A Novel System for Automatic Hand Gesture Spotting and Recognition in Stereo Color Image Sequences

Mahmoud Elmezain, Ayoub Al-Hamadi, Bernd Michaelis
Institute for Electronics, Signal Processing and Communications (IESK)
Otto-von-Guericke-University Magdeburg
Germany
{Mahmoud.Elmezain, Ayoub.Al-Hamadi}@ovgu.de

ABSTRACT

Automatic gesture spotting and recognition is a challenging task for locating the start and end points that correspond to a gesture of interest in Human-Computer Interaction. This paper proposes a novel gesture spotting system that is suitable for real-time implementation. The system executes gesture segmentation and recognition simultaneously without any time delay based on Hidden Markov Models. In the segmentation module, the hand of the user is tracked using mean-shift algorithm, which is a non-parametric density estimator that optimizes the smooth similarity function to find the direction of hand gesture path. In order to spot key gesture accurately, a sophisticated method for designing a non-gesture model is proposed, which is constructed by collecting the states of all gesture models in the system. The non-gesture model is a weak model compared to all trained gesture models. Therefore, it provides a good confirmation for rejecting the non-gesture pattern. To reduce the states of the non-gesture model, similar probability distributions states are merged based on relative entropy measure. Experimental results show that the proposed system can automatically recognize isolated gestures with 97.78% and key gestures with 93.31% reliability for Arabic numbers from 0 to 9.

Keywords

Gesture spotting, Gesture recognition, Pattern recognition, Computer vision, Application.

1. INTRODUCTION

The hand gesture recognition is an active area of research in the vision community, mainly Human-Computer Interaction (HCI). A gesture is spatio-temporal pattern which may be static, dynamic or both. The goal of gesture interpretation is to push the advanced human-computer communication to bring the performance of HCI close to human-human interaction. In the last decade, several methods of potential applications [Dey06a, Elm08a, Kim07a, Mit07a, Yan07a] in the advanced gesture interfaces for HCI have been suggested but these differ from one another in their models. Some of these models are Neural Network (NN) [Dey06a], Hidden Markov Models (HMM) [Elm08a, Elm08b] and Dynamic Time Warping (DTW) [Tak92a]. One main concern of gesture recognition is how to segment some key gestures from a continuous sequence of motions. The gesture segmentation is also called gesture spotting. This is considered as a highly difficult process for two major problems, which arise in real-time gesture recognition system for continuous gesture to extract key gestures. The first problem is segmentation that means how to determine when a gesture starts and when it ends from hand motion trajectory. The second problem is caused by the fact that the same gesture varies in shape, trajectory and duration, even for the same person.

To overcome these problems, HMM is used in our system because it is capable of modeling spatio-temporal time series of gestures effectively and can handle non-gesture patterns. On the other hand, NN and DTW hardly represent the non-gesture patterns. Lee et al. [Lee99a] proposed an ergodic model based on adaptive threshold to spot the start and the end points of input patterns, and also classify the meaningful gestures by combining all states from all trained gesture models using HMM. Kang et al. [Kan04a] developed a method to spot and recognize the meaningful movements where this method concurrently separates unintentional movements from a given image sequences. Alon et al. [Alo05a] proposed a new gesture spotting and recognition algorithm using a pruning method that allows the system to evaluate a relatively small number of hypotheses compared to Continuous Dynamic Programming (CDP). Yang et al. [Yan07a] presented a method for recognition of whole-body key gestures in Human-Robot Interaction (HRI) by HMM and garbage model for non-gesture patterns. Mostly, previous approaches use the backward spotting technique that first detects the end point of gesture by comparing the probability of gesture models and non-gesture model. Secondly, they track back to discover the start point of the gesture through the optimal path using Viterbi algorithm [Law89a] and then the segmented gesture is sent to HMM for recognition. So, there is an inevitable time delay between the key gesture segmentation and recognition, where this time delay is not well for on-line gesture recognition.

To treat this problem, we propose a forward gesture spotting system that executes gesture segmentation and recognition simultaneously. The system recognize the isolated and key gestures for Arabic numbers (0-9) in real-time from stereo color image sequences by the motion trajectory of a single hand using HMM. To spot key gesture accurately, a sophisticated method of designing a non-gesture model is proposed, which is constructed by collecting the states of all gesture models in the system. The non-gesture model is a weak model for all trained gesture models where its likelihood is smaller than that the dedicated model for a given gesture.

The start and end points of gestures are based on the competitive differential observation probability value, which is determined by the difference of observation probability value of maximal gesture models and non-gesture model. The key gesture starts (ends) when the value of competitive differential observation probability changes from negative to positive (positive to negative). To reduce the states of the non-gesture model, model reduction which merges similar probability distributions states based on relative entropy is used [Cov91a]. Moreover, each isolated gesture number is based on 60 video sequences (42 for training and 18 for testing) and the continuous gestures are based on 280 video sequences for spotting key gestures and testing it. The achievement recognition rates on isolated and key gestures are 97.78% and 93.31% respectively. The organization of this paper is as follows; in section 2, hand tracking technique is introduced. Section 3 demonstrates the key gesture spotting system in three subsections. The experimental results are described in Section 4. Finally, Section 5 gives a few concluding remarks and refers to our future aims.

2. REAL-TIME HAND TRACKING

The hand is tracked in our system by mean-shift algorithm, which is a non-parametric (i.e. kernel) density estimator that optimizes a smooth similarity function to find the direction of the hand target's movement. We decide to use m-bin histograms as the representation of the object's color probabilities density function (pdf's), as they can satisfy the low-cost requirement of real-time tracking. YC_bC_r color space is used, where Y channel represents brightness and (C_b,C_r) channels refer to chrominance. The segmentation of skin colored regions becomes robust if only the chrominance is used in analysis. Therefore, we ignore Y channel to reduce the effect of brightness variation and use only the chrominance channels, which fully represent the color information. The segmentation of the hand with complex background takes place using 3D depth map and color information, which is more robust to the disadvantageous lighting and partial occlusion. This is done using Gaussian Mixture Models (GMM). For more details, the reader refers to [Elm08a, Nie07a].

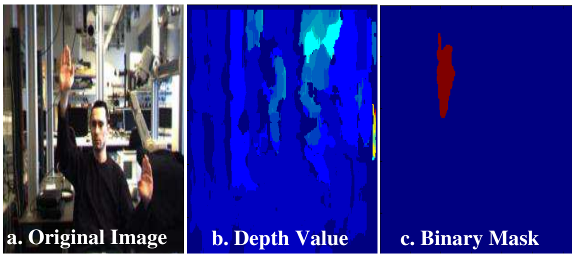


Figure 1. (a) First frame of video stream. (b) The Depth value of original image from a Bumblebee stereo camera. (c) Binary masked for left hand.

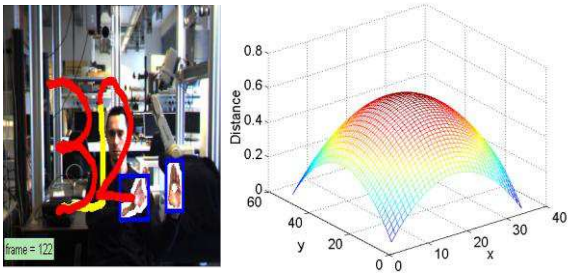
The segmentation module detects and localizes our object of interest (left hand) in the first video frame and we know exactly its position, as well as its shape and dimension (Fig. 1) [Elm08a]. Therefore, before starting with tracking, we used a binary mask to extract our hand target from the initial frame and find its color histogram with Epanechnikov kernel (monotonic decreasing kernel profile k(x)) [Com00a, Com03a, Sco92a] (Fig. 2(b)). Epanechnikov kernel assigns smaller weights to pixels father from the center. Using these weights increases the robustness of the density estimation since the peripheral pixels are the least reliable, being often affected by occlusions. Let $\{x_i^*\}$, $i=1\ldots n$ be the normalized pixel locations in the region defined as the hand target model. The probability of the feature $u=1\ldots m$ in the hand target model histogram is computed as;

$$q_u = F\sum_{i=1}^{n} k(\|x_i^*\|^2)\delta[b(x_i^*) - u] \quad (1)$$

where δ is the Kronecker delta function, equal to 1 only at u and 0 otherwise. The normalization constant

F is derived by imposing the condition $\sum_{u=1}^{m} q_u = 1$, where

$$F = \frac{1}{\sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)} \quad (2)$$



(a) (b)

Figure 2. (a) Gesture path for number 32 that is generated from connecting hand centroid points. (b) The Epanechnikov monotonically decreasing kernel for the hand target model of first image.

For the hand target candidate in the next frame, Let $\{x_i\}$, $i=1 \ldots n_h$ be the normalized pixel locations of the hand target candidate, centered at y in the current frame. Using the same kernel profile k(x), but with bandwidth h. The probability of the feature $u=1 \ldots m$ in hand target candidate histogram is computed as;

$$p_u(y) = F_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (3)$$

where

$$F_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)} \quad (4)$$

Moreover, the Bhattacharyya coefficient [Kha06a] is more suitable to evaluate the similarity between the hand target model and the chosen candidate rather than many more commonly technique, such as histogram intersection. The maximization of the Bhattacharyya coefficient between the unit vectors $\sqrt{\mathrm{q}}$ and $\sqrt{\mathrm{p(y)}}$ that representing the hand target model histogram and candidate model histogram respectively takes the following form;

$$\rho[p(y_0), q] = \sum_{u=1}^{m} \sqrt{p_u(y_0) q_u} \quad (5)$$

To find the best match of our hand target in the sequential frames, the Bhattacharyya coefficient is maximized, which means that we need to maximize the term;

$$\sum_{i=1}^{n} w_i \, k(\left\|\frac{y - x_i}{h}\right\|^2) \quad (6)$$

where h is the kernel's smoothing parameter or bandwidth and the weights w_i is given by;

$$w_i = \sum_{i=1}^{n} \sqrt{\frac{q_u}{p_u(y_0)}} \, \delta[b(x_i) - u] \quad (7)$$

Mean shift iteration uses the gradient of this similarity function as an indicator of the direction of hand's movement where the centroid point of hand candidate is shifted by Eq. 8;

$$y = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \quad (8)$$

Thereby, the hand motion trajectory so-called gesture path is generated from connecting the centroid points of hand regions (Fig. 2(a)).

3. KEY GESTURE SPOTTING

The task of locating key patterns from a stream of input signal is to find the start and end points of a meaningful gesture while ignoring the rest. Here, we discuss how to model gesture patterns discriminately and how to model non-gesture patterns effectively. Each reference pattern for Arabic numbers from 0 to 9 is modeled by gesture HMM and all other patterns are modeled by a single HMM called a non-gesture model (garbage model) [Yan07a, Lee99a], however, it is not easy to obtain the set of non-gesture patterns because there are infinite varieties of meaningless motion. Fig. 3 represents a simplified gesture spotting structure where the hand gesture path is projected into 3D-plane.

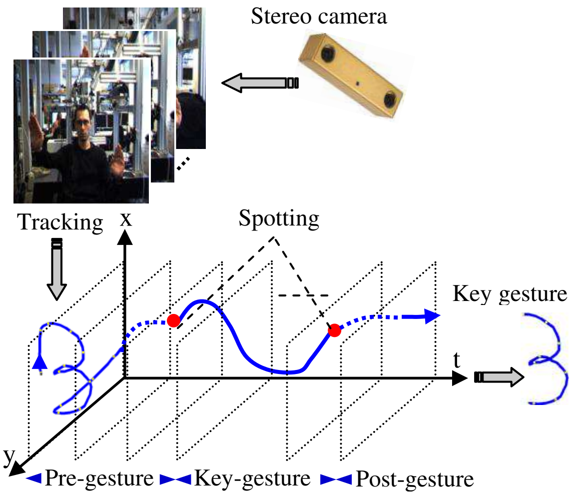


Figure 3. Gesture spotting structure, the dotted curve (Pre-and Post-) refer to non-gesture pattern and dark curve (Key-) represents gesture pattern.

3.1 Gesture Model

For each reference gesture model, each HMM state represents the local segmental part of it, while the states transition represent the sequential order structure in a gesture trajectory. The number of HMM states is an important parameter for each reference pattern because the excessive number of states can generate the over-fitting problem if the number of training samples is insufficient compared to the model parameters. When there are insufficient number of states, the discrimination power of the HMM is reduced, since more than one segmented part should be modeled on one state. Moreover, the number of states in our gesture spotting system is based on the complexity of each gesture number and is determined by mapping each straight-line segment into a single HMM state (Fig. 4). In practice, we considered the Left-Right Banded topology (LRB) [Law89a] for the following reasons. Since each state in Ergodic topology has many transitions than Left-Right (LR) and LRB topologies, the structure data can be lost easily. On the other hand, LRB topology has no backward transition where the state index either increases or stays the same as time increases. In addition, LRB topology is more restricted rather than LR topology and simple for training data that will be able to match the data to the model. Orientation dynamic features are obtained from spatio-temporal trajectories and then quantized to generate its codewords (1-18). The quantized vectors are trained by Baum-Welch (BW) re-estimation algorithm [Law89a] for the initialized HMM parameters $\lambda = (\Pi, A, B)$. For more details, the reader can refer to [Elm07a, Elm08a, Elm08b].

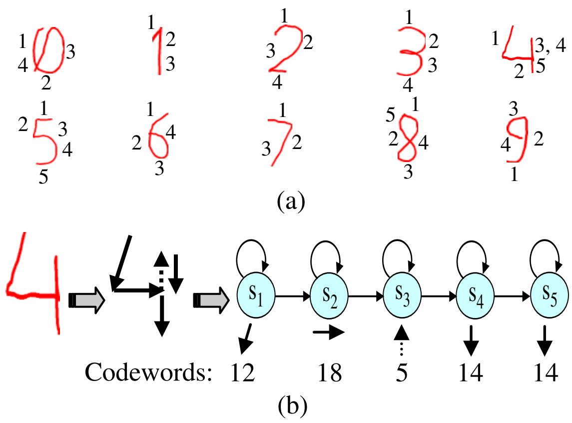


Figure 4. The hand gesture paths and straight-line segmentation. (a) The Gesture paths from hand motion trajectory for Arabic numbers (0-9) with its segmented parts. (b) The LRB topology with segmented line codewords for a gesture path 4.

3.2 Non-gesture Model

A non-gesture model represents any motion trajectory or any part of it other than gesture model. For correcting gesture spotting, the likelihood of a gesture model for a given pattern that is mentioned previously should be distinct enough. Unfortunately, the HMM recognizer selects a model with the best likelihood; we cannot ensure that the pattern is really similar to the reference gesture model unless the likelihood value is high enough. Thus, the non-gesture model is proposed where it provides a good confirmation for rejecting the non-gesture pattern. The property of HMM internal segmentation denotes that each state with its self-transition represents a segmental pattern of a target gesture and the outgoing transitions represent a sequential progression of the segments in a gesture. With this property, we construct ergodic model with the states copied from all gesture models in our system, in addition two dummy states (Start state ST and End state ET), and then fully connect the states (Fig. 5). The dummy states are called null states, which observe no symbol and are passed without time delay [Yan07a, Pre98a]. We construct our non-gesture model as follows:

1. Duplicate all states from all gesture models, each with output observation probabilities. Then, we re-estimate that probabilities with Gaussian distribution smoothing filter to make the states represent any pattern.
2. Self-transition probabilities are kept as in the gesture models.
3. All outgoing transitions are equally assigned as;

$$\hat{a_{ij}} = \frac{1 - a_{ij}}{N - 1} \quad ; for\ all\ j, i \neq j \tag{9}$$

where $\hat{a_{ij}}$ represents the transition probabilities of non-gesture model from state s_i to state s_j, a_{ij} is the transition probabilities of gesture models from state s_i to state s_j and N is the number of states in all gesture models. The non-gesture model is a weak model for all trained gesture models and represents every possible pattern where its likelihood is smaller than the dedicated reference model for a given gesture because of the reduced forward transition probabilities. Also, the likelihood of the non-gesture model provides a confidence limit for the calculated likelihood by other gesture models. Thereby, we can use confidence measures as an adaptive threshold for selecting the proper gesture model or gesture spotting. The number of states for non-gesture model increases as the number of gesture model increases. Furthermore, an increase in the number of states is nothing but dues to a waste time and space. To treat this problem, relative entropy [Cov91a] is used to reduce the non-gesture model states because there are many states with similar probability distribution.

3.3 Key Gesture Spotting & Recognition

In continuous hand motion, key gestures appear intermittently with transition connecting motion. To spot these key gestures in our system, we construct gesture spotting network as shown in Fig. 6. The gesture spotting network can be easily expanded the vocabularies by adding a new key gesture HMM model and then rebuilding a non-gesture model. This network contains ten gesture models for Arabic numbers from 0 to 9. These ten models are designed using LRB model with number of states ranging from 3 to 5 based on its complexity.

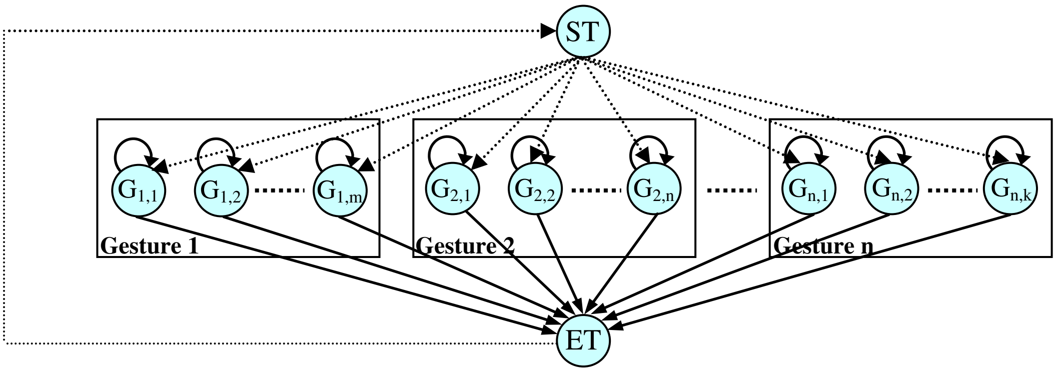


Figure 5. The general non-gesture model, where the dotted arrows represent null transitions, $G_{i,j}$ refers to the state j in gesture number i, ST and ET are the two dummy states for starting and ending, receptively.

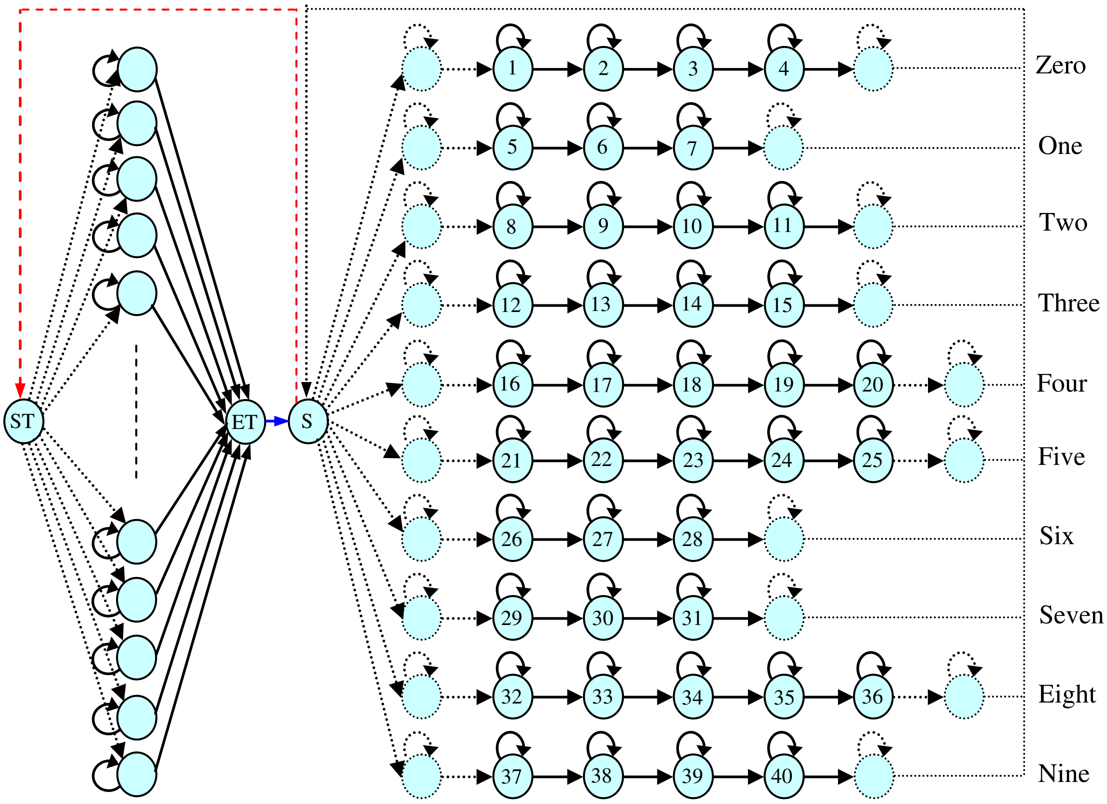


Figure 6. The gesture spotting network, where it contains ten Arabic number gesture models from 0 to 9 that are designed by using LRB model with varying states from 3 to 5 and also contains the non-gesture model after states reduction by relative entropy.

Also, it contains non-gesture model after states reduction by relative entropy function and the dummy start state S. The gesture spotting network finds the start and end points of key gestures that is embedded in the input stream and performs the segmentation and recognition tasks simultaneously. For forward spotting, we have defined a competitive differential observation probability value, which is determined by the difference observation probability value of maximal gesture models and non-gesture model (Fig. 7). The maximal gesture model is the gesture whose observation probability is the largest among all ten gesture $p(O|\lambda g)$. The transition from non-gesture to gesture occurs when the competitive differential observation probability value changes from negative to positive (Eq.10, then O can possibly as gesture g). Similarly, the transition from gesture to non-gesture occurs around the time that this value changes from positive to negative (Eq.11, then O cannot be a gesture). These observations can be used as a rule for detecting start and end point of gestures.

$$\exists g : p(o|\lambda_g) > p(o|\lambda_{non-gesture}) \quad (10)$$

$$\forall g : p(o|\lambda_g) < p(o|\lambda_{non-gesture}) \quad (11)$$

The proposed gesture spotting system contains two main modules (segmentation module and recognition module). In the gesture segmentation module, we use a sliding window technique, which calculates the observation probability of all gesture models and non-gesture model for observed segmented parts to spot the start point by competitive differential observation probability value. The optimal size of sliding window is determined empirically (equal 5 in our system) where the system is the best in term of results (Fig. 8(b)).

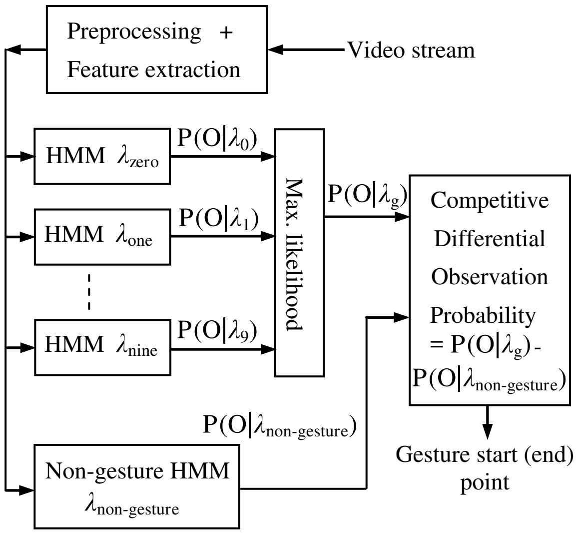


Figure 7. A block diagram shows how to calculate a competitive differential observation probability value between maximal gesture models for Arabic numbers from 0 to 9 and non-gesture model.

After spotting a start point in a continuous image sequences, then it activates gesture recognition module, which performs the recognition task for the segmented part accumulatively until it receives the end signal of a gesture. At this point, the gesture recognition module decides the type of observed gesture segmentation (argmax P(O|λ_g)) by Viterbi algorithm [Law89a]. This procedure is repeated until no input images exist. The following steps show how Viterbi algorithm works on gesture model λ_g;

1. Initialization:

$$\delta_1^g(i) = \Pi_i b_i^g(o_1); \quad for\ 1 \le i \le N \tag{12}$$

2. Recursion (accumulative observation probability computation):

$$for\ 2 \le t \le T, 1 \le j \le N$$
$$\delta_t^g(j) = \max_i\left[\delta_{t-1}^g(i) a_{ij}^g\right]. b_j^g(o_t); \tag{13}$$

3. Termination:

$$p(o|\lambda_g) = \max_i\left[\delta_T^g(i)\right] \tag{14}$$

where a_{ij}^g is the transition probability from state i to state j, $b_j^g(o_t)$ refers to the probability of emitting o at time t in state j, and $\delta_t^g(j)$ represents the maximum likelihood value in state j at time t.

4. EXPERIMENTAL RESULTS

Our proposed system showed good results to recognize Arabic numbers in real-time from stereo color image sequences via the motion trajectory of a single hand using HMM. The system was implemented in Matlab language and the input images were captured by Bumblebee stereo camera system that has 6 mm focal length for about 2 to 5 second at 15 frames per second with 240×320 pixels image resolution. Our experiment is carried out an isolated gesture recognition test and key gesture spotting test.

4.1 Isolated Gesture Recognition

In our experimental results, each isolated gesture number from 0 to 9 was based on 60 video sequences, which 42 video samples for training by BW algorithm and 18 video samples for testing (Totally, our database contains 420 video sample for training and 180 video sample for testing). The gesture recognition module match the segmented gesture against database of reference gestures, to classify which class it belongs to. The higher priority was computed by Viterbi algorithm to recognize the numbers in real-time frame by frame over LRB topology with different number of states ranging from 3 to 5 based on its complexity. From table 1, the recognition ratio of isolated gestures achieved best results with 97.78%. The recognition ratio is the number of correctly recognized gestures over the number of input gestures (Eq.15). Fig. 8(a) shows the output of our system for isolated gesture number 8.

$$Recognition\ ratio = \frac{\#\ of\ correctly\ recognized\ gestures}{\#\ of\ test\ gestures} \times 100\% \tag{15}$$

4.2 Key Gesture Spotting Test

Our database includes on 280 video samples for continuous hand motion. Each video sample either contains one key gesture or more than one key gesture. Fig. 8(b) shows that the gesture spotting performance based on the size of sliding windows. So, we measure the gesture spotting accuracy according to different window size from 1 to 8. We noted that, the gesture spotting accuracy is improved initially as the sliding window size increase, but degrades as sliding window size increase further. Therefore, the optimal size of sliding window is 5 empirically, where the reliability of automatic gesture spotting system is 93.31% (Table 1). In automatic gesture spotting task, there are three types of errors, namely, insertion, substitution and deletion. The insertion error occurs when the spotter detects a nonexistent gesture. A substitution error occurs when

Gesture path	Train Data	Isolated gestures results			Key gestures spotting Results					
		Test	Correct	Rec.(%)	Test	Insert	Delete	Substitute	Correct	Rel.(%)
0	42	18	17	94.44	28	2	1	2	25	83.33
1	42	18	18	100.00	28	0	1	1	26	92.86
2	42	18	17	94.44	28	0	0	2	26	92.86
3	42	18	18	100.00	28	0	0	0	28	100.00
4	42	18	18	100.00	28	0	0	1	27	96.43
5	42	18	18	100.00	28	0	1	1	26	92.86
6	42	18	17	94.44	28	1	1	1	26	89.66
7	42	18	18	100.00	28	0	0	0	28	100.00
8	42	18	17	94.44	28	1	0	2	26	89.66
9	42	18	18	100.00	28	0	1	0	27	96.43
Total	**420**	**180**	**176**	**97.78**	**280**	**4**	**5**	**10**	**265**	**93.31**

Table 1. Isolated gesture recognition and key spotting gesture results for Arabic numbers from 0 to 9.

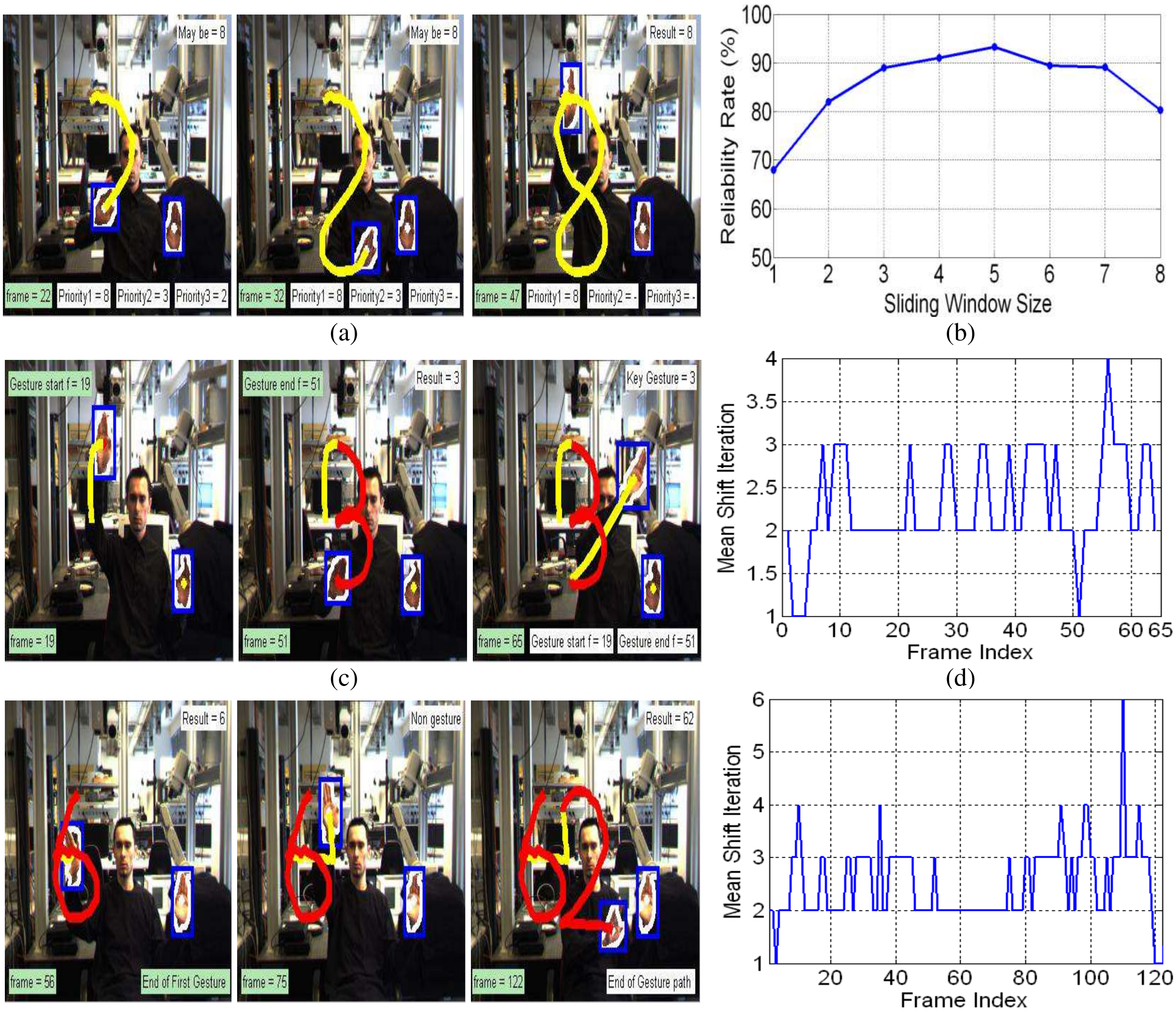


Figure 8. The system outputs. (a) Isolated gesture 8. (b) Spotting accuracy for different sliding window size from 1 to 8. (c) One key gesture spotting 3, where the start point at frame=19 and the end point at frame=51. (d) The number of mean shift iterations function (to connect hand centroid points) of the frame index for gesture path 3 where the mean number of iteration is a 2.29 per frame. (e) Gesture spotting 62. (f) The number of mean shift iterations function is 2.38 per frame for gesture path 62.

the key gesture is classified falsely. The deletion error occurs when the spotter fails to detect a key gesture. The reliability of gesture spotting system in terms of these errors is measured by Eq. 16.

$$Reliability = \frac{\#\, of\ correctly\ recognized\ gestures}{\#\, of\ test\ gestures + \#\, of\ insertion\ errors} \quad (16)$$

Here, we note that some insertion errors cause the substitution errors or deletion errors where the insertion errors affect on the gesture spotting ratio directly. Fig. 8(c) and Fig. 8(e) show the results of key gesture spotting 3 and 62 respectively.

5. SUMMARY AND CONCLUSION

This paper proposes an automatic system that recognizes isolated gesture, in addition to key gesture spotting from continuous hand motion for Arabic numbers from 0 to 9 based on HMM. The proposed system describes the gesture spotting network, which finds the start and end points of key gestures that is embedded in the input stream by the difference observation probability value of maximal gesture models and non-gesture model. Our system uses forward spotting technique that performs the segmentation and recognition tasks simultaneously. This technique is suitable for real-time applications and solves the issues of time delay between segmentation and recognition tasks. Our results show that; an average recognition rate is 97.78% and 93.31% for isolated and key gestures spotting, respectively. In future, our research focuses on the motion trajectory will carried out by a fingertip using multi-camera system over combined features.

6. ACKNOWLEDGMENTS

This work was supported by Bernstein-Group (BMBF: 01GQ0702), LSA grants (C4-NIMITEK 2, FKZ: XN3621E/1005M) and DFG grants (SFB/TR 62, TP C1).

REFERENCES

[Alo05a] Alon, J., Athitsos, V., Scharoff, S.: Accurate and Efficient Gesture Spotting via Pruning and Sungesture Reasoning. 3766 LNCS, pp. 189--198, 2005.

[Com00a] Comaniciu, D., Ramesh, S., Meer, P.: Real-Time Tracking of Non-Rigid Objects Using Mean Shift. Conf. on Computer Vision and Pattern Recognition (CVPR), Vol. 2, pp. 1--8, 2000.

[Com03a] Comaniciu, D., Ramesh, S., Meer, P.: kernel-Based Object Tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 25(5), pp. 564--577, 2003.

[Cov91a] Cover, T.M., Thomas, J.A.: Entropy, Relative Entropy and Mutual Information. Elements of Information Theory, pp. 12--49, 1991.

[Dey06a] Deyou, X.: A Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG. ICPR 06, pp. 519--522, 2006.

[Elm07a] Elmezain, M., Al-Hamadi, A., Michaelis, B.: Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 1209--1214, 2007.

[Elm08a] Elmezain, M., Al-Hamadi, A., Michaelis, B.: Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences. The Journal of WSCG'08, Vol. 16(1), pp. 65--72, 2008.

[Elm08b] Elmezain, M., Al-Hamadi, A., Appenrodt, J., Michaelis, B.: A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition. The International Conference on Computer Vision, Image and Signal Processing (CVISP2008), PWASET, Vol. 31, ISSN 1307-6884, pp. 394--401, 2008.

[Kan04a] Kang, H., Lee, C., Jung, K.: Recognition-based Gesture Spotting in Video Games. Pattern Recognition Letters, Vol. 25(15), pp. 1701--1714, 2004.

[Kha06a] Khalid, S., Ilyas, U., Sarfaraz, S., Ajaz, A.: Bhattacharyya Coefficient in Correlation of Gary-Scale Objects. Journal of Multimedia, Vol. 1(1), pp. 56--61, 2006.

[Kim07a] Kim, D., Song, J., Kim, D.: Simultaneous Gesture Segmentation and Recognition Based on Forward Spotting Accumlative HMMs. The Journal of the Pattern Recognition Society, Vol. 40(2007), pp. 3012--3026, 2007.

[Law89a] Lawrence, R.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceeding of the IEEE, Vol. 77(2), pp. 257--286, 1989.

[Lee99a] Lee, H., Kim,J.: An HMM-Based Threshold Model Approach for Gesture Recognition. IEEE Trans. PAMI, Vol. 21(10), pp. 961--973, 1999.

[Mit07a] Mitra, S., Acharya, T.: Gesture Recognition: A Survey. IEEE Transactions on Systems, MAN, and Cybernetics, pp. 311--324, 2007.

[Nie07b] Niese, R., Al-Hamadi, A., Michaelis, B.: A Novel Method for 3D Face Detection and Normalization, Journal of Multimedia, ISSN,1796-2048, 2007.

[Pre98a] Du Preez, J.A.: Efficient Training of High-Order Hidden Markov Models Using First-Order Representations. Comput. Speech Lang., Vol. 12(1), pp. 23--39, 1998

[Sco92a] Scott, D., W.: Multivariate Density Estimation. Wiley, 1992.

[Tak92a] Takahashi, K., Sexi, S., Oka, R.: Spotting Recognition of Human Gestures From Motion Images. Technical Report IE92-134, pp. 9--16, 1992.

[Yan07a] Yang, H., Park, A., Lee, S.: Gesture Spotting and Recognition for Human-Robot Interaction. IEEE Transaction on Robotics, Vol. 23(2), pp. 256--270, 2007.