Multimed Tools Appl (2014) 72:843–864
DOI 10.1007/s11042-013-1368-1

A high-performance training-free approach for hand gesture recognition with accelerometer

Liang Yin · Mingzhi Dong · Ying Duan · Weihong Deng · Kaili Zhao · Jun Guo

Published online: 2 March 2013

Abstract In previous research on human machine interaction, parameters or templates of gestures are always learnt from training samples first and then a certain kind of matching is conducted. For these training-required methods, a small number of training samples always result in poor or user-independent performance, while a large quantity of training samples lead to time-consuming and laborious sample collection processes. In this paper, a high-performance training-free approach for hand gesture recognition with accelerometer is proposed. First, we determine the underlining space for gesture generation with the physical meaning of acceleration direction. Then, the template of each gesture in the underlining space can be generated from the gesture trails, which are frequently provided in the instructions of gesture recognition devices. Thus, during the gesture template generation process, the algorithm does not require training samples any more and fulfills training-free gesture recognition. After that, a feature extraction method, which transforms the original acceleration sequence into a sequence of more user-invariant features in the underlining space, and a more robust template matching method, which is based on dynamic programming, are presented to finish the gesture recognition process and enhance the system performance. Our algorithm is tested in a 28-user experiment with 2,240 gesture samples and this training-free algorithm shows better performance than the traditional training-required algorithms of Hidden Markov Model (HMM) and Dynamic Time Warping (DTW).

L. Yin · M. Dong · Y. Duan · W. Deng (✉) · K. Zhao · J. Guo
Beijing University of Posts and Telecommunications, Beijing, China
e-mail: whdeng@bupt.edu.cn

L. Yin
e-mail: Yin@bupt.edu.cn

M. Dong
e-mail: 2008dmz@gmail.com

Y. Duan
Beijing Normal University, Beijing, China

Keywords Training-free · Gesture recognition · Accelerometer

1 Introduction

As computers and mobile devices are becoming increasingly important in our daily life, many researches are focused on human machine interaction. Traditional methods, such as mouse, keyboard or button based remote control, bore the youngsters and, more seriously, trouble the blind and the old, since they can not see the buttons on the keyboard or remoter controller clearly, nor can they track the trail of the mouse. Therefore it is necessary to propose other human machine interaction methods. Hand gesture, which is a natural way of communication between individuals, becomes one of the most suitable ways and attracts a large number of research groups [15, 24, 35, 37, 41].

With the rapid development of sensors, 3D accelerometers are becoming much cheaper and more widely used in mobile devices. Accelerometer based gesture recognition has been highlighted by a number of research groups. For example, Elisabetta Farella, et al., detected the movement of the human body by analyzing the accelerometer network [9]. Wolfgang Hürst et al., chose the accelerometer, together with the compass, to fulfill gesture-based interaction for mobile augmented reality [14]. Andrew Wilson et al., implemented a gesture based interaction device called Xwand [38]. Cho et al., proposed Magic Wand to control home electronic facilities [7]. Chen et al., adopted the gesture-aware device as a presentation tool [6]. There has been numerous researches on the applications of the acceleration sensor in mobile phone platform [3, 5, 8, 19–21, 36]. Also, accelerometer based gesture researches have shown great potential for applications in e-Learning/e-Teaching [11–13].

In previous work, the gesture recognition algorithms can be mainly divided into two categories: template based methods and model based methods, as shown in Table 1. Among the template based methods, which set training examples as gesture templates, Dynamic Time Warping (DTW) is perhaps the most frequently used algorithm. It is easy to implement and requires only one training sample to initiate. DTW based researches have shown satisfied performance in a number of user-dependent systems [2, 17, 40]. However, since people may have various personalities when performing the same gesture, setting gesture samples from one person as templates always results in poor performance when used by others. Thus DTW fails in the user-independent applications [39]. In contrast, model based methods learn the parameters which describe each gesture from a large quantity of training samples. Thus, HMM, which is the most widely used model based method, can perform well in many user-independent applications [16, 23, 27, 30]. Also, Hidden Conditional Random Fields (HCRF) [33], Dynamic Bayesian Networks (DBN) [34], and Self Organizing Networks (SON) [10] have been implemented in some previous gesture recognition systems. However, the determination of the parameters in model based method must require a large number of training samples from different users, which leads to a time-consuming and laborious sample collection process. But with a limited number of training samples, there will be a sharp decline in system performance.

Whether template based or model based methods, the templates or parameters of the gestures are always learnt from training samples first and then certain kind of matching is conducted to output the classification result. For these training-required

Table 1 Traditional gesture recognition methods

Category	Template based methods	Model based methods
Example	Dynamic time warping (DTW)	Hidden Markov model (HMM)
Strengths	Satisfied user-dependent performance; One sample to initiate; Easy to implement	Satisfied user-independent performance (with enough training samples)
Weaknesses	Poor user-independent performance	Number of training samples: Large → laborious sample collection process Small → poor performance

methods, a small number of training samples always result in poor performance and a large quantity of training samples result in time-consuming and laborious sample collection process. Thus, training-free recognition is becoming a hot research topic in many fields recently [26, 31]. Without useful information of the gestures, training-free gesture recognition seems a task impossible. However, standard gesture trails, the crucial prior information ignored in previous researches, are frequently provided in the instructions for user-independent gesture recognition applications. With this prior knowledge, we can do much more.

In this paper, by exploiting gesture trail information, high-performance training-free hand gesture recognition with accelerometer is fulfilled. First, we determine the underlining space for gesture generation with the physical meaning of acceleration direction. Then, the template of each gesture in underlining space can be generated from gesture trails directly, which are frequently provided in the instructions of gesture recognition devices. Thus, during the gesture template generation process, fulfills the training-free gesture recognition without requiring training samples. After that, a feature extraction method, which transforms the original acceleration sequence into a sequence of more user-invariant features in underlining space, and a more robust template matching method, which is based on dynamic programming, are presented to finish the gesture recognition process and enhance the system performance. To test the algorithm's performance, 2,280 gesture samples are collected from 28 persons for 8 gestures used in DVD control [18]. An accuracy of 93.1 %, which is even better than standard training-required methods of DTW(an accuracy of 88.7 % for the best template, and an accuracy of 72.9 % on average) and HMM(an accuracy of 91.1 % when using 90 % percentage samples as training ones), is achieved by our training-free method of MMDTW in the user-independent experiment.

The remainder of the paper is organized as follows. First, classical algorithms of DTW and HMM are discussed in Section 2. In Section 3, key points in designing high-performance training-free algorithm are addressed. Then Section 4 proposes the algorithm for training-free accelerometer based gesture recognition. Section 5 tests the performance of the algorithm and some conclusions are drawn in Section 6.

2 Related work

This section discusses the classical algorithms for gesture recognition: DTW and HMM.

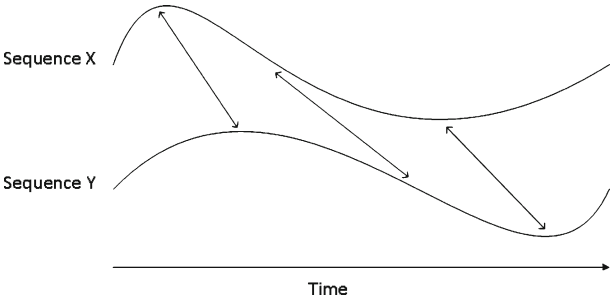


Fig. 1 The time alignment of two sequences varying in length

2.1 DTW

DTW is a well-known algorithm for measuring the similarity between two sequences which may vary in length [22] (Fig. 1). For two sequences of $\mathbf{X} := (x_1, x_2, \ldots, x_N)$ of length $N \in \mathbb{N}$ and $\mathbf{Y} := (y_1, y_2, \ldots, y_M)$ of length $M \in \mathbb{N}$, we define a *local cost measure* $c(x_n, y_m) \geq 0$,[1] which depicts the difference between x_n and y_m, we can obtain the *cost matrix* $C \in \mathbb{R}^{N \times M}$ defined by $C(n, m) := c(x_n, y_m)$. Then (N, M)-warping path is a sequence $p = (p_1, \ldots, p_L)$ with $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ for $l \in [1 : L]$ satisfying the following three conditions.

1. Boundary condition: $p_1 = (1, 1)$ and $p_L = (N, M)$;
2. Monotonicity condition: $n_1 \leq n_2 \leq \cdots \leq n_L$ and $m_1 \leq m_2 \leq \cdots \leq m_L$;
3. Step size condition: $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $l \in [1 : L - 1]$.

The *total cost* $C_p(X, Y)$ of a warping path p between X and Y with respect to the local cost measure c is defined as.

$$c_p(X, Y) := \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}). \tag{1}$$

An *optimal warping path* is a warping path p^* having minimal total cost among all possible warping paths. The *DTW distance* DTW(X, Y) between X and Y is then defined as the total cost of p^*,

$$DTW(X, Y) := c_{p^*}(X, Y) = \min\{c_p(X, Y) | p \text{ is an } (N, M)\text{-warping path}\}. \tag{2}$$

With prefix sequences $X(1 : n) := (x_1, \ldots, x_n)$ for $n \in [1 : N]$ and $Y(1 : m)$, we can define:

$$D(n, m) := DTW(X(1 : n), Y(1 : m)). \tag{3}$$

DTW finds an alignment between X and Y with minimal overall cost $DTW(X, Y) = D(N, M)$ by dynamic programming with the following recursive process.

1. Initialization:
   $D(n, 1) = \sum_{k=1}^{n} c(x_k, y_1)$, where $n \leq N$;
   $D(1, m) = \sum_{k=1}^{m} c(x_1, y_k)$, where $m \leq M$.

[1] $c(x, y)$ is smaller (lower cost) when x and y are more similar to each other.

2. Iteration:

$$D(n,m) = min \begin{cases} D(n-1,m) & + C(n,m) \\ D(n-1,m-1) & + C(n,m) \\ D(n,m-1) & + C(n,m) \end{cases};$$

where $1 < n \leq N$,$1 < m \leq M$.

In summary, the key steps of DTW are

1. determine the local cost measure function $c(x_n, y_m) \geq 0$ and obtain the count cost matrix $C \in \mathbb{R}^{N \times M}$;
2. output overall cost $D(N, M)$ via dynamic programming.

In practical, the above process is performed between the sequence for classification ($\mathbf{X}$ of length N) and the delegates of each class ($\mathbf{Y_i}$ of length M_i). Then the classification result is given as the best matched class $C = arg\min_i D_i(N, M_i)$.

2.2 HMM

2.2.1 Introduction of HMM

HMM is a famous and widely applied model [28] dealing with the sequential data efficiently. In the model, we assume that it is the latent variables $\mathbf{x}_i$ which form a Markov chain that give rise to the observations $\mathbf{z}_i$. As a generative model shown in Fig. 2, HMM is depicted by the following three sets of parameters: (1) Initial probabilities $\pi_k \equiv p(z_{1k} = 1)$; (2) Transition probabilities $A_{jk} \equiv p(z_{nk} = 1|z_{n-1,j} = 1)$, determining the relationship between adjacent latent states under Markov assumption, where z_{nk} indicates the kth states of the latent variable state in time n; (3) Emission probabilities $p(\mathbf{x}_n|\mathbf{z}_n, \phi)$, determining the relationship between the latent states and the observations in time n, where ϕ indicates the parameters of the emission distribution.

According to the relationship between variables, we can get the joint distribution of all variables in the model(both latent variable and observations) as follows:

$$p(\mathbf{X}, \mathbf{Z}|\theta) = p(\mathbf{z}_1|\pi)\left[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A})\right]\prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m, \phi); \quad (4)$$

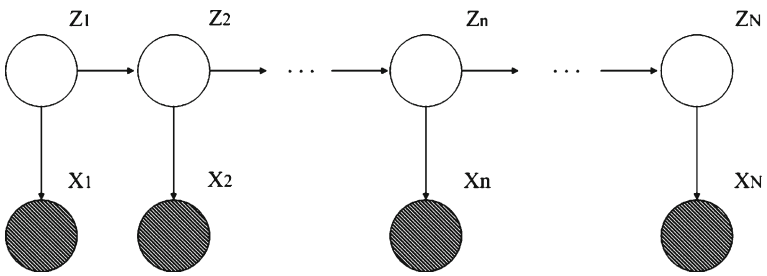


Fig. 2 Graphical structure of hidden Markov model

where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, $\theta = \{\pi, \mathbf{A}, \phi\}$ represent all parameters in the model.

2.2.2 Training process of HMM

When using HMM for classification, we should first determine the model parameters for each class $i : \theta_i = \{\pi_\mathbf{i}, \mathbf{A_i}, \phi_\mathbf{i}\}$ with the training samples of class i. The (local optimal) maximum likelihood solution can be determined by Expectation Maximum (EM) Algorithm:[2]

1. E-step: $\theta^{old} = \theta^{new}$, obtain $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{old})$;
2. M-step: Maximize $Q(\theta, \theta^{old}) = \sum_\mathbf{Z} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) lnp(\mathbf{X}, \mathbf{Z}|\theta)$ with respect to model parameters θ, obtain θ^{new}.

When adopting EM algorithm in HMM, by defining

$$\gamma(\boldsymbol{z}_n) = p\left(\boldsymbol{z}_n|\boldsymbol{X}, \boldsymbol{\theta}^{old}\right), \tag{5}$$

$$\xi\left(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n\right) = p\left(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n|\boldsymbol{X}, \boldsymbol{\theta}^{old}\right), \tag{6}$$

$$\gamma(z_{nk}) = \mathbb{E}[z_{nk}] = \sum_{\boldsymbol{z}} \gamma(\boldsymbol{z}) z_{nk}, \tag{7}$$

$$\xi\left(z_{n-1,j}, z_{nk}\right) = \mathbb{E}[z_{n-1,j} z_{nk}] = \sum_{\boldsymbol{z}} \gamma(\boldsymbol{z}) z_{n-1,j} z_{nk}. \tag{8}$$

the optimization objective function in M-step can be reformulated as,

$$Q\left(\theta, \theta^{old}\right) = \sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln \boldsymbol{A}_{jk} + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\boldsymbol{x}_n|\boldsymbol{\phi}_k). \tag{9}$$

The updating function of parameters can be obtained by using Lagrange multipliers,

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})}; \tag{10}$$

$$A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})}. \tag{11}$$

[2]The EM algorithm used in HMM is also called Baum Welch Algorithm.

The updating function of ϕ is related to the formulation emission probabilities $p(\mathbf{x}_n|\mathbf{z}_n, \phi)$. The quantities $\gamma(\boldsymbol{z}_n)$ and $\xi(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n)$ in each iteration can be efficiently computed via Sum-Product Algorithm.[3]

2.2.3 Classification process of HMM

Then the likelihood for a sequence belonging to each class $P(\mathbf{X}|C_i) = P(\mathbf{X}|\theta_i)$ can be efficiently determined. According to Bayesian rule, the posterior probabilities $P(C_i|\mathbf{X})$ can be obtained and the final recognition result is achieved:

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|\theta_i)\,P(C_i)}{\sum_i P(\mathbf{X}|\theta_i)\,P(C_i)}. \tag{12}$$

3 Well-performed training-free algorithm design

To recognize a gesture, it's significantly important to analyze the generation process of the acceleration data and then find out the reason why the traditional methods perform well or not. Based on the analysis results, we can design well-performed and training-free algorithms.

3.1 Gesture generation process

When we are conducting a gesture, the brain analyzes the trail of the gesture first and divides it into a series of hand operations in turn. Then, individual discrepancy is also addressed due to habitual hand motion. Eventually, after the accelerometer senses the data and outputs it, the acceleration sequences are observable. In the above process, as illustrated in Fig. 3, the acceleration sequences are observations obtained from the accelerometer. There are underlining factors determining these observations, namely motion process in our case. For the same gestures performed by different people, though the observations vary greatly, the underlining factors are roughly the same.

3.2 Analysis of traditional algorithms

DTW chooses one or more acceleration samples as templates for each gesture. Then the matching is performed in the observable space of acceleration sequences directly. Since Dynamic Programming based matching can capture long term dependencies of sequences [29], DTW performs well in user dependent applications [17]. However, in the case of user-independent, it is nearly impossible for DTW to get rid of the personal features and thus DTW fails to show satisfied performance.

HMM, in a different way, depicts the underlining factors for gesture generation via latent variables. Consequently, it has displayed a rather desirable performance in user-independent applications. However, HMM depicts the relationship between

[3]The Sum-Product Algorithm used in HMM is also called Forward-backward Algorithm.

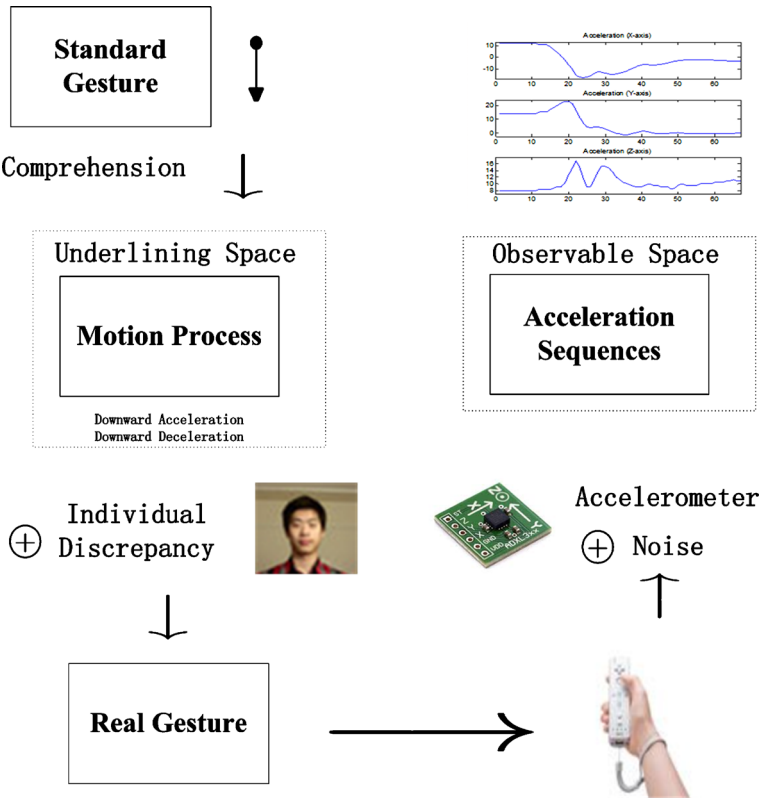


Fig. 3 The generating process of the gesture data

latent variables via 1-order transition matrix. Thus it can not discriminate gestures with similar transition matrix and fails to model the long term dependencies (high order relations) between underlining factors [25, 32].

3.3 Well-performed training-free algorithm design

3.3.1 Key points for well performance

Inspired by the analysis in the previous Sections of 3.1 and 3.2, to obtain well-performed gesture recognition algorithms, it is important for us to consider the following aspects during the design of the algorithm:

- Underlining Space: the underlining space should be able to depict the user-invariant underlining factors of gestures.
- Matching Algorithm: the matching should be able to capture long term dependencies.

3.3.2 Key points for training-free

Assume gesture trails and observations of acceleration sequences are given in the system, to fulfill training-free gesture recognition, the following aspects should be considered during the design of the algorithm:

- Generation of gesture templates in underlining space: atomic gestures should be defined so that gesture trails can be made full use of and generate the gesture templates in underlining space can be generated without training samples.
- Observation representation in underlining space: an unsupervised feature extraction should be designed to map the original acceleration sequences into the underlining space.
- Training-free matching algorithm: the designed matching algorithm should have no parameters to learn.

3.3.3 Framework of our well-performed training-free algorithm

It's convenient and suitable for us to determine the underlining space with the physical meaning of acceleration direction for the following reasons:

- Provided the gesture trails, their corresponding templates of acceleration direction sequences can be analyzed conveniently;
- The observations of acceleration sequences can be mapped into acceleration direction sequences conveniently;
- Acceleration direction sequences enjoy user-invariant property.

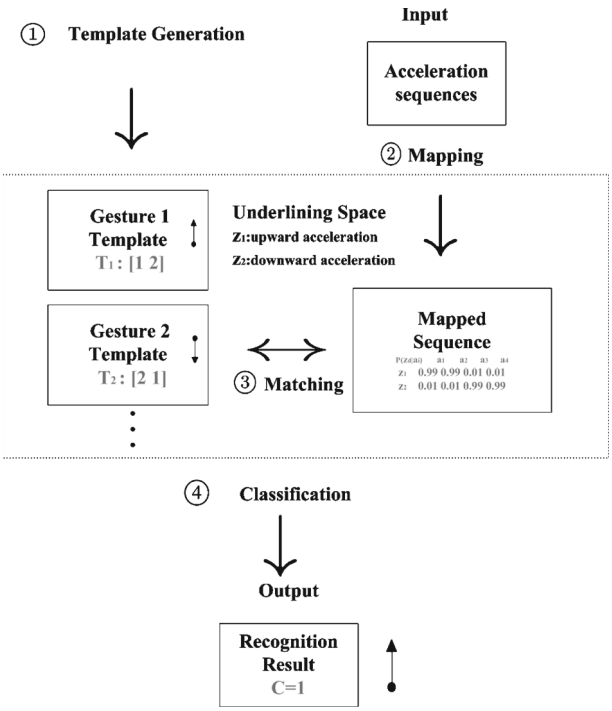


Fig. 4 Flowchart of the proposed mapping based modified DTW

Thus in the underlining space, we define the atomic gestures as the acceleration action in certain directions. Then we can obtain the template of each gesture and the mapped sequences of accelerometer data. After that, the modified DTW matching is conducted to obtain the recognition result since this matching strategy of dynamic programming captures long term dependencies better than 1-order transition matrix in HMM [29].

- The Underlining Space (Physical Meaning of Acceleration Direction)
  - Gesture template generation (*no training samples required*):
    Gesture trails → Templates of gestures (atomic gesture sequences);
  - Observation Feature Extraction (*unsupervised*):
    Acceleration sequences → Mapped sequences (probabilistic atomic gesture sequences);
- Matching via Modified DTW (*no parameters to train*).

For the process of our algorithm is mainly comprised of template generation, mapping, matching and classification, our algorithm is called Mapping based Modified DTW (MMDTW), which is shown can see seen in Fig. 4.

4 Process of our algorithm

In this section, the process of our algorithm will be discussed in details.

4.1 Gesture template generation

In this gesture recognition process, we make the following definitions: X indicates the right-left direction (left positive), Y indicates the vertical direction (upward positive), and Z indicate the backward-forward direction (forward positive). As shown in Fig. 5, the Nokia gestures defined in [18] will be discussed as an example. Since the gestures are all defined in XY space, the atomic gestures are

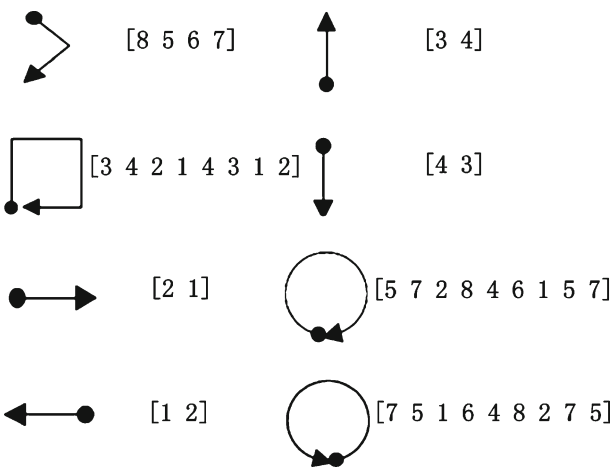


Fig. 5 Template sequences of gestures

represented by normalized acceleration vectors with Z dimension always remaining 0: $\mathbf{z}_1 = (1, 0, 0)$, $\mathbf{z}_2 = (-1, 0, 0)$, $\mathbf{z}_3 = (0, 1, 0)$, $\mathbf{z}_4 = (0, -1, 0)$, $\mathbf{z}_5 = (\sqrt{2}/2, \sqrt{2}/2, 0)$, $\mathbf{z}_6 = (\sqrt{2}/2, -\sqrt{2}/2, 0)$, $\mathbf{z}_7 = (-\sqrt{2}/2, \sqrt{2}/2, 0)$, $\mathbf{z}_8 = (-\sqrt{2}/2, -\sqrt{2}/2, 0)$.[4]

Now under the following assumptions, the gesture templates can be generated according to straight forward physical analysis:

1. For basic gestures moving uni-directionally, there exists both an acceleration process and a deceleration process.
   For example, the left moving gesture in Fig. 5 can be regarded as a left acceleration (corresponding to the underlining state $\mathbf{z}_1$) followed by left deceleration (correspond to the underlining state $\mathbf{z}_2$). Thus the template for this gesture is written as [1 2].[5]
2. For complex gestures originated from basic gestures, assumption 1 follows during each basic gesture process.
   For example, gesture square in Fig. 5 is comprised of up [3 4], right [2 1], down [4 3], left [1 2]. Since each basic gesture follows assumption 1, we can give out the total template for square as [3 4 2 1 4 3 1 2].
3. For circle gestures, they contain the following sub-motions,

   (1) a uniform motion during the whole gesture process,
   (2) an acceleration process when the gesture starts,
   (3) an deceleration process at the end of the gesture.

   For example, gesture clock-wise circle in Fig. 5 contains (a) Uniform Circular Motion: approximated by underlining states as [3 7 2 8 4 6 1 5 3]; (b) Beginning Acceleration: [1]; (c) Ending Acceleration: [2]. When we adding submotion b to the first state of submotion a, together with submotion c added to the final state of submotion a, we get the template for clock-wise circle: [5 7 2 8 4 6 1 5 7].

Under above assumptions, we can get the template sequences of gestures $\mathbf{T} = [t_1, \ldots, t_m, \ldots, t_M]$ as shown in Fig. 5. And the fact the mth element of sequence $\mathbf{T}$, t_m, numbered d indicates the underlining state at this time to be $\mathbf{z}_d$.

4.2 Feature extraction

This subsection will describe how to map the acceleration sequence of length N into the underlining space and how to present the mapped sequence $\mathbf{L} = (\mathbf{l}_1, \ldots, \mathbf{l}_i, \ldots, \mathbf{l}_N)$. In most accelerometers, the raw acceleration sequence contains the influence of gravity. Here we minus g in the vertical direction and get sequence $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_i, \ldots, \mathbf{a}_N)$.

In order to minimize the loss in the mapping process, soft mapping is adopted in our algorithm. With the distance between $\mathbf{a}_i$ and the underlining state $\mathbf{z}_d$ set, the distance is mapped into probabilities with Softmax function:

$$p(\mathbf{z}_d|\mathbf{a}_i) = \frac{exp(a_{id})}{\sum_d exp(a_{id}))}; \tag{13}$$

[4]The lasting time for each atomic gesture to continue will be considered in the step of matching.

[5]Number d here indicates the underlining state of $\mathbf{z}_d$ for simplicity.

where a_{id} is determined by

$$a_{id} = \frac{1}{\epsilon + dis(\mathbf{a}_i, \mathbf{z}_d)}; \tag{14}$$

where a small positive number ϵ is added to avoid a_{id} being infinite.

By assigning $l_{id} = p(\mathbf{z}_d|\mathbf{a}_i)$, we can obtain a probabilistic underlining sequence $\mathbf{L}$, where l_{id} indicates the probability of $\mathbf{a}_i$ to be underlining state $\mathbf{z}_d$.

4.3 Modified DTW based matching

This subsection will describe the matching method between the mapping result $\mathbf{L}$ and the template sequence of a gesture $\mathbf{T}$ in the underlining space.

4.3.1 Cost matrix

The cost matrix $C(i, m)$ indicates the distances between the ith element in the mapped sequence $\mathbf{L}$: $\mathbf{l}_i$ and the mth element in the template sequence $\mathbf{T}$: t_m decided by.

$$C(i, m) = w_i \cdot d_{im} \tag{15}$$

where w_i is the weight of $\mathbf{l}_i$ and d_{im} is the distance between $\mathbf{l}_i$ and t_m.

Since points with larger amplitudes are more important for recognition, w_i is given in proportion to the amplitude of the point as

$$w_i = |\mathbf{a}_i|_2. \tag{16}$$

In particular, if $t_m = d$, then the distance between t_m and $\mathbf{l}_i$ can be measured in proportion to $l_{id} = p(\mathbf{z}_d|\mathbf{a}_i)$. Thus we define the following piecewise function

$$d_{im} = \begin{cases} 1 - l_{id} & \text{if } l_{id} > \theta \\ k(1 - l_{id}) & \text{if } l_{id} < \theta \end{cases} \tag{17}$$

where $0 < t < 1$. The threshold θ indicates that when l_{id} is very small, it's nearly impossible for $\mathbf{l}_i$ to be t_m and the distance between $\mathbf{l}_i$ and t_m should be very large. In this case, we multiply the distance by $k > 1$.
So that $C(i, m)$ is given by the following function

$$C(i, m) = \begin{cases} |\mathbf{a}_i|_2 \cdot (1 - l_{id}) & \text{if } l_{id} > t \\ k \cdot |\mathbf{a}_i|_2 \cdot (1 - l_{id}) & \text{if } l_{id} < t \end{cases} \tag{18}$$

where $d = t_m$.

4.3.2 Modified DTW

For each element of a template sequence t_m should last a period of time, when given the optimal alignment of DTW matching, there should be at least K adjacent elements of mapped sequence $\mathbf{M}$ to be assigned to one element t_m in the template. So DTW is modified in the following way. First, each underlining state in template is repeated K times, for example, when K equals 2, the original sequence of

[1 2] extends to [1 1 2 2]; then the symmetric traditional DTW is modified into an *unsymmetric* one as follows

- Initialization
  $D(i, 1) = \sum_{k=1}^{n} C(x_k, y_1)$, where $i \leq N$;
  $D(1, m) = \sum_{k=1}^{m} C(x_1, y_k)$, where $m \leq KM$
- Modified DTW

$$D(i, m) = min \begin{cases} D(i-1, m-1) + C(i, m) \\ D(i-1, m) \quad\quad + C(i, m) \end{cases} \tag{19}$$

where $1 < i \leq N$,$1 < m \leq KM$;
N indicates the length of the mapped sequence;
M indicates the length of the original template,
thus KM indicates the length of extended template

4.3.3 Classification

After matching, we get the tamplate of each gesture C: $MDTW(\mathbf{L}, \mathbf{T}_c) = D(N, KM_c)$ for templates of each gesture c. Now the sequence should be recognized as the most similar gesture

$$C = arg \min_{c} MDTW(T_c, L) \tag{20}$$

where C is the recognition result.

5 Experiment

In this section, the environment and results of the experiment will be discussed in details.

5.1 Experiment description

5.1.1 Hardware and software environment

We built a real time recognition system to test the performance of our algorithm. As shown in Fig. 6, the system is mainly comprised of Wii remote, Bluetooth Module and Laptop.

1. Wii remote: collect the acceleration data.
   Wii remote [1] has a built-in three-axis accelerometer: ADXL330 from Analog Devices. When operating at 100 HZ, the accelerometer can sense acceleration between −3 g to 3 g with noise below 3.5 *mg*. The acceleration data and button action information can be sent from Wii Remote to Laptop via Bluetooth module. When collecting samples, the process of each gesture begins when the participant presses 'A' button on Wii remote, and it finishes when the 'A' button is released.
2. Bluetooth module: communicate between the wii remote and the laptop.
   IVT Bluesoleil is utilized to connect the laptop and Wii and the sample rate is 10 Hz.

Fig. 6 Real time gesture recognition system

3. Laptop: recognize the gesture in real time.
   The computer is a Thinkpad Laptop with Intel T5870 processer and 2 G memory. Our real time gesture recognition is implemented on the platform of Matlab. The Operation System is Windows XP and the Matlab edition is R2010a. To read the date from the Wii remote by bluetooth communication, we use the toolbox of WiiLab [4].

5.1.2 Samples description

The algorithm is tested on eight gestures arising from research undertaken by Nokia[18], shown in Fig. 5. We collect samples from 28 persons. 15 of them are male while others are female. All of them are graduate students and they have never used gesture recognition devices before. Each of the 8 gestures is repeated 10 times. Consequently we get the database of $28 person \times 8 gestures \times 10 times$, which aggregates totally 2,240 samples.

5.1.3 Parameters setting and experiment process

In our algorithm shown in Algorithm 1, we do not need any training samples. We use cosine metric to measure the distance between the acceleration vectors. The parameters are set as follows, $\epsilon = 0.001, k = 100, N = \lfloor \frac{1}{2} \frac{l}{l_{M_i}} \rfloor$. We get our recognition result although the real time gesture recognition system mentioned above.

We also implement HMM and DTW to conduct a comparison. When testing DTW, we take the first sample of each gesture collected by a person as the template. To see the influence of personality, we adopt the templates by different persons

Algorithm 1 MMDTW

Input: Z: Raw acceleration sequence;
Output: Gesture C

Step 1 **(Initialization)**
1.1 Generate gesture templates $\mathbf{T} = [t_1, \ldots, t_m \ldots, T_M]$ according to straight forward physical analysis in Section 4
Step 2 **(Hand Gesture Recognition)**
For each gesture acceleration sequence
2.1 Minus g in the vertical direction and get sequence $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_i, \ldots, \mathbf{a}_N)$.
2.2 Mapping the distance into probabilities with Softmax function to obtain a probabilistic underlining sequence $\mathbf{L}$ by sequence $\mathbf{A}$, where l_{id} indicates the probability of $\mathbf{a}_i$ to be underlining state $\mathbf{z}_d$ using Eq. 13.
2.3 Calculate the cost matrix $C(i, m)$ which indicates the distances between the ith element in the mapped sequence $\mathbf{L}$: $\mathbf{l}_i$ and the mth element in the template sequence $\mathbf{T}$: t_m by solving Eq. 18
2.4 Conduct Modified DTW Matching according to Eq. 19, $MDTW(\mathbf{L}, \mathbf{T}_c) = D(N, KM_c)$.
2.5 Classification by recognized the most similar gesture Input MMDTW Cost $MDTW(\mathbf{L}, \mathbf{T}_c)$, Output Classification Result C $C = arg\min_c MDTW(T_c, W)$
Step 3 **(Output Gesture)**
3.1 Output the gesture C

in turn, and classify all the other samples according to their similarity with the templates. In the experiment of HMM, since discrete HMM and continual HMM get similar performance [18], we utilize discrete HMM. We employ leave one out strategy and use all other persons' samples besides the testing person's as training samples. We use different percentage of training samples to see the recognition performance. The training samples of HMM are randomly selected from the data set and the parameters of HMM are determined according to [18] which sets 8 cluster classes and 5 latent states, In addition we use HMM toolbox by Murphy[6] to realize our left-to-right HMM test system.

5.2 Result

As displayed in Fig. 7, the comparison between DTW and MMDTW shows the result that the choice of template is closely related to the recognition result of DTW. With the best templates, DTW achieves an accuracy of 88.7 %; while the least accuracy of DTW sharply drops to 26.5 %. The average accuracy for DTW is 72.9 % and 9 out of 28 templates get an accuracy less than 70.0 %, which is intolerable in practical applications. In comparison, the accuracy of MMDTW is 93.1 % and is

[6] http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

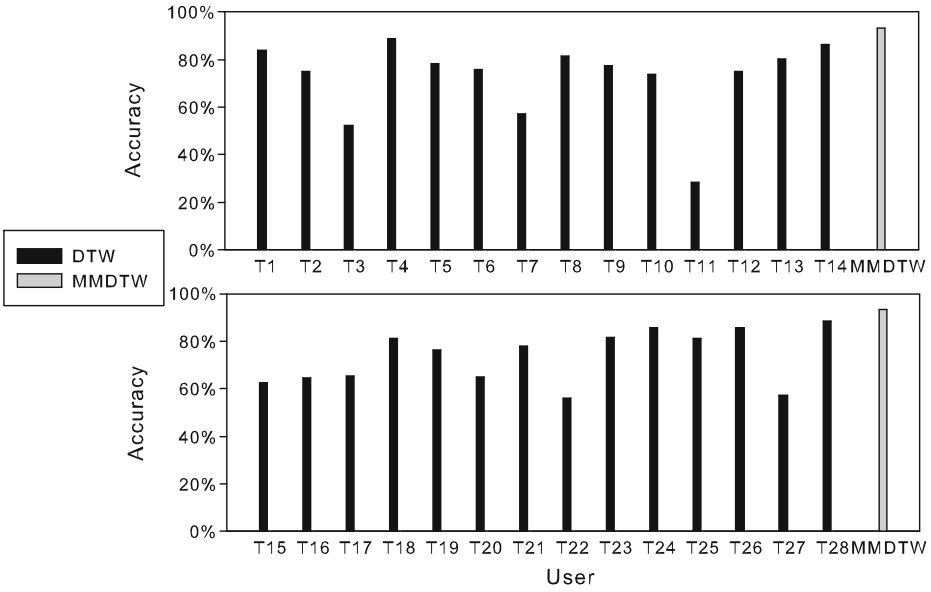


Fig. 7 Comparison of MMDTW and DTW. The accuracy of DTW changes when choosing different users' first sample of each gesture as template and all other users' samples as testing ones. MMDTW needs no sample from users to start and generates template of each gesture itself

better than DTW whichever the template is. This improvement just stems from the no-inclination templates and the matching in underlining space.

Then in Fig. 8, we can see the comparison of HMM and MMDTW. The result shows that MMDTW displays a much better performance than HMM when the latter uses relative small percentage of training samples. With increasing of training

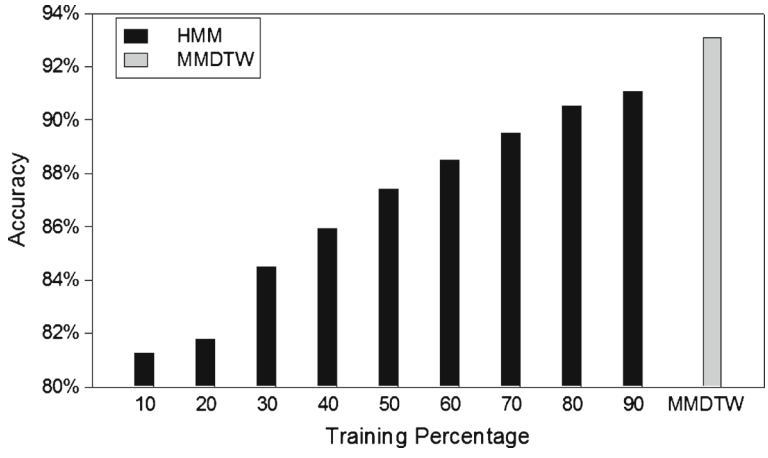


Fig. 8 Comparison of HMM and MMDTW. The accuracy of HMM changes when choosing different training samples percentage. MMDTW needs no training samples

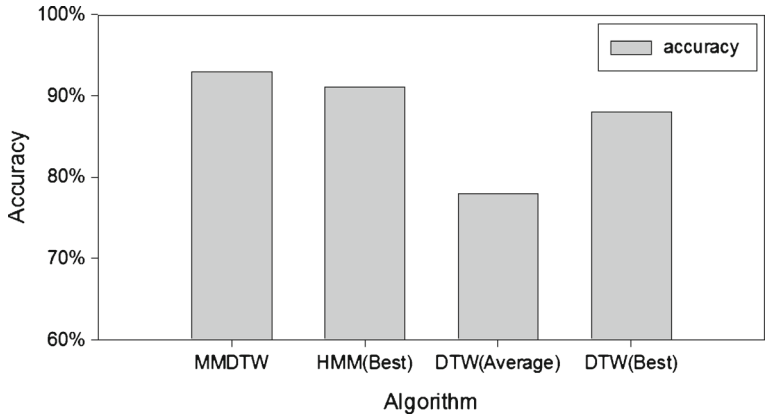


Fig. 9 Comparison of MMDTW, HMM and DTW

samples, HMM performs better. However, the accuracy of HMM can not exceed that of MMDTW even when 90 % samples are used for training. Compared with HMM, the improvement of MMDTW may come from the Modified DTW matching, which compares the differences between the template and the mapped sequence more efficiently.

Just we can conclude in Fig. 9, the best accuracy of HMM is 91.1 % and the average accuracy of DTW is 72.9 %. Thus, we can figure out that our algorithm can not only meet the training-free requirement, but also show a better accuracy than the training-required algorithms of DTW and HMM.

6 Conclusion

In this paper, in order to fulfill the training-free gesture recognition with accelerometer for user-independent applications, we make use of the crucial prior knowledge of gesture trail and design the algorithm of MMDTW. The algorithm defines an underlining space with the physical meaning of acceleration direction. After that, template for each gesture is defined in this space according to the gesture trail. Then acceleration sequence, the output of the accelerometer, is mapped into the underlining space to conduct a modified DTW matching with templates. Finally the best matched gesture turns out to be the recognition result. When tested in an user-independent experiment with 2,240 samples, the training-free algorithm even shows better performance than the traditional training-required algorithms of DTW and HMM, which verifies the effectiveness of MMDTW.

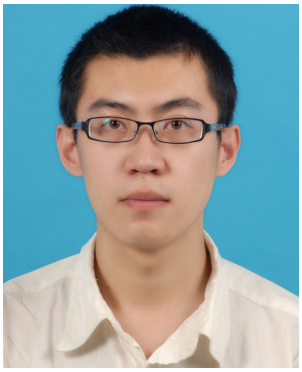
We believe our algorithm is the first step toward the training-free gesture recognition. In future work, by exploiting the gesture trail information, more complicated physical analysis can be adopted to generate the templates of general gestures with more general sensors, such as video camera. Also, without the time-consuming and laborious sample collection process, the well-performed training-free gesture recognition makes it much more convenient for users to extend the original gesture vocabulary themselves. Thus, this kind of training-free algorithms enjoys great

potential for a large number of applications, such as self-defined vocabulary based gesture recognition, gestures devices to support E-teaching, and so on.

References

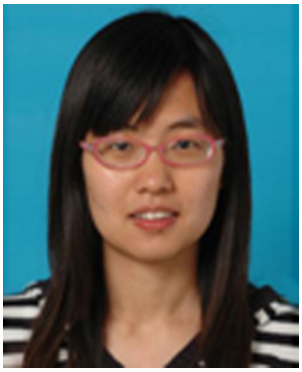
1. Adxl330 datasheet (2006)
2. Akl A, Valaee S (2010) Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In: 2010 IEEE international conference on acoustics speech and signal processing (ICASSP), IEEE, pp 2270–2273
3. Brezmes T, Gorricho JL, Cotrina J (2009) Activity recognition from accelerometer data on a mobile phone. In: Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living, pp 796–799
4. Brindza J, Szweda J, Liao Q, Jiang Y, Striegel A (2009) Wiilab: bringing together the nintendo wiimote and matlab. In: Frontiers in education conference, 2009. FIE'09. 39th IEEE. IEEE, pp 1–6
5. Byrne D, Doherty AR, Snoek CGM, Jones GJF, Smeaton AF (2010) Everyday concept detection in visual lifelogs: validation, relationships and trends. Multimed Tools Appl 49(1):119–144
6. Chen Y, Liu M, Liu J, Shen Z, Pan W (2011) Slideshow: Gesture-aware ppt presentation. In: 2011 IEEE international conference on multimedia and expo (ICME), IEEE, pp 1–4
7. Cho SJ, Oh JK, Bang WC, Chang W, Choi E, Jing Y, Cho J, Kim DY (2004) Magic wand: a hand-drawn gesture input device in 3-d space with inertial sensors
8. Choi ES, Bang WC, Cho SJ, Yang J, Kim DY, Kim SR (2005) Beatbox music phone: gesture-based interactive mobile phone using a tri-axis accelerometer. In: IEEE international conference on industrial technology, 2005. ICIT 2005. IEEE, pp 97–102
9. Farella E, Pieracci A, Benini L, Rocchi L, Acquaviva A (2008) Interfacing human and computer with wireless body area sensor networks: the wimoca solution. Multimed Tools Appl 38(3):337–363
10. Flórez F, García JM, García J, Hernández A (2002) Hand gesture recognition following the dynamics of a topology-preserving network. In: Fifth IEEE international conference on automatic face and gesture recognition, 2002. Proceedings. IEEE, pp 318–323
11. Holzinger A, Nischelwitzer AK, Kickmeier-Rust MD (2006) Pervasive e-education supports life long learning: some examples of x-media learning objects (xlo). Digital Media, pp 20–26
12. Holzinger A, Softic S, Stickel C, Ebner M, Debevc M (2009) Intuitive e-teaching by using combined hci devices: experiences with wiimote applications. In: Universal access in human-computer interaction. applications and services, pp 44–52
13. Holzinger A, Softic S, Stickel C, Ebner M, Debevc M, Hu B (2012) Nintendo wii remote controller in higher education: development and evaluation of a demonstrator kit for e-teaching. Comput Inform 29(4):601–615
14. Hürst W, van Wezel C (2012) Gesture-based interaction via finger tracking for mobile augmented reality. Multimed Tools Appl 62(1):233–258
15. Kettebekov S, Sharma R (2000) Understanding gestures in multimodal human computer interaction. Int J Artif Intell Tools 9(2):205–223
16. Lee HK, Kim JH (1999) An hmm-based threshold model approach for gesture recognition. IEEE Trans Pattern Anal Mach Intell 21(10):961–973
17. Liu J, Zhong L, Wickramasuriya J, Vasudevan V (2009) uwave: accelerometer-based personalized gesture recognition and its applications. Pervasive Mob Comput 5(6):657–675
18. Mäntyjärvi J, Kela J, Korpipää P, Kallio S (2004) Enabling fast and effortless customisation in accelerometer based gesture interaction. In: ACM international conference proceeding series
19. Mantyla VM, Mantyjarvi J, Seppanen T, Tuulari E (2000) Hand gesture recognition of a mobile device user. In: 2011 IEEE international conference on multimedia and expo (ICME), vol 1. IEEE, pp 281–284
20. Montoliu R, Blom J, Gatica-Perez D (2013) Discovering places of interest in everyday life from smartphone data. Multimed Tools Appl 62(1):179–307
21. Montoliu R, Gatica-Perez D (2010) Discovering human places of interest from multimodal mobile phone data. In: Proceedings of the 9th international conference on mobile and ubiquitous multimedia. ACM, p 12
22. Müller M (2007) Ltd MyiLibrary. Information retrieval for music and motion, vol 6. Springer Berlin

23. Park CB, Roh MC, Lee SW (2008) Real-time 3d pointing gesture recognition in mobile space. In: 8th IEEE international conference on automatic face & gesture recognition, 2008. FG'08. IEEE, pp 1–6
24. Pavlovic VI, Sharma R, Huang TS (1997) Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans Pattern Anal Mach Intell 19(7):677–695
25. Pei M, Jia Y, Zhu SC (2011) Parsing video events with goal inference and intent prediction. In: 2011 IEEE international conference on computer vision (ICCV), IEEE, pp 487–494
26. Peng X, Bennamoun M, Mian AS (2011) A training-free nose tip detection method from face range images. Pattern Recogn 44(3):544–558
27. Quintana GE, Sucar LE, Azcárate G, Leder R (2008) Qualification of arm gestures using hidden markov models. In: 8th IEEE international conference on automatic face & gesture recognition, 2008. FG'08. IEEE, pp 1–6
28. Rabiner LR (1989) A tutorial on hidden markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
29. Rabiner LR, Juang BH (1993) Fundamentals of speech recognition
30. Rajko S, Qian G (2008) Hmm parameter reduction for practical gesture recognition. In: 8th IEEE international conference on automatic face & gesture recognition, 2008. FG'08. IEEE, pp 1–6
31. Seo HJ, Milanfar P (2010) Training-free, generic object detection using locally adaptive regression kernels. IEEE Trans Pattern Anal Mach Intell 32(9):1688–1704
32. Sminchisescu C, Kanaujia A, Li Z, Metaxas D (2005) Conditional models for contextual human motion recognition. In: Tenth IEEE international conference on computer vision, 2005. ICCV 2005, vol 2. IEEE, pp 1808–1815
33. Song Y, Demirdjian D, Davis R (2011) Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In: 2011 IEEE international conference on automatic face & gesture recognition and workshops (FG 2011), IEEE, pp 388–393
34. Suk HI, Sin BK, Lee SW (2008) Recognizing hand gestures using dynamic bayesian network. In: 8th IEEE international conference on automatic face & gesture recognition, 2008. FG'08. IEEE, pp 1–6
35. Takahashi M, Fujii M, Naemura M, Satoh S (2013) Human gesture recognition system for tv viewing using time-of-flight camera. Multimed Tools Appl 62(3):761–783
36. Tsukada K, Yasamura M (2002) Ubi-finger: gesture input device for mobile use. In: Asia-Pacific computer and human interaction
37. Wang D, Xiong Z, Zhang M (2012) An application oriented and shape feature based multi-touch gesture description and recognition method. Multimed Tools Appl 58(3):497–519
38. Wilson A, Shafer S (2003) Xwand: Ui for intelligent spaces. In: Computer human interaction, pp 545–552
39. Wilson D, Wilson A (2004) Gesture recognition using the xwand
40. Wu J, Pan G, Zhang D, Qi G, Li S (2009) Gesture recognition with a 3-d accelerometer. In: Ubiquitous intelligence and computing, pp 25–38
41. Zhu Y, Xu G, Kriegman DJ (2002) A real-time approach to the spotting, representation, and recognition of hand gestures for human–computer interaction. Comput Vis Image Underst 85(3):189–208



Liang Yin is a Master student in Information and Telecommunication Engineering School at Beijing University of Posts and Telecommnications(BUPT) where he graduated with a Bachelor of Automation in the Fall 2010. He was the leader of a National Innovation Experiment Program for University Students. He had great performance during undergraduate study and was admitted by PRIS Lab in BUPT without the usually required Graduate Entrance Examination in China. Now he's a participant of a National Science Foundation Project.

Mingzhi Dong is a Master student in Information and Telecommunication Engineering School at Beijing University of Posts and Telecommnications(BUPT) where he graduated with a Bachelor of Automation in the Fall 2010. During his undergraduate years, Mingzhi was awarded First Level Scholarship in BUPT every year and was admitted by Pattern Recognition and Intelligent System lab in BUPT without the usually required Graduate Entrance Examination in China. Now he's a participant of a National Science Foundation Project.



Ying Duan is a Master student in Mathematics School at Beijing Normal University(BNU). She is a member of the Beijing LPS group which is a research team concentrating on mathematical education. She got her bachelor of Mathematics at Capital Normal University. She had great performance during undergraduate study and was admitted by BNU without the usually required Graduate Entrance Examination in China. Now she's a teaching assistant for the course of linear algebra.

Weihong Deng is an Associate Professor and Master supervisor at Beijing University of Posts and Telecommunications. He received his bachelor and PhD degree from Beijing University of Posts and Telecommunications in 1995 and 2010 respectively. He has published more than 20 papers on international journal and conferences including SCIENCE, IEEE TPAMI, CVPR, SIGIR. Now he is the responsible person for a National Science Foundation Project.

Kaili Zhao is a Phd student in Information and Telecommunication Engineering School at Beijing University of Posts and Telecommunications(BUPT). She got her bachelor of Automation at Hefei University of Technology(HFUT).

During her undergraduate years, Kaili has been awarded National Encouragement Scholarship and second level Scholarship each year in HFUT.

Jun Guo is a full Professor and PHD supervisor at Beijing University of Posts and Telecommunications. He's the Dean of School of Information and Communication Engineering in BUPT and the Director of Pattern Recognition and Intelligent System lab. He received his bachelor and Master degree from Beijing University of Posts and Telecommunications in 1982 and 1985 respectively, and in 1993 he received his PHD degree from Tohoku Gakuin University, Japan. His research interests are pattern recognition, machine learning, cross-media information retrieval, short message filtering, web public sentiment information analysis, network management and control. He is the responsible person for many projects funded by National 863 High-tech and National Natural Science Foundation of China, He published more than 100 papers on international journals and conferences, including SCIENCE, IEEE TPAMI, CVPR, SIGIR, and won many provincial honors, received 5 authorized patents.