

Research Article

Hand Gesture Recognition Using Particle Swarm Movement

Clementine Nyirarugira, Hyo-rim Choi, and TaeYong Kim

Graduate School of Advanced Imaging Sciences, Film, and Multimedia, Chung-Ang University, Seoul 156-756, Republic of Korea

Correspondence should be addressed to TaeYong Kim; kimty@cau.ac.kr

Received 4 March 2016; Revised 11 May 2016; Accepted 24 May 2016

Academic Editor: Erik Cuevas

Copyright © 2016 Clementine Nyirarugira et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a gesture recognition method derived from particle swarm movement for free-air hand gesture recognition. Online gesture recognition remains a difficult problem due to uncertainty in vision-based gesture boundary detection methods. We suggest an automated process of segmenting meaningful gesture trajectories based on particle swarm movement. A subgesture detection and reasoning method is incorporated in the proposed recognizer to avoid premature gesture spotting. Evaluation of the proposed method shows promising recognition results: 97.6% on preisolated gestures, 94.9% on stream gestures with assistive boundary indicators, and 94.2% for blind gesture spotting on digit gesture vocabulary. The proposed recognizer requires fewer computation resources; thus it is a good candidate for real-time applications.

1. Introduction

Hand gestures are a powerful human to human communication channel that forms a major part of information transfer in our daily life. Incorporating hand gestures into the human computer interface is becoming an important research area. The usage of vision-based hand gesture is often preferred for its noncumbersome interaction [1, 2]. To interact with a device using hands, computers should be able to visually detect the hand and recognize its gestures from video input [3–5].

The latest computer vision technologies make real-time hand detection and gesture recognition promising [6]. Many different approaches that use hand as an interface device have been proposed [1, 2]. To support gestures interaction, a recognizer must be integrated into the system and trained to the specific gestures the system will support. However, most of recognizers have inherent limitations in the types of gestures they can efficiently discriminate [7], which often results from high intergesture categories correlation.

In addition, online hand gesture recognition brings more challenges, as gestures are often characterized by unpredictable boundary noise (see Figure 1(a)), due to the lack of perfect vision-based gesture segmentation methods, between gestures (when gestures are coarticulated) (see Figure 1(b)), leading to ambiguous recognition [8].

In online hand gesture recognition systems, often two assumptions are made: the existence of gesture boundary indicators or blind gesture spotting, the latter being more difficult. Each approach has different implications and the preference of one over the other is application-dependent. Few attempts have been made in blind spotting [9, 10] that are predominant in real-world applications. Blind gesture recognition makes the classification ambiguous and more difficult, as less information is presented to the recognizer.

One of the challenges is that gestures vocabularies often contain highly correlated gestures that lead to ambiguous recognition [8, 11]. For instance, in the digit gesture vocabulary the hand motion performed to gesticulate the digit two is a part of the one performed for the digit three. To avoid ambiguity in recognition, additional actions are often required. Alon et al. [10] proposed a template matching method with subgesture detection and reasoning method, which avoid premature gesture spotting. In [9], the gestures boundaries are detected prior to gesture recognition using Hidden Markov Models (HMM). These methods produce promising recognition rates; however, the computation resources are still high.

In this paper, we adapt particle swarm optimization (PSO) to the problem of gesture recognition. PSO is a pattern search method [12]. In general, a PSO algorithm is initialized with a group of N particles. Each particle is characterized

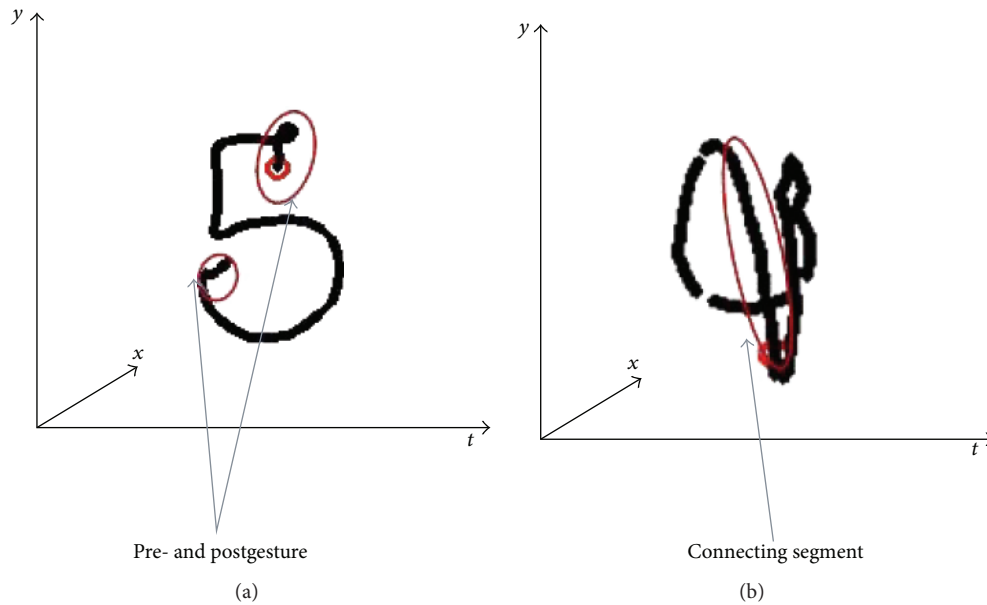


FIGURE 1: Distorted gestures: (a) boundary noise that may lead to classify 5 as 8 and (b) connecting segment that lead to classify 4 as 9.

by its personal best position, which is updated according to its fitness value (likelihood). Within the gesture recognition context, the search/solution space is composed of gesture templates to which we assign particles and allow them to evolve through a deterministic matching process, guided by observed data. The recognized gesture category is the one whose particle has the highest matching score and has reached or is closest to the end of the template. Gesture detection under PSO matching offers a more efficient way of gesture segmentation as gesture boundaries can be inferred directly from the matching process, that is, the particle's personal best positions; thus, there is no need for backtracking. The contribution of this paper can be summarized as follows:

- (1) An automated process of segmenting meaningful gesture trajectories based on particle swarm movement is proposed.
- (2) To avoid premature gesture spotting, a subgesture detection and reasoning method is incorporated within the proposed recognizer.
- (3) Reduce considerably the processing time of gesture spotting, as gesture boundaries can be inferred directly from particle's personal best position.

We evaluate the performance of the proposed method under three gesture spotting and recognition assumptions, that is, manually spotted (with consistent boundaries), pre-segmented with boundary indicators (with possible boundary noise), and blind gesture spotting [8]. The remainder of this paper is organized as follows; Section 2 discusses recent related works in dynamic hand gesture recognition. Section 3 gives details on the proposed method, followed by Section 4 that discusses a series of experiments to evaluate the proposed methods. Finally, Section 5 concludes this work.

2. Hand Gesture Recognition

Hand gesture recognition is a difficult and challenging problem that has been addressed in many ways. A widely used approach is the Hidden Markov Model (HMM) [9, 13–16]. HMM based gesture recognition methods represent each gesture by a set of states associated with probabilities (initial, transition, and observation) learned from the training examples. HMM recognizers choose a model with the best likelihood and classify a given gesture to the corresponding gesture category. Although HMM recognition systems choose a model with the best likelihood, it is not guaranteed that the pattern is really similar to the reference gesture unless the likelihood value is high enough, above some threshold. In the case a simple threshold does not work well, a sophisticated threshold model can be derived as done in [14] or other verification mechanisms are applied as done in [4].

To produce good results, HMM need to be well trained to get good representative models [9]. Rule based trajectory segmentation for modeling the hand motion trajectory has been proposed [13], to provide a robust initialization. The authors did an extensive study on good initialization of the HMM, which is often of primary concern in HMM based recognition method. They derived an automated process for determining the number of states and a robust initialization of HMM. The proposed method can separate each angular state of the training data at the initialization step, thus providing a solution to mitigate the ambiguities on initializing the HMM and increase the recognition of the HMM.

An automatic system that handles hand gesture spotting and recognition simultaneously based on a generative model as HMM has been proposed in [9]. To spot meaningful (key) gestures of numbers (0–9) accurately, a stochastic method for designing a nongesture model with HMM was proposed

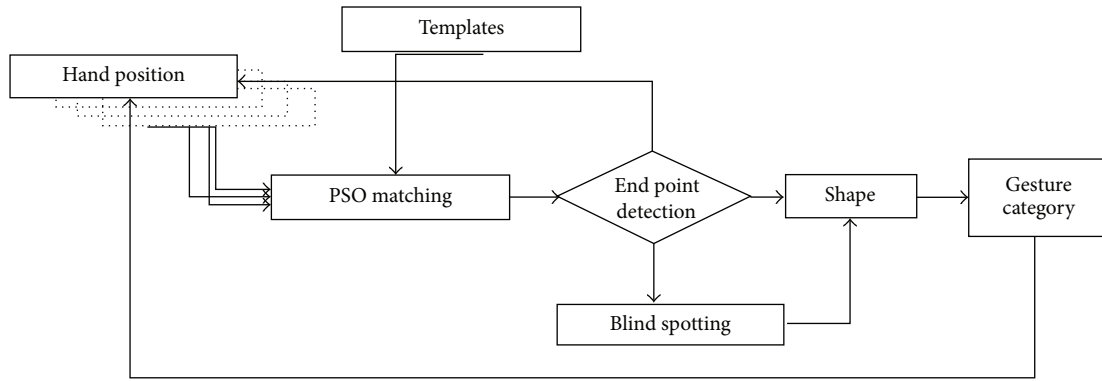


FIGURE 2: Flowchart of the proposed recognizer.

without training data. The nongesture model provides a confidence measure that is used as an adaptive threshold to find the start and the end point of meaningful gestures, which are embedded in the input video stream. Threshold model approach is often computationally expensive, as one has to create a big size nongesture model [4]. To filter out garbage gesture, in [4], authors appeal to a simple Gaussian model threshold based on a single Gaussian probability density, learned during the training process. The main disadvantages of HMM based recognition methods are that they require a large number of samples and long training time to calibrate the models [14]. When there are not enough training examples, template matching methods such as Dynamic Time Warping (DTW) are preferred.

DTW is another approach often used for dynamic gesture recognition task [3, 17, 18]. The DTW based recognition methods attempt to line up a given sequence to gestures templates. To produce good results, many templates may be required to take into account variation of a given gesture category as it may be the case for other template matching methods. Moreover, DTW is shown, in time series, to be sensitive to noise and outliers. Different distance metrics have been derived to improve DTW results. Probability-based DTW and bag-of-visual-and-depth-words for human gesture recognition in RGB-D (Red Green Blue-Depth) used a soft distance based on the probabilistic similarity measure [17]. The aforementioned derived distance measure improves the recognition rate of classical DTW that uses Euclidean distance as a cost function. In [19], a comparative evaluation of six trajectory distance measures was performed. The longest common subsequence (LCS) measure outperforms the others on datasets with varying characteristics. The LCS is a string matching algorithm that focuses on the matched subsequence. This makes it robust to noise. The LCS has recently gained more attention and has been successfully used in dynamic hand gesture recognition systems.

Stratified hand gesture using normalized longest common subsequence with rough sets [8] aimed to increase LCS discrimination capabilities, as LCS is a global alignment method. The authors achieved good performance through the normalization of LCS and pairing the LCS with rough sets theory. Gestures are represented using rough set

approximations, through which discriminative information was generated and used to resolve ambiguous recognition.

The success of a gesture recognition system does not only rely on recognition but also should run in real time. Vision-based gesture segmentation is difficult and gesture boundary detection is often computationally expensive. Most approaches rely on the usage of sliding windows. However, without constraining users to perform gestures at nearly the same speed, window management is challenging which may result from high variation in gesture size and speed. The failure to predict the window size leads to a recomputation of the likelihood at every window size update, with delay. It also may lead to premature gesture recognition. To avoid uncertainty in the choice of the sliding windows, different approaches are followed such as the usage of gesture boundary indicators or classification of motion primitive. The aforementioned boundary indicators increase gesture detection rates; however, users have to bear them in mind and apply them for a successful interaction. This might be tiring based on application, such as string recognition system [8].

The following section introduces gestures detection using particle swarm movement, which does not require explicit gesture boundary detection, nor the usage of sliding windows, as gesture boundary detection is inferred from PSO matching process.

3. PSO Based Gesture Recognition

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution. In general, a PSO algorithm is initialized with a group of N particles. Each particle is characterized by its personal best position, which is updated according to its fitness value (likelihood). The flowchart of the proposed recognizer is depicted in Figure 2.

3.1. Gesture Representation. Given a sequence of images (video file) in which a hand gesture is being performed, the gesturing hand is first detected and its trajectory p_s is acquired; $p^s = [p_1, p_2, \dots, p_t]$, with length t^s .

Each point is represented in 2D, that is, $p^s(t) = [x(t), y(t)]$.

- (1) We first computed the Euclidean distance from the start and end of the gesture trajectory to infer the shape of a gesture, in the sense of being either closed or open, as follows:

$$\text{Euclid}_p^s = d(p_n^s - p_1^s). \quad (1)$$

This will become handy in recognizing gestures with the same motion, but different shapes, such as digit 0 and digit 6.

- (2) Gesture trajectory is then mapped into the motion orientation segments:

$$\theta^s = [\theta_1, \theta_2, \dots, \theta_{t^s-1}], \quad (2)$$

where $\theta^s(t) = \text{atan2}(\Delta y(t), \Delta x(t))$. Orientation segments are created as detailed in [8].

A gesture is then represented by $[\theta^s, \text{Euclid}_p^s]$.

3.2. Gesture Recognition Using Particle Swarm Movement. PSO was originally developed for continuous problems. Adapting it to discrete domains is somehow tricky, approximations need to be made, and some concepts may lose some of their meaning. In our gesture recognition setting, the search space is composed of gesture templates, to which particles are assigned. The solution space is defined as follows:

$$\Psi = \{\theta^i = [\theta_1^i, \theta_2^i, \dots, \theta_{n_i}^i]\}_{i=1}^L, \quad (3)$$

where $L \leftarrow 1, \dots, |\Psi|$ is number of particles, θ^i is the i th template, and n_i is the length of the i th template.

Gestures recognition using particles swarm is performed as follows.

- (1) Initialize particles in the search (templates) space,

$$\{x_j^i = 0; p_{\text{best}}^i = x_j^i\}_{i=1}^L, \quad (4)$$

where x_j^i represent the j th position of the i th particle, in the i th template, and p_{best}^i is the personal best position of the i th particle, to simplify notation $x_j^i = j$ when used as an index.

- (2) Evaluate the particles' fitness function.

For each motion orientation segment θ_t , a matching score is calculated and the particle's fitness value is computed as follows:

$$\begin{aligned} \gamma &= f(\theta_j^i, \theta^t); \quad \gamma \in [-1, 0, 1]; \\ F_t(\theta^i) &= F_{t-1}(\theta^i) + |\theta_t^i| * f(\theta_j^i, \theta^t), \end{aligned} \quad (5)$$

where $\gamma = -1$ when $\theta_j^i \neq \theta^t$ and $p_{\text{best}}^i \neq 0$, $\gamma = 0$ when $\theta_j^i \neq \theta^t$, and $\gamma = 1$ when $\theta_j^i = \theta^t$.

- (3) The personal best position is updated only when the current fitness value is greater than the previous one:

$$\begin{aligned} p_{\text{best}}^i &= p_{\text{best}}^i + 1, \\ \text{if } F_t(\theta^i) &> F_{t-1}(\theta^i). \end{aligned} \quad (6)$$

In our gesture recognition setting each particle moves locally towards a more likely position in the search space:

$$x_j^i = x_j^i + 1. \quad (7)$$

- (4) Compute a subset of possible solution:

$$\begin{aligned} \psi_{\text{sol}} &= \{\theta^i\} \\ \text{such that } p_{\text{best}}^i &\geq \beta^i, \end{aligned} \quad (8)$$

where β^i is a threshold for the i th gesture category.

- (5) If $\Psi_{\text{sol}} \neq \Phi$, the recognized gesture category i_0 is determined as follows:

$$i_0 = \arg \max_{\theta^i \in \Psi_{\text{sol}}} \left(\frac{F_t(\theta^i)}{|\theta^i| + |\theta|} \right). \quad (9)$$

3.3. Blind Gestures Spotting through PSO Matching. Given a set of templates Ψ representing gesture categories within a gesture vocabulary, a list of subgestures-super gestures relationship is generated by cross matching all templates and populating their personal best position in the matrix G , $G \leftarrow T \times T$:

$$G_{n,n} = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n,1} & g_{n,2} & \cdots & g_{n,n} \end{pmatrix}, \quad (10)$$

where $G_{i,j}$ represents the personal best position of the particle j when matched with particle i , that is, the number of their common segments. A super gesture category is the one that has a part that overlaps with another whole gesture within the same gesture vocabulary. The list of subgesture or super gesture will be checked during gesture blind spotting to avoid premature gesture spotting. The sub_super gesture relationship is detected as follows:

$G \leftarrow T \times T$: created using PSO matching

for $i = 1: n$

for $j = 1: n$

if $(G_{i,j} = |T^i|)$

$Y_{(\text{sub}, \text{sup})} \leftarrow \langle i, j \rangle$

end if

end for

end for

TABLE 1: Confusion matrix of recognition of gesture with known boundaries.

	0	1	2	3	4	5	6	7	8	9
0	98.1%						1.9%			
1		100%								
2			93.3%					5.3%		
3				100%						
4					100%					
5						100%				
6	6.1%						94.9%			
7								100%		
8									92.3%	
9										97.1%

Gesture spotting is the process of detecting a meaningful gesture in a continuous gesture stream. The following are gestures spotting rules used with particle swarm movement matching. Given a list of sub-super gesture relationship, $Y_{(sub,sup)}$, a gesture is detected as follows:

- (1) To spot a gesture, we first check if there is a particle that has already reached the end of its corresponding template:

$$i_0 \leftarrow p_{best}^i == |T^i|, \quad \forall i \in L. \quad (11)$$

- (2) Detection of nonsubgesture: if the detected gesture i_0 is not a subgesture, $i_0 \notin Y_{(sub)}$, that is, it could be a super gesture or normal gesture, then it is reported.
- (3) Spotting subgestures: if the detected gesture is subgesture, that is, $\langle i_{sub}^0, i_{sup}^0 \rangle$ relationship exists, and the following condition checks out, a subgesture is reported:

$$\Delta F \left(p_{best}^{i_{sup}^0} \right) == 0. \quad (12)$$

Otherwise, a delay is observed when (12) is not true, that is, waiting for the next segment to avoid premature gesture spotting.

Gesture spotting with PSO matching does not require a traceback as gesture boundaries (start and end points) can be directly inferred from the p_{best} .

4. Experimental Results

To evaluate the performance of the proposed method, we collected gestures video clips at 30 FPS using Kinect v1, from five people, drawing in the air digits “0–9.” There was no restriction made on gesturing speed or size. Implementation was done using C++ and OpenCV libraries. PSO algorithm is implemented based on a discrete version [20], which is modified to track gestures and to accommodate the blind gesture spotting, and parameters for PSO are tuned by the speed of a gesturing hand. In the first round, each video contains ten or more gestures with pause between gestures,

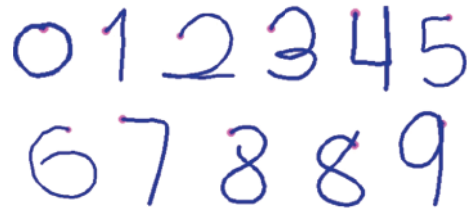


FIGURE 3: Gesture vocabulary.



FIGURE 4: Coarticulated gestures (left to right): 93, 58, 88, 34, and 81.

mounting to 50 gestures for each gesture category as shown in Figure 3. In the second round, each video file contained more than 2 random coarticulated gestures with possible connecting segments; see Figure 4. The evaluation of the effectiveness of the recognition method was performed using the following expression:

$$\text{recognition_rate} = \left(1 - \left(\frac{I + D + S}{N} \right) \right) \times 100, \quad (13)$$

where I stand for the number of inserted gestures, D missed gestures, and S substitution errors. The computed rate shows the prediction of the right response when these gestures are used in control systems or gaming.

The first evaluation was done on manually spotted gestures. An average recognition rate of 97.5% was obtained on 524 gestures, as shown in Table 1. The recognition rate shows that template matching based on the particle’s personal best positions performs well.

In the recognition on stream gestures with assistive boundary indicators, we perform temporal gesture segmentation as a preprocessing step; depth profile is used as temporal segmentation indicator, as the hand pull-back at the end of a gesture. The start of a gesture is detected when a hand is extended and starts to move. Similarly the end of the gesture is detected when a hand is retracted. The resulting gestures may

TABLE 2: Confusion matrix recognition of gesture with unknown boundaries.

	0	1	2	3	4	5	6	7	8	9
0	95.7%						4.3%			
1		92.6%			6.8%					
2			93.5%							
3				95.7%						
4					93.1%					6.9%
5						97.9%				
6	4.2%						95.8%			
7			2.9%					96%		
8					3.6%				91.5%	
9										98.1%

TABLE 3: Blind gesture spotting.

	Single gestures	Paired gestures
Number of gestures	130	45
Recognition rate	94.2%	86.36%

contain additional parts, pre- and postgestures considered as boundary noise, since they are not a part of the gesture; see Figure 1. Table 2 shows the confusion matrix; the average recognition rate on stream digits is 94.9%. The recognition rate dropped by almost 3%. Due to boundary noise, digit one is confused with four, four with nine, and seven with two. This can be explained by the sub-super gesture relations existing between confused gestures and the presence of boundary noise.

We evaluate blind gesture spotting on continuous gestures, without gesture boundary indicator or special treatment of connecting segments. Each video file contains two or more random gestures. For paired gestures, a pair is recognized, only if both gestures are correctly recognized. As shown in Table 3, the recognition rate of single gesture shows good performance despite coarticulation with the others. However, the recognition rate on paired gestures drops as gestures are recognized together.

We compare the proposed recognizer to the state-of-the-art methods in dynamic hand gesture recognition, based on the recognition accuracy and the processing time, on the same gesture vocabulary. Note that the experimental environment may be different; thus the following comparison is to be understood as the ability of the methods under consideration to discriminate a given gesture vocabulary and relative processing time, in general.

Table 4 summarizes performance of some implementation of gesture recognizers. As shown in Table 4, in their experiments, the authors of [13] evaluated their method, which models lines and curves using Von Mises Distribution (VMD) and realigns them according to gesture structures, by comparing it to a fixed number of states. The method produced good recognition rate, 97.1%, on preisolated gestures.

In [9], a recognition rate of 98.3% was achieved on training samples and 97.7% on test samples, showing that a well-trained HMM produces better rate, as reported in

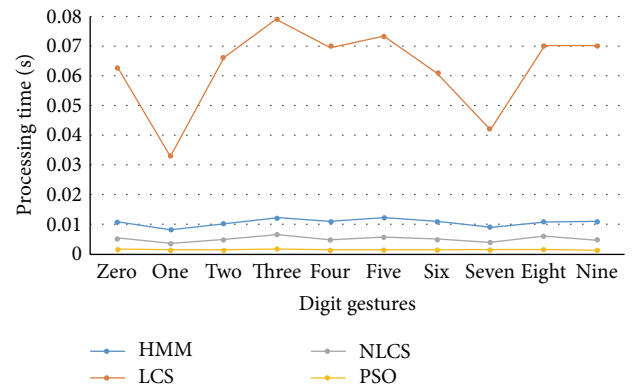


FIGURE 5: Processing time comparison.

different studies. The main contribution of [9] was to provide a gesture spotting mechanism, through the start and end points detection, under the HMM. A nongesture model was constructed by combining all HMM models of the gesture vocabulary and used as a threshold. They achieved a recognition rate of 93.3% on stream digits. The Most Probable Longest Common Subsequence (MPLCS) in [11] modeled different intragesture variations through the use of Mixture of Gaussian Models (GMM). Recognition rates of 98.7% and 94% were achieved on preisolated and stream gestures, respectively. In this work, we achieved on average, the recognition rates of 97.56% and 94.5%, on manually segmented and stream gestures with unknown boundaries, respectively.

Figure 5 shows comparative processing time of HMM, LCS, and PSO based gesture recognition on digit gesture vocabulary. Point matching LCS has the highest processing time. In our evaluation, the HMM requires approximately 16.8% of LCS, the normalized longest common subsequence (NLCS) with segment matching requires 8.1%, and the proposed PSO based matching method requires only 2.3% of the highest processing time.

5. Conclusion

An online gesture recognition method based on particle swarm optimization has been derived through this work.

TABLE 4: Dynamic gesture recognition with HMM, LCS, and PSO.

Paper	Representation	Isolated	Stream	Vocabulary	Method
[13], 2014	VMD	97.1%	—	Alpha number	HMM
[9], 2014	Codewords	97.7%	93.3%	Number	HMM
[8], 2015	Codewords	88.7%	—	Number	HMM
[11], 2013	GMM	98.7%	94.0%	Number	LCS
This work, 2016	Codewords	97.5%	94.2%	Number	PSO

Gesture vocabularies often contain highly correlated gestures which reduce the accuracy of most recognizers. Furthermore, the choice of assistive temporal segmentation cues is challenging and their usage may not be the right approach in the systems that need a string input such as word search or recognition. In this paper, we integrated a subgesture reasoning mechanism into the proposed PSO based matching method, to avoid premature gesture spotting.

The proposed recognizer showed promising recognition rate under different assumptions: a recognition rate of 97.5% was obtained on gestures with known boundaries, 94.9% on gesture with boundary noise, and 94.2% on continuous gesture without assistive boundary detection. A pairwise gesture recognition evaluation achieved a rate of 86.3%. The processing time was reduced to 2.3% of LCS point matching method. In the comparison to existing gesture recognition method, the proposed recognizer shows similar recognition rate and requires less processing time.

In the future work, we plan a further analysis on paired gestures, as some intergesture categories relationship cannot be detected from single gesture analysis basis.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (NRF-2015R1D1A1A01058394) and by the Chung-Ang University Research Scholarship grants in 2015.

References

- [1] J. P. 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.
- [2] S. Lian, W. Hu, and K. Wang, "Automatic user state recognition for hand gesture based low-cost television control system," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 107–115, 2014.
- [3] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in *Proceedings of the 4th ACM International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '11)*, May 2011.
- [4] J. H. Lee, T. Delbruck, M. Pfeiffer et al., "Real-time gesture interface based on event-driven processing from stereo silicon retinas," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2250–2263, 2014.
- [5] C. Nyirarugira and T. Kim, "Adaptive differential evolution algorithm for real time object tracking," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 833–838, 2013.
- [6] M. B. Holte, T. B. Moeslund, and P. Fihl, "View-invariant gesture recognition using 3D optical flow and harmonic motion context," *Computer Vision and Image Understanding*, vol. 114, no. 12, pp. 1353–1361, 2010.
- [7] L. Anthony, R. D. Vatavu, and J. O. Wobbrock, "Understanding the consistency of users' pen and finger stroke gesture articulation," in *Proceedings of the Graphics Interface (GI '13)*, pp. 87–94, Regina, Canada, May 2013.
- [8] C. Nyirarugira and T. Kim, "Stratified gesture recognition using the normalized longest common subsequence with rough sets," *Signal Processing: Image Communication*, vol. 30, pp. 178–189, 2015.
- [9] F. F. M. Ghaleb, E. A. Youness, M. Elmezain, and F. S. Dewdar, "Hand gesture spotting and recognition in stereo color image sequences based on generative models," *International Journal of Engineering Science and Innovative Technology*, vol. 3, no. 1, pp. 78–88, 2014.
- [10] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1685–1699, 2009.
- [11] D. Frolova, H. Stern, and S. Berman, "Most probable longest common subsequence for recognition of gesture character input," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 871–880, 2013.
- [12] L. Liu, S. Yang, and D. Wang, "Particle swarm optimization with composite particles in dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 6, pp. 1634–1648, 2010.
- [13] J. Beh, D. Han, and H. Ko, "Rule-based trajectory segmentation for modeling hand motion trajectory," *Pattern Recognition*, vol. 47, no. 4, pp. 1586–1601, 2014.
- [14] H.-K. Lee and J. H. Kim, "An HMM-Based threshold model approach for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961–973, 1999.
- [15] A. D. Wilson and A. F. Bobick, "Parametric hidden Markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.
- [16] A. Just and S. Marcel, "A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition," *Computer Vision and Image Understanding*, vol. 113, no. 4, pp. 532–543, 2009.

- [17] M. Ángel Bautista, A. Hernández-Vela, V. Ponce et al., “Probability-based dynamic time warping for gesture recognition on RGB-D data,” in *Advances in Depth Image Analysis and Applications*, vol. 7854, pp. 126–135, Springer, 2013.
- [18] S. Zhou, F. Fei, G. Zhang et al., “2D human gesture tracking and recognition by the fusion of MEMS inertial and vision sensors,” *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1160–1170, 2014.
- [19] B. Morris and M. Trivedi, “Learning trajectory patterns by clustering: experimental studies and comparative evaluation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 312–319, Miami, Fla, USA, June 2009.
- [20] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 4104–4108, Orlando, Fla, USA, October 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

