

Gesture Recognition using the linear combination of membership degrees of observations

Afshin Khoshraftar *

Computer Engineering Department, Islamic Azad University, Qazvin Branch, Qazvin, Iran

Received 23 November 2010; revised 27 February 2011; accepted 10 March 2011

Abstract

This paper introduces a novel gesture recognition method. In the method, hand trajectory is represented by the sequence of symbols and each symbol has a specific membership degree obtained from the genetic algorithm training. In order to determine the membership degree of input observations sequence in a class, the system uses the linear combination of membership degrees of observations in sequence. Because of using negative and positive samples for training gesture classes in the proposed method, the recognition system has a good performance in distinguishing very similar gestures. Experiments show that the method developed in this study outperforms HMM and SOMM methods in different gesture datasets.

Keywords: Gesture Recognition, Genetic Algorithm, Hidden Markov Model, SOMM

1. Introduction

Gesture recognition and gesture-based Human-Computer Interaction (HCI) have increasingly attracted the attention of researchers across disciplines such as machine learning, pattern recognition, computer vision, HCI, and linguistic and natural language processing. Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or the body with the intent of: 1) conveying meaningful information or 2) interacting with the environment [1]. Gesture recognition is the process through which the gestures made by the user are recognized by the receiver [1].

In this paper, we focus on hand gestures and information conveyed by them. To use human hands as a natural HCI, glove-based devices such as the Cyber Glove have been used to capture human hand motions. However, the gloves and their attached wires are still quite cumbersome and awkward for users to wear. Moreover, the cost of the gloves is often too expensive for regular users.

With the latest advances in the fields of computer vision, image processing, and pattern recognition, real-time vision-based hand gesture classification is becoming more and more feasible for HCI.

A gesture recognition system comprises of several subsystems that do specific tasks and interact with each other. Typically, these subsystems are hand detection, hand tracking, and recognition subsystems. Figure 1 shows the subsystems of the gesture recognition system and the interaction flows.

The hand detection subsystem is responsible for finding the hand and initializing the hand model for tracking in further frames. The hand tracking subsystem, sometimes known as the real-time hand detection, searches the hand in the current frame with respect to previous hand models and extracts the hand trajectory. Then it passes the hand trajectory to the recognition subsystem and updates the hand model. The recognition subsystem compares the input hand trajectory with pre-learned gestures classes and if some specific criteria are met, the trajectory will be recognized as one gesture class. The criteria can be a pre-defined hard threshold or more complex rules.

In the present paper, we introduce a novel gesture recognition method that uses a genetic algorithm for gesture class training. In the proposed system, hand

*Corresponding author. Email: afshin.khoshraftar@gmail.com

trajectories are represented by symbol sequences and each symbol has a specific membership degree which is determined by training. To determine the membership degree of input observations sequence in a class, the system uses the linear combination of membership degrees of all observations in sequence.

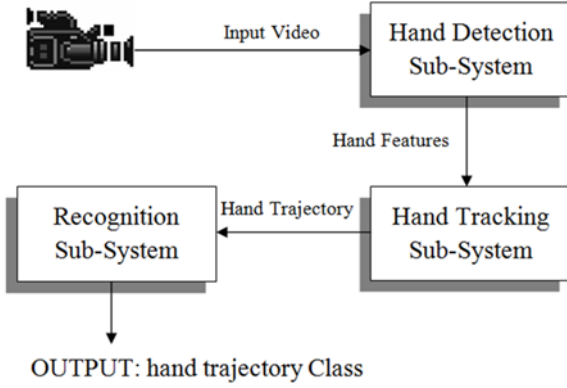


Fig. 1. Sub-Systems of the Gesture Recognition System

The rest of the paper is organized as follows. Section 2 discusses the related studies; Section 3 introduces the proposed system; Section 4 describes system training by a GA; Section 5 presents the feature extraction method; Section 6 shows experimental results of the overall system, and finally Section 7 concludes the article and provides some suggestions for further research.

2. Review of the literature

Gesture recognition can be considered as a time sequence series classification when the isolated movement sequence exists. It is also an instance of pattern spotting when the input sequence is continues [2]. Pattern spotting is the task of extracting meaningful segments from input signals and recognizing them in a predefined set of data [2], [3] and [4]. For a comprehensive review of pattern spotting methods for sequential data, refer to Dietterich [5], and for a good survey on gesture recognition to Mitra and Acharya [1].

There are several techniques used for gesture recognition. Being one of them, the Hidden Markov Model (HMM) [6] is a widely used model for gesture recognition with spatiotemporal variability. Coogan et al. also presented a quite common architecture involving color-based hand detection, scale and rotation invariant PCA classification for hand shape and discrete hidden Markov models (DHMMs) for position information. While they reported the adequate recognition rate of 94.5% for the static hand shape recognition, in two subject tests the recognition rate of the dynamic hand shape recognition varied from 83% to 98.6% depending on the training/testing ratio [7]. Interestingly, HMM has been also successfully applied in several sign language recognition systems, e.g., [8], [9], [10]. Nickel and Stiefelwagen used it

for pointing the gesture classification used in the human-robot interaction [11]. Being concerned with different versions of HMM, Just and Marcel provided a comparison of the two commonly used HMM versions. They compared the discrete HMM (DHMM) with the Input/output HMM (IOHMM) in different datasets of gestures and the results showed that DHMM is better than IOHMM regarding the recognition rate [12].

More recently, new techniques such as Conditional Random Fields (CRFs) [13] and its extension called Hidden Conditional Random Fields (HCRFs) [14] are used for recognition tasks. Unlike the HMMs, CRFs and HCRFs create a unique model for all gesture classes in a dataset and have some advantages over the HMMs. CRFs and HCRFs do not have the label bias problem [13]. Besides, they are able to model the dependency between all observations while the HMMs only model the dependency of observations related to their state. Therefore, as observations in gestures have a lot of dependencies to each other, CRFs and HCRFs can model them more accurately than the HMMs.

Another new approach to gesture recognition is SOMM presented by Caridakis et al. [15]. SOMM is a hybrid of a Self Organizing Map neural network (SOM) and a Markov Model (MM). It uses the capacity of SOM for the clustering of spatial data and the power of MM for the modeling of temporal information. Caridakis et al state that SOMM has a better recognition rate than the HMMs.

3. An overview of the system

As mentioned before, the recognition sub-system maps the input observations sequence to its corresponding gesture class. In this paper, each observation is represented by a specific symbol and each symbol has a membership degree in each class which is determined by a genetic algorithm in the training phase. To determine the membership degree of input observations sequence in a class, the system uses the linear combination of membership degrees of observations in sequence. More formally, if $S=\{s_1, s_2, \dots, s_n\}$ is the set of all possible observations of symbols and $C=\{c_1, c_2, \dots, c_m\}$ is the set of gestures classes, class C_i is represented by $M=\{d_{(i,1)}, d_{(i,2)}, \dots, d_{(i,j)}, \dots, d_{(i,n)}\}$ where $d_{(i,j)}$ is the membership degree of symbol S_j in class C_i . Now the membership degree of a given observation sequence $O=[O_1 O_2 O_3 \dots O_t \dots O_T]$ where O_t is the observation symbol at time t and T is the length of observation sequence in class C_i at time t is computed by equation (1) as follows:

$$MD_{(C_i, O, t)} = MD_{(C_i, O, t-1)} \times Z + d_{(i, O_t)} \quad (1)$$

$$MD_{(C_i, O, 1)} = d_{(i, O_1)}$$

Where $MD_{(C_i, O, t)}$ is the membership degree of observation sequence O in class C_i at time t and Z is the constant factor of previous observations membership

degrees. If Z is equal to zero, the membership degree of observation sequence O at time t only depends on O_t and $Z=1$ means that the order of observation sequence does not matter. In this situation, the recognition system is not able to recognize the different classes which have the same symbols with different occurrence times from each other. The experiments show that the best value for Z is 0.9. Therefore, the membership degree of observation sequence O in class C_i is equal to $MD_{(C_i,O,T)}$, and observation sequence O is a member of class C_k if $MD_{(C_k,O,T)}$ is the greatest membership degree.

4. Training of the recognition system

The aim of training is to determine the membership degree of each symbol in each gesture class based on the following conditions:

1. Time and number of occurrences of the symbol in the class;
2. Membership degrees of other symbols in the class and order of their occurrences;
3. The membership degree of the same symbol in other classes.

The importance of symbols is directly related to the first condition and indirectly to the second and third conditions.

Statistical models like the HMMs used for the modeling of gestures meet the first two conditions but do not satisfy the third one because they use positive examples for training the model. Yet, the third condition must be considered in many cases.

For example, suppose by the given digits classes (0, 1... 9) as hand gestures and the direction of hand movement between consecutive positions as the representation method, we have to distinguish the class of digit 2 from the class of digit 3 while they are represented by symbols in Figure 2. As Figure 2 shows, the classes of digits 2 and 3 are mostly formed of the same symbols and just differ in the directions which are drawn by the red color. The occurrence order of the symbols is also similar in both classes. In this situation, as the gestures have common sub-gestures, and common symbols cannot show the difference between these gestures, non-common symbols should be used. Therefore, during the training of a specific class if we neglect other classes which probably have common sub-gestures with the class in training, the low accuracy of classification may occur. This problem can be solved by adding negative samples to the training dataset.

4.1. Genetic Algorithm (GA)

Genetic algorithms (GAs) [16] are a kind of evolutionary computation derived from the evolution theory. According to the theory of evolution, within a population only the individuals well adapted to their environment can survive and transmit some of their characters to their descendants. In genetic algorithms, this

principle is introduced into the problem of finding the best individuals represented by chromosomes. Thus, each chromosome encodes a possible solution for a given problem, and starting from a population of chromosomes, the evolution process performs a parallel search through the solution's space. The fitness is measured for each individual by a function related to the objective function of the problem to be solved. Basically, a genetic algorithm consists of three major operations: selection, crossover, and mutation. The selection operation evaluates each individual and keeps only the fittest ones in the population. In addition to those fittest individuals, some less fit ones could be selected according to a small probability. The others are removed from the current population. The crossover operation recombines two individuals to have new ones which might be better. The mutation operation induces changes in a small number of chromosomes units since its purpose is to maintain the diversity of the population during the optimization process.

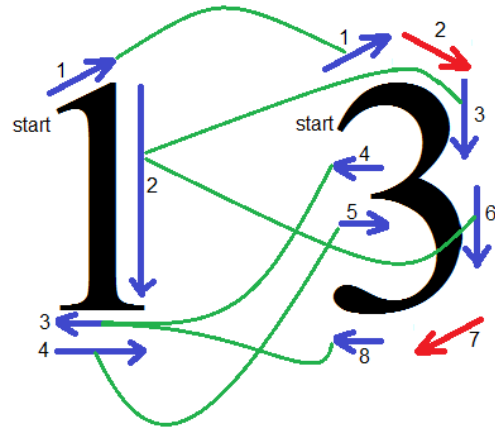


Fig. 2. Two digit gestures classes (2, 3) and the symbols. The direction symbols are shown by the arrows. The blue arrows are common in both gestures but the red arrows are not. Numbers on the arrows describe the order of symbols.

4.2. Problem formulation

If set $S = \{s_1, s_2, \dots, s_n\}$ is representative of all possible symbols, then the length of chromosomes are n which is equal to the number of symbols. Each unit of chromosomes called gene corresponds to the membership degree of a symbol which is in the range of $[-R, R]$ where R is a real value.

Also if $d = \{O_{(1,1)}, O_{(1,2)}, \dots, O_{(1,m)}, O_{(2,1)}, \dots, O_{(i,j)}, \dots, O_{(c,m)}\}$ is the set of training dataset where $O_{(i,j)}$ is the j 'th instance observation sequence for gesture class C_i and $MD_{(C_k, O_{(i,j)}, T)}$ is the membership degree of $O_{(i,j)}$ in class C_k , the fitness function of the GA for class C_k training is equation (2).

$$f_{ch(c)} = \frac{Tval}{R \times T} + (TP - FP) \quad (2)$$

If $MD_{(C_k, O_{(i,j)}, T)} > TH$ and $O_{(i,j)}$ is a Positive Sample

Then $TP = TP + 1, Tval = Tval + |MD_{(C_k, O_{(i,j)}, T)}|$

If $MD_{(c_k, O_{(i,j), T})} < TH$ and $O_{(i,j)}$ is a Positive Sample

Then $Fval = Fval + |MD_{(c_k, O_{(i,j), T})}|$

If $MD_{(c_k, O_{(i,j), T})} < TH$ and $O_{(i,j)}$ is a Negative Sample

Then $Tval = Tval + |MD_{(c_k, O_{(i,j), T})}|$

If $MD_{(c_k, O_{(i,j), T})} > TH$ and $O_{(i,j)}$ is a Negative Sample

Then $FP = FP + 1, Fval = Fval + |MD_{(c_k, O_{(i,j), T})}|$

Where $f_{ch(c)}$ is the fitness value of chromosome $ch(c)$, T is the length of $O_{(i,j)}$ and TH is the predefined threshold value.

It is worthy to mention that training is done separately for each class. The selection method for parent's selection is the tournament [17] where the best individual is selected as a parent among a randomly selected set of individuals with respect to the fitness values. Note that the number of parents corresponds to the population size. An advantage of the tournament method for parent selection is bearing the selection pressure and preventing the early convergence.

The selection method for the survivor selection is steady state [17] where some good individuals of the previous population are kept and others are replaced with offspring – children- population.

The crossover operation is an arithmetic crossover that combines two parent chromosomes $P1$ and $P2$ by applying equation (3) as follows.

$$Ch1 = P1 \times \beta + P2 \times (\beta - 1) \quad (3)$$

$$Ch2 = P2 \times \beta + P1 \times (\beta - 1)$$

Where β is a random number in the range of (0, 1) and $Ch1$ and $Ch2$ are two offspring.

The mutation operation on chromosome CH is applied by equation (4).

$$CH = CH + \alpha \quad (4)$$

Where α is a random number in the range of (-R/10, R/10). Note that crossover and mutation operators apply in every gene of chromosomes.

Finally, we add some domain knowledge to the GA to improve chromosomes fitness. Before the old population replacement stage in every generation, we define an additional operator called "modifier" which applies to some chromosomes with the highest fitness. This operator changes the membership degree of the symbols which are not seen in the positive samples of the class in the training dataset equal to $-R$. This means that if, for example, symbol s_j does not have any role in gesture class c_i , the membership degree of s_j in gesture class c_i must be very small.

Reaching to the maximum predefined number of generations while there is no improvement in the fitness of the best individual is the stop condition for training.

The algorithm of training is as follows:

- 1) For every class C_i , do step 2 to 10;
- 2) **initialize** the population with random individuals;
- 3) compute the **fitness** of individuals;
- 4) when the stop condition is not satisfied, do step 5 to 9;
- 5) **select** parents;
- 6) apply **crossover** and **mutation** operators;

7) compute the **fitness** of offspring;

8) apply **the modifier** to some best individuals;

9) **replace** the old population with the offspring population.

10) select **the best individual** of population as the **Model** of C_i .

5. Feature Extraction

In this study we use skin color segmentation and motion information for the hand detection and tracking as introduced by Yang et al. [18].

The hand tracker returns the hand trajectory as an absolute hand center coordinate in the 2D space related to the upper left corner of the frame. Therefore, in this step the hand trajectory can be represented by the sequence of its absolute (x,y) coordinates. For example, $O = [(x_1, y_1), (x_2, y_2), \dots, (x_t, y_t), \dots, (x_T, y_T)]$ is the trajectory of hand absolute coordinates in 2D space where (x_t, y_t) presents the hand coordinate at time t related to the upper left corner of the frame.

This representation for hand positions is not suitable since the distance from the camera and the position of the user within the frame during recording is not known beforehand. What's more, differences in the distance from the camera would result in differences in the height and arm length between users. We, therefore, need to normalize the hand trajectory so that it would become invariant to the user position and distance from camera. We use two normalizing methods:

In the first method, the head position is used as the coordinate center and the head size as the scaling parameter in this way: First, according to Rowley et al. [19], the head detector finds the head. Then the hand position is calculated in relation to the head position and its elements are normalized with respect to the head size. Finally, the conversion of hand coordinates to symbols is done by clustering hand coordinates using a Self Organizing Map (SOM) [20] trained by all hand positions in all trajectories of gesture classes in the training dataset. The number of SOM units which have a hexagonal architecture is determined through a trial-and-error process. Afterwards, each cluster is interpreted as a symbol. Note that clustering reduces noises caused by different hand trajectories from the same class which do not have the same coordinate sequence.

The second normalizing method does not require head detection and thus is simpler to implement. But it leads to more similar gestures classes, and in turn makes the recognition of different classes from each other difficult. In this normalization method, hand trajectories are represented by orientations instead of their coordinates.

In the second method, we use the mean orientations of five consecutive hand coordinates as local hand motion orientations. Then the obtained orientation is quantized by dividing it by 10 in order to generate symbols from 1 to 36.

If $O=[(x_1,y_1), (x_2,y_2), \dots, (x_t,y_t), \dots, (x_T,y_T)]$ is the trajectory of hand absolute coordinates in 2D space where (x_t,y_t) presents the hand coordinate at time t related to the upper left corner of the frame, the orientation of hand is computed by equation (5) as follows.

$$\theta_t = \frac{\sum_{i=t-4}^t \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right)}{5} \quad t=5 \dots T \quad (5)$$

6. Experimental results

In this section, we use two different hand gesture datasets for evaluating our proposed method for gesture

recognition against HMM- [12] and SOMM- [15] based gesture recognition. The first dataset contains 30 gaming gestures classes with each gesture class including 10 hand trajectories normalized by the first normalization method mentioned above. As Figure 3 shows, the classes vary in complexity from very simple directive gestures to very complex ones.

The second dataset has 10 gesture classes where each class, according to Figure 4, shows a number from zero to nine. The number of trajectories, normalized by the second normalization method, in each class is 10.

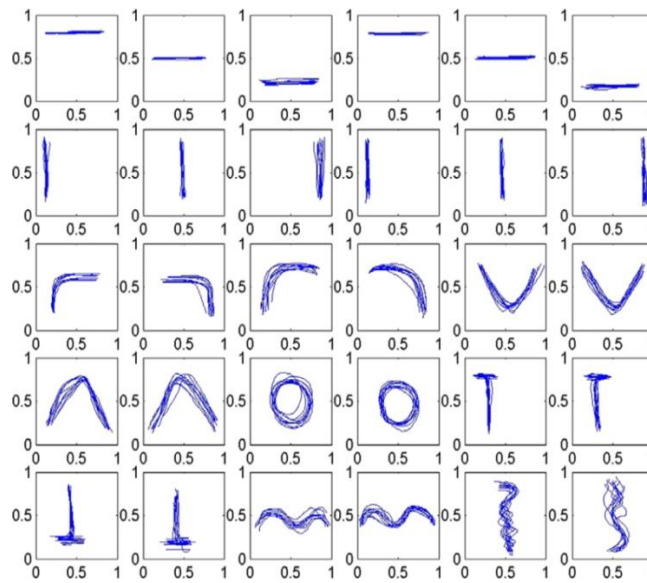


Fig. 3. Classes in gaming gesture dataset.

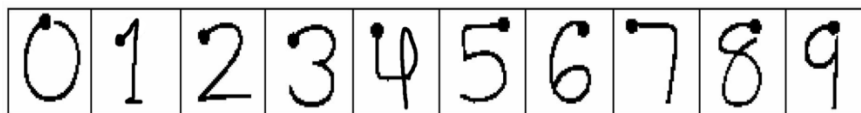


Fig. 4. Classes in gesture dataset of numbers.

In the HMM-based recognition, we use the discrete bounded left to right HMM architecture which is the most commonly used one in gesture recognition. In this architecture, the leftmost side state is the first state and the rightmost side state is the final state. Each state is only connected to itself and immediately to its left state. The number of states is determined through a trial-and-error process. We use the Baum-Welch algorithm for the HMM training [21].

However, in the SOMM-based recognition, two models represented by the Markov Model are created to describe the orientation and position information of each gesture class. Having the unknown gesture trajectory as the input for classification, SOMM finds the probability of this trajectory belonging to each class with respect to the both

models of each class, and finally selects the most probable class as the input gesture trajectory class.

It should be pointed out that since SOMM needs hand position and hand orientation for training, it is only evaluated in the first dataset.

For the evaluation within the gaming gestures dataset, we use %50 of each class samples as the training set and %50 as the test set. Table 1 shows the accuracy of different methods for the training and test sets. Furthermore, for evaluating the gestures dataset of numbers, %60 of each class samples is used as the training set and %40 as the test set. Table 2 reports the accuracy of the method proposed in this paper and HMM for this dataset.

Table 1
Accuracy of HMM, SOMM and the Proposed Method for the gaming gestures dataset.

Method	Accuracy in training set	Accuracy in test set
HMM	99%	78%
SOMM	100%	93%
Proposed	100%	96%

Table 2
Accuracy of HMM and the Proposed Method for the gestures dataset of numbers

Method	Accuracy in training set	Accuracy in test set
HMM	100%	50%
Proposed	100%	80%

According to Tables 1 and 2, the proposed method outperforms HMM and SOMM methods in accuracy. In addition, when the gestures classes are similar, the method developed in this study has a good accuracy while HMM treats an almost random classification (see Table 2).

Other advantages of the proposed method for gesture recognition are faster recognition and smaller size of models. In the recognition stage, the method is linear in terms of time and only depends on the number of symbols in the trajectory. However, HMM's addition to the number of symbols depends on the number of states, and SOMM needs to do more preprocessing for extracting features and also needs more storage space to store models of gestures. Table 3 shows the time order in the recognition and storage order of each method when the number of symbols is S , the number of classes is C , the length of trajectory is T , and the number of states in HMM is N .

Table 3
Time Order in the recognition and Storage Order of each method when the number of symbols is (S), the number of classes is (C), the length of trajectory is (T) and the number of states in HMM is (N).

Method	Time Order	Storage Order
HMM	$O(TNC)$	$O(NSC)$
SOMM	$O(2TC)$	$O(2S^2C)$
Proposed	$O(TC)$	$O(SC)$

All in all, our proposed method for gesture recognition is better than HMM and SOMM regarding the accuracy and speed of recognition, and needs less storage space to store models of gestures.

7. Conclusions

In the present paper, a novel gesture recognition method is proposed. In the method hand trajectories are represented by the sequence of symbols and each symbol has a specific membership degree in each gesture class which is determined by a genetic algorithm (GA). By the given input hand trajectory, the system uses the linear combination of

membership degrees of observations in sequence to determine membership degrees of the input trajectory in each class. Then the class which includes the hand trajectory with the greatest membership degree is chosen as the trajectory class.

Two normalization methods for symbols representation are used in this paper, and the proposed method is compared with HMM and SOMM in different datasets. Experiments indicated that our proposed method outperforms HMM and SOMM in the accuracy of gesture recognition in both datasets. It is also faster than HMM and SOMM, and needs less storage space to store the gestures models.

Future research can compare the method suggested in this study with other new methods like CRFs and HCRFs for more evaluations. Another future research direction may concern finding suitable segmentation methods for using our proposed method on continuous hand trajectories and sign language spotting.

References

- [1] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 37, NO. 3, pp. , MAY 2007
- [2] J. Alon, "Spatiotemporal Gesture Segmentation," PhD thesis, Computer Science Dept., Boston Univ., 2006.
- [3] J. Alon, V. Athitsos, and S. Sclaroff, "Accurate and Efficient Gesture Spotting via Pruning and Subgesture Reasoning," Proc. IEEE Human Computer Interface Workshop, pp. 189-198, Oct. 2005.
- [4] H.-K. Lee and J.-H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 10, pp. 961-973, Oct. 1999.
- [5] T.G. Dietterich, "Machine Learning for Sequential Data: A Review," Proc. Joint IAPR Int'l Workshops Structural, Syntactic, and Statistical Pattern Recognition, pp. 15-30, Aug. 2002.
- [6] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, vol. 77, pp. 257-286, 1989.
- [7] Coogan, Thomas, Awad, George, Han, Junwei, Sutherland, Alistair, 2006. Real time hand gesture recognition including hand segmentation and tracking. Adv. Visual Comput., 495-504.
- [8] W. Gao, G. Fang, D. Zhao, and Y. Chen, "Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition," Proc. Sixth IEEE Int'l Conf. Automatic Face and Gesture Recognition, pp. 553-558, May 2004.
- [9] T. Starner, J. Weaver, and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pp. 1371-1375, Dec. 1998.
- [10] C. Vogler and D. Metaxas, "A Framework for Recognizing the Simultaneous Aspects of American Sign Language," Computer Vision and Image Understanding, vol. 81, no. 3, pp. 358-384, 2001.
- [11] K. Nickel, R. Stiefelhagen, "Visual recognition of pointing gestures for human-robot interaction," Image and Vision Computing 25, pp. 1875-1884, 2007.
- [12] A. Just, S. Marcel, "A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition," Computer Vision and Image Understanding 113, pp. 532-543, 2009.
- [13] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence

- Data,” Proc. 18th Int’l Conf. Machine Learning, pp. 282-289, June 2001.
- [14] A. Quattoni, S. Wang, L.P. Morency, M. Collins, and T. Darrell, “Hidden Conditional Random Fields,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 10, pp. 1848-1852, Oct. 2007.
- [15] G. Caridakis, K. Karpouzis, A. Drosopoulos, S. Kollias, “SOMM: Self organizing Markov map for gesture recognition,” Pattern Recognition Letters 31, pp. 52–59, 2010.
- [16] J. H. Holland, *Adaptation in Natural and Artificial Systems*, AnnArbor, MI: Univ. of Michigan Press, 1975.
- [17] A.E.Eiben and J.E. Smith, *Introduction to Evolutionary Computing*, Springer, Natural Computing Series, 2nd printing, 2007.
- [18] R. Yang, S. Sarkar, B. Loeding, “Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming,” IEEE TRANS. ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 32, NO. 3, pp. 462- 477, MARCH 2010.
- [19] H. Rowley, S. Baluja, and T. Kanade, “Neural Network-Based Face Detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, Jan. 1998.
- [20] T. Kohonen, *Self-Organizing Maps*, Springer, 3rd edition, 2001.
- [21] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, “A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains,” The Annals of Math. Statistics, vol. 41, no. 1, pp. 164-171, 1970.