category

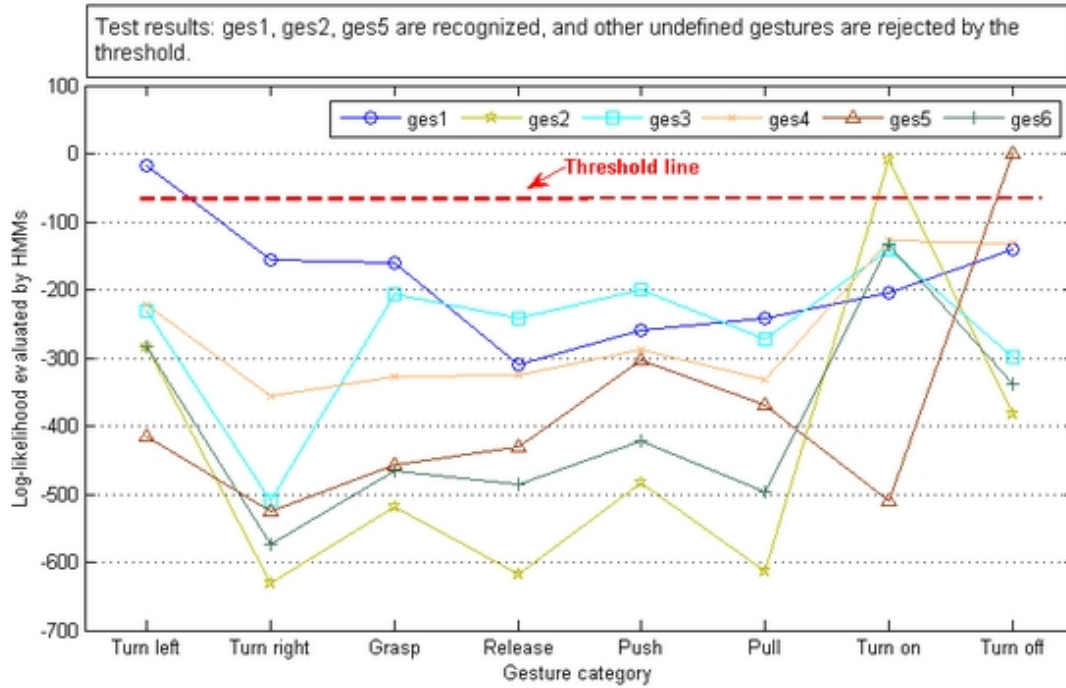


FIGURE 4. The recognition results on some defined or undefined dynamic gestures

above the threshold is accepted, while under the threshold is rejected. To find the optimal threshold, about tens of trials are taken, and the value is found to be set at -80.0 .

4.3. Experiment III: Demo application and analysis. Gesture interfaces came into being for solving practical problems. Gesture data sequences for recognition in practical applications are different from performance testing. For testing, gesture sequences may be predefined, participants may be trained for a long time, numerous trials may be taken for better statistical results. While in practical applications, gesture sequences are defined according to task sequences, most participants are naive users of the system, these gestures may be conducted a few times or even once.

Hence, to test the effectiveness and efficiency of the gesture spotting and recognition methods, a practical gesture interface for navigation and interaction in a virtual apartment is developed. Gestures data sequence is captured as video and analyzed on-line using a common web camera, with a resolution of 640×480 at a frame rate of 20fps. The distance between participant and camera is between 0.5m and 1.0m. System configuration is the same as used for previous experiments.

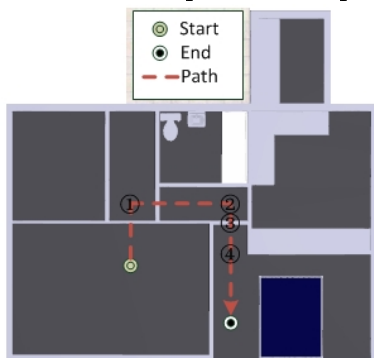


FIGURE 5. Task route

TABLE 4. Corresponding gestures defined on task route

Task	Gestures	Category
① Turn right	Turn Right	trajectory
② Turn right	Turn Right	trajectory
③ Pause	Stop	static
③ Open door	Push	depth
④ Hold switch	Grasp	shape
④ Turn on	Turn On	trajectory
④ Turn off	Turn Off	trajectory

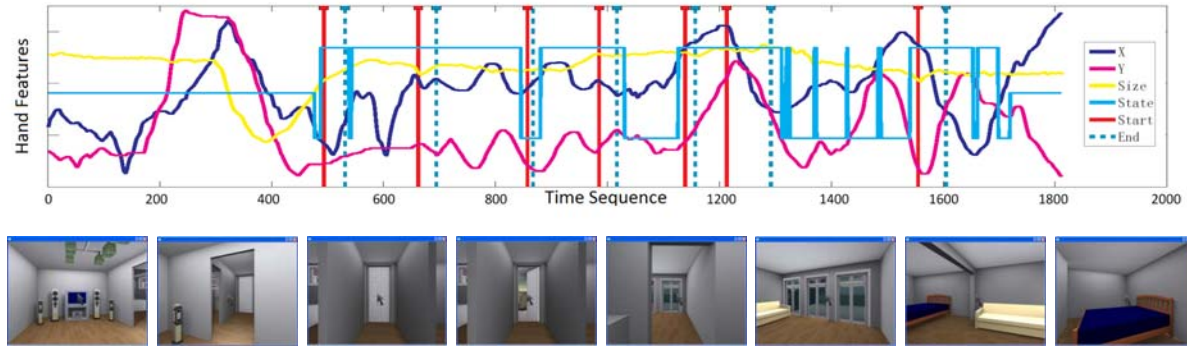


FIGURE 6. Results of gesture performed in virtual apartment

Totally 7 gestures including 1 static gesture ('stop') and 6 dynamic gestures ('turn right', 'turn left', 'push', 'grasp', 'turn on' and 'turn off') are defined for operation. Task route is shown in Figure 5 and corresponding gestures are shown in Table 4. If no valid gesture detected, default action is moving forward in current direction.

Results of current view in the virtual apartment after each gesture performed are shown in Figure 6. The frame length recorded for the whole task route is 1811. The first 492 frames mean no valid hand is detected, meaning in state s_1 . From frames 493 – 531 and 662 – 694, the hand performs turn right gesture, meaning state s_2 . There is a pause in frames 857 – 867, meaning state s_0 . Next four tasks are closely connected, with frames 985 – 1017 of opening the door, frames 1139 – 1137 of holding the switch, frames 1213 – 1292 of turning light on and frames 1555 – 1605 of turning light off.

In this practical gesture sequence, static and dynamic gestures are followed one by one closely with varied time for performing each gesture. From the observed sequence, recognition results affected by the order of appearing gestures in different categories fall into four cases, seen in Figure 4.

5. Conclusion and Future Work. In this paper, a recognition system using markless vision-based method from mixed static and dynamic gesture sequence is presented. This system proposes a state-based spotting algorithm for automatically identifying static and dynamic gestures. Static gestures are recognized using voting theory and relief algorithm, dynamic gestures are recognized using HMMs. Experimental results show that this system is able to recognize gestures with reasonable performance. A practical application of operating in virtual apartment is presented using the recognition methods proposed, and proves that this system is effective and efficient.

As experimental results have shown, the features used for classification and recognition are not accurate enough, and recognition rate of several gestures categories are not good enough. More precise features and specific categories must be defined in order to include a larger set of gestures for recognition. On the other hand, new devices as Microsoft Kinect® may be used as another ready-to-use recognition solution. What is more is that human experience in attending such interaction paradigm should be improved for a wider range of applications.

Acknowledgments. This work is supported by grants from National High Tech R&D Program of China (No. 2012AA011602), the National Basic Research Program of China (No. 2013CB329506), the National Natural Science Foundation of China (No. 61031002, 61001172, 61233015), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20100101120104), and the Fundamental Research Funds for the Central Universities.

REFERENCES

- [1] P. Garg, N. Aggarwal and S. Sofat, Vision based hand gesture recognition, *World Academy of Science, Engineering and Technology*, vol.49, no.1, pp.972-977, 2009.
- [2] J. Wachs, M. Kölsch, H. Stern and Y. Edan, Vision-based hand-gesture applications, *Communications of the ACM*, vol.54, no.2, pp.60-71, 2011.
- [3] S. Mitra and T. Acharya, Gesture recognition: A survey, *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.37, no.3, pp.311-324, 2007.
- [4] A. Wexelblat, An approach to natural gesture in virtual environments, *ACM Trans. on Computer-Human Interaction*, vol.2, no.3, pp.179-200, 1995.
- [5] S. Rautaray and A. Agrawal, Vision based hand gesture recognition for human computer interaction: A survey, *Artificial Intelligence Review*, pp.1-54, 2012.
- [6] J. Zhang, H. Lin and M. Zhao, A fast algorithm for hand gesture recognition using relief, *The 6th International Conference on Fuzzy Systems and Knowledge Discovery*, vol.1, pp.8-12, 2009.
- [7] E. Sangineto and M. Cupelli, Real-time viewpoint-invariant hand localization with cluttered backgrounds, *Image and Vision Computing*, 2011.
- [8] H. Birk, T. Moeslund and C. Madsen, Real-time recognition of hand alphabet gestures using principal component analysis, *Proc. of the Scandinavian Conference on Image Analysis*, vol.1, pp.261-268, 1997.
- [9] J. Triesch and C. von der Malsburg, Classification of hand postures against complex backgrounds using elastic graph matching, *Image and Vision Computing*, vol.20, no.13, pp.937-943, 2002.
- [10] Y. Pang, N. Ismail and P. Gilbert, A real time vision-based hand gesture interaction, *The 4th Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, pp.237-242, 2010.
- [11] M. de La Gorce, D. Fleet and N. Paragios, Model-based 3d hand pose estimation from monocular video, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.33, no.9, pp.1793-1805, 2011.
- [12] M. Van den Bergh and L. Van Gool, Combining rgb and tof cameras for real-time 3d hand gesture interaction, *IEEE Workshop on Applications of Computer Vision*, pp.66-72, 2011.
- [13] H. Lee and J. Kim, An hmm-based threshold model approach for gesture recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.21, no.10, pp.961-973, 1999.
- [14] K. Patwardhan and S. D. Roy, Hand gesture modelling and recognition involving changing shapes and trajectories, using a predictive eigentracker, *Pattern Recognition Letters*, vol.28, no.3, pp.329-334, 2007.
- [15] D. Ionescu, B. Ionescu, C. Gadea and S. Islam, A multimodal interaction method that combines gestures and physical game controllers, *Proc. of the 20th International Conference on Computer Communications and Networks*, pp.1-6, 2011.
- [16] D. Huang, W. Tang, Y. Ding, T. Wan, X. Wu and Y. Chen, Motion capture of hand movements using stereo vision for minimally invasive vascular interventions, *The 6th International Conference on Image and Graphics*, pp.737-742, 2011.
- [17] Q. Chen, N. Georganas and E. Petriu, Real-time vision-based hand gesture recognition using haar-like features, *Instrumentation and Measurement Technology Conference Proceedings*, pp.1-6, 2007.
- [18] A. Várkonyi-Kóczy and B. Tusor, Human-computer interaction for smart environment applications using fuzzy hand posture and gesture models, *IEEE Trans. on Instrumentation and Measurement*, vol.60, no.5, pp.1505-1514, 2011.
- [19] D. Ionescu, B. Ionescu, C. Gadea and S. Islam, An intelligent gesture interface for controlling tv sets and set-top boxes, *The 6th IEEE International Symposium on Applied Computational Intelligence and Informatics*, pp.159-164, 2011.
- [20] G. Hackenberg, R. McCall and W. Broll, Lightweight palm and finger tracking for real-time 3d gesture control, *Virtual Reality Conference*, pp.19-26, 2011.
- [21] K. Murphy, *Hidden Markov Model (HMM) Toolbox for Matlab*, <http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html>, 1998.