Gesture Recognition with a 3-D Accelerometer

Jiahui Wu[1], Gang Pan[1], Daqing Zhang[2], Guande Qi[1], and Shijian Li[1]

[1] Department of Computer Science
Zhejiang University, Hangzhou, 310027, China
{cat_ng,gpan,shijianli}@zju.edu.cn
[2] Handicom Lab, Institut TELECOM SudParis, France
daqing.zhang@it-sudparis.eu

Abstract. Gesture-based interaction, as a natural way for human-computer interaction, has a wide range of applications in ubiquitous computing environment. This paper presents an acceleration-based gesture recognition approach, called *FDSVM (*Frame-based Descriptor and multi-class SVM), which needs only a wearable 3-dimensional accelerometer. With FDSVM, firstly, the acceleration data of a gesture is collected and represented by a frame-based descriptor, to extract the discriminative information. Then a SVM-based multi-class gesture classifier is built for recognition in the nonlinear gesture feature space. Extensive experimental results on a data set with 3360 gesture samples of 12 gestures over weeks demonstrate that the proposed FDSVM approach significantly outperforms other four methods: DTW, Naïve Bayes, C4.5 and HMM. In the user-dependent case, FDSVM achieves the recognition rate of 99.38% for the 4 direction gestures and 95.21% for all the 12 gestures. In the user-independent case, it obtains the recognition rate of 98.93% for 4 gestures and 89.29% for 12 gestures. Compared to other accelerometer-based gesture recognition approaches reported in literature FDSVM gives the best resulrs for both user-dependent and user-independent cases.

1 Introduction

As computation is getting to play an important role in enhancing the quality of life, more and more research has been directed towards natural human-computer interaction. In a smart environment, people usually hope to use the most natural and convenient ways to express their intentions and interact with the environment. Button pressing, often used in the remote control panel, provides the most traditional means of giving commands to household appliances. Such kind of operation, however, is not natural and sometimes even inconvenient, especially for elders or visually disabled people who are not able to distinguish the buttons on the device. In this regard, gesture-based interaction offers an alternative way in a smart environment.

Most of previous work on gesture recognition has been based on computer vision techniques [12]. However, the performance of such vision-based approaches depends strongly on the lighting condition and camera facing angles, which greatly restricts its applications in the smart environments. Suppose you are enjoying movies in your home theater with all the lights off. If you intend to change the volume of TV with gesture, it turns out to be rather difficult to accurately recognize the gesture under poor lighting condition using a camera based system. In addition, it is also uncomfortable

D. Zhang et al. (Eds.): UIC 2009, LNCS 5585, pp. 25–38, 2009.

and inconvenient if you are always required to face the camera directly to complete a gesture.

Gesture recognition from accelerometer data is an emerging technique for gesture-based interaction, which suits well the requirements in ubiquitous computing environments. With the rapid development of the MEMS (Micro Electrical Mechanical System) technology, people can wear/carry one or more accelerometer-equipped devices in daily life, for example, Apple iPhone [21], Nintendo Wiimote [22]. These wireless-enabled mobile/wearable devices provide new possibilities for interacting with a wide range of applications, such as home appliances, mixed reality, etc.

The first step of accelerometer-based gesture recognition system is to get the time series of a gesture motion. Previous studies have adopted specific devices to capture acceleration data of a gesture. For example, TUB-Sensor Glove [23] can collect hand orientation and acceleration, and finger joint angles. Tsukada [2] designed a finger-based gesture input device with IR transmitter, touch sensor, bend sensor and acceleration sensor. Mäntyjärvi [4] put a sensor box into a phone in order to detect 3-axis acceleration of users' hand motion. Now most accelerometers can capture three-axis acceleration data, i.e. 3D accelerometers, which convey more motion information than 2D accelerometers. They have been embedded into several commercial products such as iPhone [21] and Wiimote [22]. This paper employs Wiimote as the gesture input device for experimental set-up and performance evaluation.

To recognize a gesture from the captured data, researchers have applied diverse machine learning and pattern recognition techniques. The main algorithms in the literature are DTW (dynamic time warping)-based approach [17,18,20] and HMM (hidden Markov model) -based approach [1,5,6,7,24]. Both sets of algorithms process the acceleration data in the time domain. For example, in the seminal work of acceleration-based gesture recognition [7], the acceleration data is fed to HMM immediately after vector quantization. The DTW-based approach also exploited the acceleration data directly without doing feature extraction. Besides, Cho[16] utilizes Bayesian Networks using local maxima and minima as features to identify the majority of the selected gestures and further adopts binary-class SVM to discriminate the confusion gesture pair. Most of previous work on accelerometer-based gesture recognition performed matching or modeling in time domain, there is no feature extraction stage, which may result in sensitiveness of noise and variation of gesture data. We believe that a good feature extraction method can not only give a compact informative representation of a gesture, but also reduce the noise and variation of a gesture data to improve the recognition performance.

There are usually two ways to evaluate the performance of the gesture recognition algorithms: *user-dependent* and *user-independent*. Previous work focuses more on user-dependent gesture recognition [1,6,18], in which each user is required to perform a couple of gestures as training/template samples before using the system. One of the solutions to this problem is to reduce the user efforts in training by adding artificial noise to the original gesture data to augment the training data [5]. The user-independent gesture recognition does not require a user to enroll any gesture samples for the system before using it, where the model for classification has been well-trained before and the algorithm does not depend on users. The user-independent gesture recognition is more difficult than the user-dependent one since there is much more variation for each same gesture.

This paper addresses the gesture recognition problem using only one three-axis accelerometer. In order to reduce the effect of the intra-class variation and noise, we introduce a frame-based feature extraction stage to accelerometer-based gesture recognition. A gesture descriptor combining spectral features and temporal features is presented. The SVM-based gesture classification is proposed to deal with the highly nonlinear gesture space issue and the limited sample size problem. To evaluate our method, we conduct both the user-dependent experiments and the user-independent ones.

2 Frame-Based Gesture Descriptor

2.1 Frame-based Gesture Descriptor

An accelerometer can sense discretely the acceleration data of three spatial orthogonal axes in each gesture in a certain sampling frequency. We can denote a gesture as:

$$\boldsymbol{G} = (\boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z) \tag{1}$$

where $\boldsymbol{a}_T = (a_T^0, a_T^1, \ldots, a_T^{L-1})$, $T = x, y, z$, is the acceleration vector of an axis and L is the length of the temporal sequence.

To describe the whole gesture but distinguish the periods from each other, we divide a gesture into N+1 segments identical in length, and then every two adjunct segments make up a *frame* (cf. Fig. 1), which thus can be represented as $\boldsymbol{R}_k$ as follows:

$$\boldsymbol{R}_k = (\boldsymbol{r}_{x,k}, \boldsymbol{r}_{y,k}, \boldsymbol{r}_{z,k}), k = 0, \ldots, N-1, \tag{2}$$

$$\boldsymbol{r}_{T,k} = \left(r_{T,k}^0, r_{T,k}^1, \ldots, r_{T,k}^{L_S \cdot 2-1}\right), T = x, y, z, \tag{3}$$

$$r_{T,k}^n = a_T^{L_S \cdot k+n}, n = 0, \ldots, L_S \cdot 2 - 1, \tag{4}$$

where $L_s = \lfloor L/(N+1) \rfloor$ is the length of a segment. Each two adjunct frames have a segment-length overlap, as shown in Fig.1.

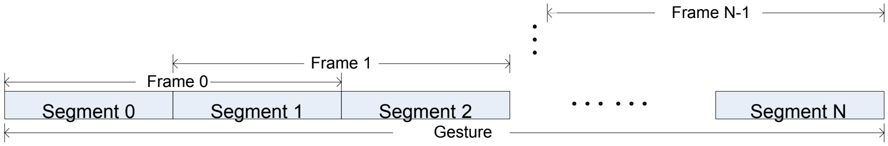

Fig. 1. Illustration of segments and frames for a gesture

Given a feature-type set $\mathscr{F}$ which describe the characteristics of a signal sequence, we can select a subset $F = \{f^{(i)}\}$, $i = 1, \ldots, n$, $f^{(i)} \in \mathscr{F}$ to describe one frame of a gesture based on a single axis. Since there are 3 axes and totally N frames per gesture, we combine all features to form a vector, whose dimension is d=3•n•N. More specifically, a gesture can be represented as:

$$\boldsymbol{\tau} = (f_{x,0}^{(1)}, \quad f_{y,0}^{(1)}, \quad f_{z,0}^{(1)}, \quad f_{x,0}^{(2)}, \quad f_{y,0}^{(2)}, \quad f_{z,0}^{(2)}, \dots, \quad f_{x,0}^{(n)}, f_{y,0}^{(n)}, \quad f_{z,0}^{(n)}, \\ \dots\dots \\ f_{x,N-1}^{(1)}, f_{y,N-1}^{(1)}, f_{z,N-1}^{(1)}, f_{x,N-1}^{(2)}, f_{y,N-1}^{(2)}, f_{z,N-1}^{(2)}, \dots, f_{x,N-1}^{(n)}, f_{y,N-1}^{(n)}, f_{z,N-1}^{(n)},). \tag{5}$$

Intuitively, more frames a gesture is broken up into, more details we know about the gesture. However, it may lead to the over-fitting problem if the frame number N becomes large. It will also increase the dimension of the feature space, which increases computational complexity. We will conduct an experiment to determine the optimal frame number N later.

2.2 Features for One Frame

According to the signal processing theory, features in the frequency domain have rich information about the signal. Hence not only the temporal features but also spectral features should be included in the feature-type set $\mathscr{F}$ to describe a single frame. In this paper, we make up the feature-type set $\mathscr{F}$ by combining the *mean* μ, *energy* ε and *entropy* δ in frequency domain and *standard deviation* σ of the amplitude and *correlation* γ among the axes in time-space domain, whose subsets will be utilized to describe a gesture:

$$\mathscr{F}=\{\mu, \varepsilon, \delta, \sigma, \gamma\}. \tag{6}$$

To obtain the features in frequency domain, we firstly employ a discrete Fourier transform (DFT) on each frame per axis to build a spectrum-based gesture representation:

$$t_{T,k}^{m} = \sum_{n=0}^{L_s\cdot 2-1} r_{T,k}^{n} e^{-\frac{2\pi i}{L_s\cdot 2-1}kn}, \\ T = x, y, z, \\ k = 0, \dots, N-1, \\ m = 0, \dots, L_s \cdot 2 - 1 \tag{7}$$

Afterwards, three spectral features are extracted from a frame: Mean μ , Energy ε and Entropy σ. Bao et al.[19] have successfully applied these three features in activity recognition. The mean feature is the DC component of the frequency domain over the frame:

$$\mu_{T,k} = t_{T,k}^{0} \tag{8}$$

The energy feature is the sum of all the squared DFT component magnitudes except the DC component of the signal, for the DC component has be used as an individual feature μ. The sum is divided by the number of the components for the purpose of normalization:

$$\varepsilon_{T,k} = \frac{\sum_{n=1}^{L_s \cdot 2-1} |t_{T,k}^{n}|^2}{|L_s \cdot 2 - 1|} \tag{9}$$

The entropy feature is the normalized information entropy of the DFT component magnitudes, where the DC component is also excluded:

$$\delta_{T,k} = \sum_{m=1}^{L_s \cdot 2-1} p_{T,k}^{m} \log(\frac{1}{p_{T,k}^{m}}) \tag{10}$$

$$p_{T,k}^{m} = \frac{|t_{T,k}^{m}|}{\sum_{n=1}^{L_s \cdot 2-1} |t_{T,k}^{n}|} \tag{11}$$

In the time domain, two features, i.e., standard deviation σ and correlation γ are obtained. Standard deviation of a gesture indicates the amplitude variability of a gesture, whose magnitude is equivalent with the other features compared with deviation:

$$\sigma_{T,k} = \sqrt{\sum_{n=0}^{L_s \cdot 2-1} (r_{T,k}^{n} - \bar{r}_{T,k}^{n})^2} \tag{12}$$

$$\bar{r}_{T,k}^{n} = \frac{r_{T,k}^{n}}{\sum_{n=0}^{L_s \cdot 2-1} r_{T,k}^{i}} \tag{13}$$

The correlation feature implies the strength of a linear relationship between two axis:

$$\gamma_{T_1 \sim T_2,\mathrm{k}} = \frac{v_{T_1 \sim T_2,k}^{n} - \bar{r}_{T_1,k}^{n} \cdot \bar{r}_{T_2,k}^{n}}{\sqrt{v_{T_1 \sim T_1,k}^{n} - (\bar{r}_{T_1,k}^{n})^2} \cdot \sqrt{v_{T_2 \sim T_2,k}^{n} - (\bar{r}_{T_2,k}^{n})^2}} \tag{14}$$

$$v_{T_1 \sim T_2,k}^{n} = \frac{\sum_{n=0}^{L_s \cdot 2-1} |r_{T_1,k}^{n} \cdot r_{T_2,k}^{n}|}{L_s \cdot 2} \tag{15}$$

$$T_1, T_2 = x, y, z$$

3 SVM-Based Gesture Classification

The Support Vector Machine (SVM) [11] is a small sample size method based on statistic learning theory. It is a new method to deal with the highly nonlinear classification and regression problems. SVM has been widely used in various applications, for example, gender classification [14], face detection [15], activity recognition [8].

When there are limited training data available, SVM usually outperforms the traditional parameter estimation methods which are based on Law of Large Numbers, since it benefits from structural risk minimization principle and avoidance of overfitting by its soft margin. SVM originally is used to solve the binary-class classification problems in the sense that it aims at finding the maximum-margin hyperplane that separates two classes of samples in the feature space.

Suppose there are two type of gestures *GTR*1, *GTR*2 needed to be classified. We denote the training set with n samples as

$$\{(\boldsymbol{\tau}_i, g_i)\},\ i=1,\ldots,n \tag{16}$$

where $\boldsymbol{\tau}_i \in R^d$represents a feature vector of a gesture like (5) and

$$g_i = \begin{cases} +1, & if\ \boldsymbol{\tau}_i\ belongs\ to\ GTR1 \\ -1\,, & if\ \boldsymbol{\tau}_i\ belongs\ to\ GTR2 \end{cases} \tag{17}$$

A separating plane can be written as

$$\boldsymbol{w} \cdot \boldsymbol{\tau} + b = 0. \tag{18}$$

The dual representation of margin maximum problem is:

$$\textbf{Maximize:} \qquad W(\alpha) = \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} g_i g_j \alpha_i \alpha_j K(\boldsymbol{\tau}_i, \boldsymbol{\tau}_j) \tag{19}$$

$$\textbf{s.t.} \qquad \textstyle\sum_{i=1}^{n} g_i \alpha_i = 0 \tag{20}$$

$$0 \le \alpha_i \le C \qquad i=1,\ldots,n \tag{21}$$

where α_i is the Lagrange factor and $K(\boldsymbol{\tau}_i, \boldsymbol{\tau}_j)$ is the kernel function [11]. Then the classification function would be:

$$f(x) = \mathrm{sgn}\big(h(x)\big), \tag{22}$$

$$h(x)=\textstyle\sum_{i=1}^{n} \alpha_i^* g_i K\big(\boldsymbol{\tau}_i, \boldsymbol{\tau}_j\big) + b^* \tag{23}$$

where α_i^*, $b^*$is the optimal solution of (19).

A gesture recognition system is supposed to recognize more than two types of gestures. For this reason, a multi-class SVM is required. The predominant approach is to convert a multi-class problem into several binary-class problems. Two typical strategies are available to build up a multi-class classifier: *one-versus-one* and *one-versus-all* (cf. Fig. 2). The former is to distinguish every pair of gestures and select the winner with more votes; the latter is to distinguish each type from the rest and select the one with largest output $|h(x)|$ in Equation (23). Figure 2 demonstrates the two strategies in classification of four types of gestures. For convenience of visualization, each gesture is represented by a 2D mean feature -- X-mean and Y-mean as shown in Equation (8).

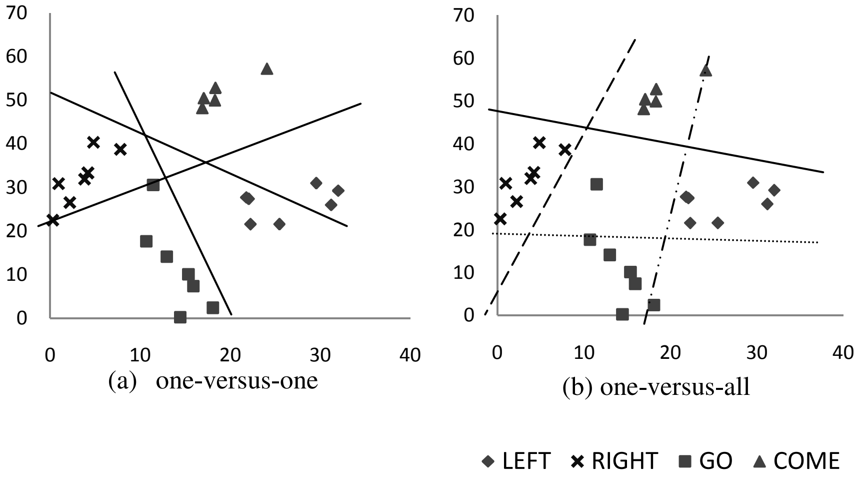

Fig. 2. Illustration of SVM classification of the four gestures LEFT, RIGHT, GO, COME. For visualization simplicity, each gesture is represented by a 2D mean feature. (a) illustrates the one-versus-one strategy by giving the separating lines between COME and one of the other three gestures. (b) shows the one-versus-all strategy by giving the separating lines between a gesture and the rest three ones.

4 Experiments

To evaluate the presented gesture recognition approach FDSVM, we implemented the gesture recognition system and collected a data set with 12 gestures of 10 individuals over two weeks. Three experiments were designed and reported in this section: the first one aimed at determining the optimal frame number N ; the second and the third ones were to evaluate the recognition performance for user-dependent and user-independent gesture recognition respectively.

4.1 Data Collection

To evaluate our system, we collected a gesture acceleration data set with 12 gestures of 10 individuals. We adopted *Wiimote*, the controller of Nintendo Wii equipped with a 3D accelerometer, to acquire gesture acceleration data. It can transmit users’ gesture motion acceleration data via Bluetooth. The accelerometer samples at a rate of nearly 100Hz. The start and end of a gesture were labeled by pressing the A button on the Wiimote during data acquisition. Figure 3 illustrates the acquisition devices.

In order to evaluate our gesture recognition algorithm, we choose three kinds of typical gestures: 4 direction gestures, 3 shape gestures, 5 one-stroke alphabet letters (the other 4 one-stroke letters, "O, L, U, M", are not included since 'O' is similar to CIRCLE, 'L' is similar to RIGHT-ANGLE, 'U' is similar to 'V', and 'W' is close to 'M'), as illustrated in Fig.4. These twelve gestures are divided into four groups (cf. Table 1) for evaluating the recognition performance. The first group is for direction gestures, the second for direction gestures plus shape gestures, the third is for one-stroke letters, and the last group is for all 12 gestures.

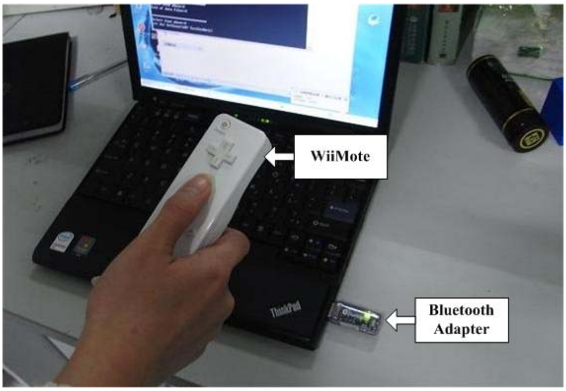

Fig. 3. Acquisition devices of gesture acceleration data

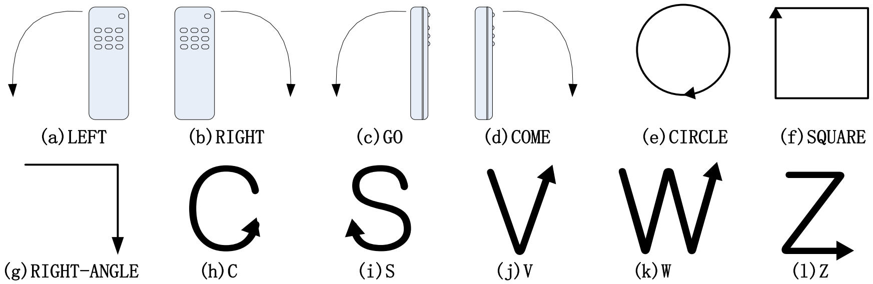

Fig. 4. Twelve gestures in the data set. (a) – (d) describe the gestures to swing the remote to left, right, forward or backward centered at the bottom point of the remote. (e) – (l) describe the gestures to draw a shape or letter.

Ten people participated in the data collection, including two female and eight male students aged between 21 and 25. Each student was asked to perform each gesture for 28 times over two weeks, namely 336 gesture samples in total. Every 12 different gestures performed sequentially was regarded as one set of gesture data. In order to consider variability of user performance, a participant was not allowed to perform more than two sets each time and not more than twice per day. Some acceleration data samples of the same gesture from a single participant are depicted in Fig.5, which obviously demonstrate the large variation of a gesture.

Table 1. Twelve gestures are divided into 4 groups for extensive evaluation

No.	Group Name	Included Gestures
1	Direction	LEFT, RIGHT, GO, COME
2	Direction+Shape	LEFT, RIGHT, GO, COME, CIRCLE, SQUARE, RIGHT-ANGLE
3	One-stroke Letter	C, S, V, W, Z, CIRCLE(O), RIGHT-ANGLE(L)
4	All	LEFT, RIGHT, GO, COME, CIRCLE, SQUARE, RIGHT-ANGLE, C, S, V, W, Z

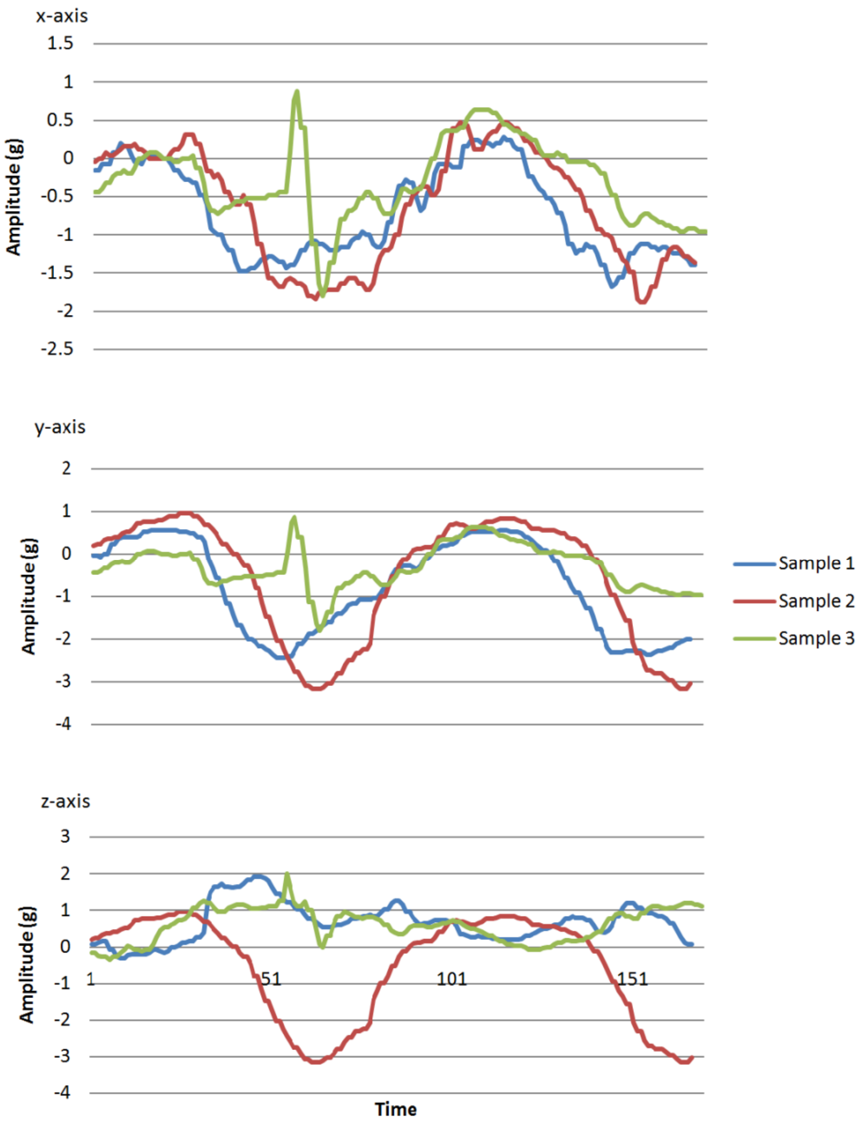

Fig. 5. Variation of a gesture for the same person . The three acceleration samples are from the gesture 'W' of the same person, shown in x-axis, y-axis, z-axis respectively. The three samples were performed at different time.

We employed the *4-fold cross validation* for the user-dependent case and the *leave-one-person-out cross validation* for the user-independent case. For the 4-fold cross validation, we divided 28 samples of the same gesture into four partitions, namely 7 samples per partition. At each time, one of the four partitions is for testing; the other three partitions are for training. We repeated it four times and finally took the average recognition rate. For the leave-one-person-out cross validation, the samples from one participant of ten are used as the testing data and the other nine participants' data as the training data.

4.2 Implementation of the System

The gesture recognition system has four main components: acceleration data acquisition, feature extraction, training phase, and recognition phase, as shown in Fig. 6. The *Acceleration Data Acquisition* is conducted with a Wiimote device, followed by the data transfer to a PC through Bluetooth. The continuous data streams are divided into individual gesture according to the A-button pressing label.

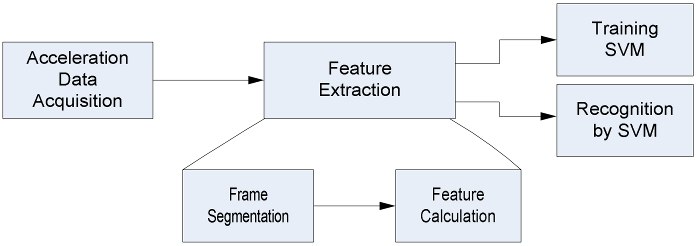

Fig. 6. Major components of the gesture recognition system

For the *feature extraction* of each gesture, firstly, it is divided into *N* frames (see Section 2). *Fourier transformation* is then employed on each frame, which uses an open source software package of FFTW [9]. Features *mean, energy, entropy, correlation,* and *standard deviation* of individual axis in one frame are calculated using equation (7)-(15), combined as a feature vector using (9).

The feature vector is eventually put into a classifier in order to train an SVM model or retrieve a recognized gesture type. The SVM component utilizes the open source package SVM$^{multi\text{-}class}$[10].

4.3 Experiment 1: Effect of Frame Number *N*

The purpose of analyzing a gesture in frames rather than in a whole is to describe its local characteristics corresponding to time span. The frame count *N* indicates the precision we know about a gesture. Yet increasing the number *N* results in higher computational complexity. This experiment is to examine the effect of varing *N*.

Figure 7 shows the experimental result for varying frame number *N* using the data set of Group 4. As it is shown, higher-rating occurs at the center in both curves, and lower-rating at both ends. The result supports our assumption that the feature will convey little discriminative information when *N* is too small, and the over-fitting problem will occur when N is too large. The recognition accuracy is obviously lower than the rest when *N* is 2. The two curves are nearly flat when *N* is between 5 and 11. In the following experiments, we will choose *N* = 9.

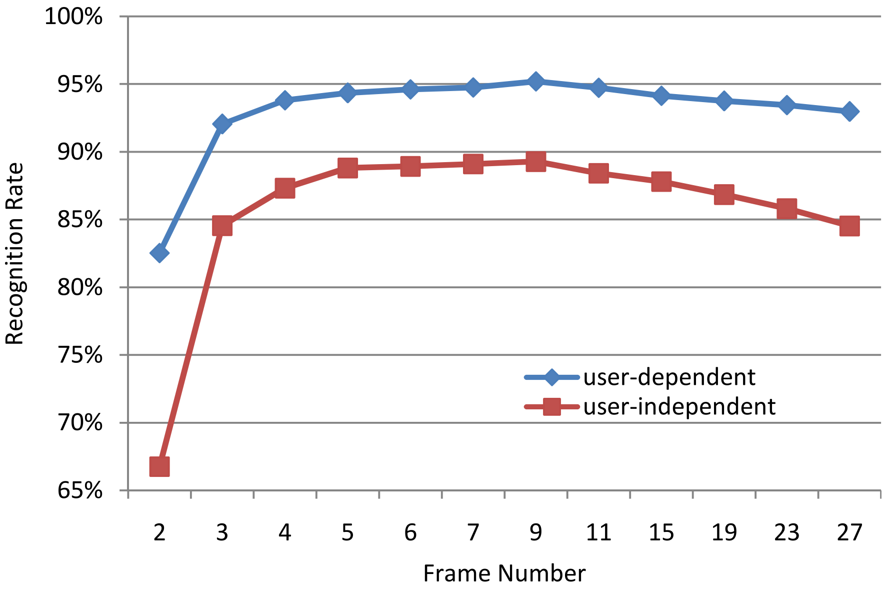

Fig. 7. The experimental result for various frame number *N*

4.4 Experiment 2: User-Dependent Gesture Recognition

In this experiment, to demonstrate the performance of our method, we compare it with four methods: decision tree C4.5, Naïve Bayes, DTW, and the HMM algorithm implemented by the package *WiiGee* [1] (which is an HMM-based method derived from [24]). We employed the implementation of C4.5 by Quinlan [13] and the WiiGee system developed by the authors of [1] for comparison purpose..

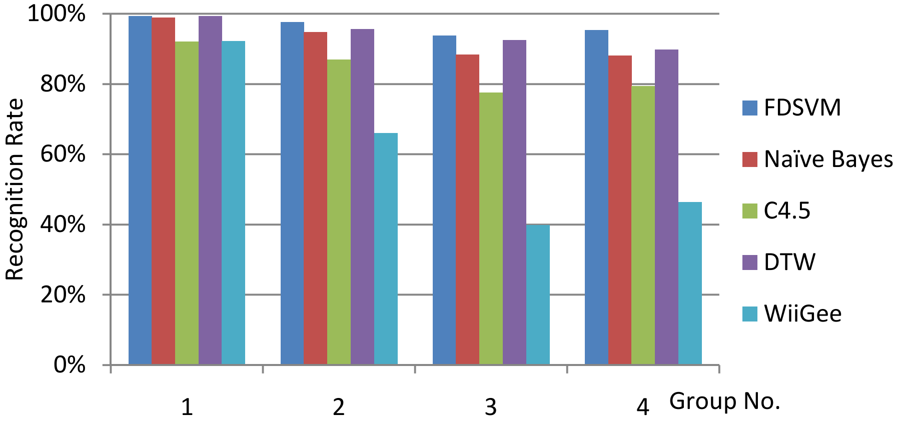

Fig. 8. The experimental results for the user-dependent case

We carried out the experiments and comparison tests on the four groups of data set respectively. The comparison results are shown in Fig.8. When recognizing the four gestures of Group 1, the recognition rate of the five approaches are all more than 90%, where our proposed FDSVM achieved 99.38%. From the figure, the performance of WiiGee decreases significantly when the number of gesture type increase. In contrast, our FDSVM method performs well even in recognizing all the 12 gestures, with the recognition rate of 95.21%. DTW is slightly better than Naïve Bayes while it is still outperformed by our FDSVM for Group tests 2/3/4.

4.5 Experiment 3: User-Independent Gesture Recognition

User-independent case means that the system is well-trained before users use it. Such implementation avoids users' efforts to perform several gestures as training data. The results of user-independent gesture recognition test and comparison are shown in Fig. 9. Obviously, the recognition rate of user-independent gesture recognition is lower than that of user-dependent one. Our FDSVM significantly outperforms the others. It obtains the recognition rate of 98.93% for 4 gestures of Group 1 and 89.29% for 12 gestures of Group 4. DTW and Naïve Bayes achieve recognition rate of 99.20% and 98.30% respectively for Group 1, very close to the performance of FDSVM. However, our FDSVM significantly outperforms DTW and Naïve Bayes for 7 gestures of Group 2, 7 gestures of Group 3, and 12 gestures of Group 4. The result reveals that our FDSVM has good generalization capability when the gesture type increases.

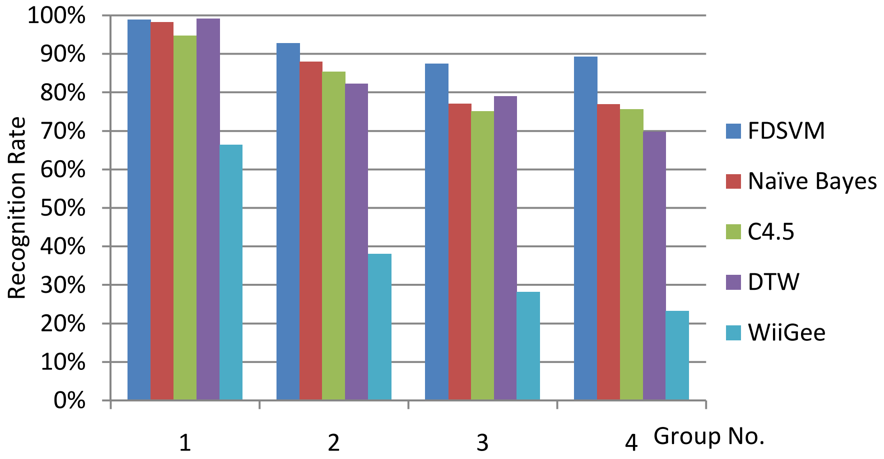

Fig. 9. The experimental result for the user-independent case

5 Conclusions

In this paper, an accelerometer-based gesture recognition method, called FDSVM, is presented and implemented. Different from the popular accelerometer-based gesture recognition approaches in the literature such as HMM and DTW, which don't include feature extraction explicitly this paper proposes a frame-based gesture descriptor for recognition, which combines spectral features and temporal features together. This explicit feature exaction stage effectively reduces the intra-class variation and noise of gestures. To tackle the issue of high nonlinearity of the gesture feature space, a multi-class SVM-based gesture classifier is built.

The extensive experiments on a data set with 3360 gesture samples of 12 gestures over time demonstrate that our approach FDSVM achieves the best recognition performance both in the user-dependent case and in the user-independent case, exceeding other four methods, DTW, HMM, Naïve Bayes and C4.5. In particular, the perfect user-independent performance of FDSVM on the large dataset, the high recognition rate of 98.93% for 4 gestures and 89.29% for 12 gestures hints us that the practical user-independent gesture recognition could be possible in the near future.

The future work is planned in three folds: new applications of the accelerometer-based gesture recognition, new approaches to improve the user-independent performance, and the continuous gestures recognition

Acknowledgements

This work is supported in part by the National High-Tech Research and Development (863) Program of China (No. 2008AA01Z132, 2009AA010000), the Natural Science Fund of China (No. 60525202, 60533040). The authors would thank anonymous reviewers for their valuable comments. The corresponding author is Dr. Gang Pan.

References

1. Schlömer, T., Poppinga, B., Henze, N., Boll, S.: Gesture Recognition with a Wii Controller. In: International Conference on Tangible and Embedded Interaction (TEI 2008), Bonn Germany, Feburary 18-20, pp. 11–14 (2008)
2. Tsukada, K., Yasumura, M.: Ubi-Finger: Gesture Input Device for Mobile Use. In: Proceedings of APCHI 2002, vol. 1, pp. 388–400 (2002)
3. Sawada, H., Hashimoto, S.: Gesture Recognition Using an Accelerometer Sensor and Its Application to Musical Performance Control. Electronics and Communications in Japan Part 3, 9–17 (2000)
4. Mäntylä, V.-M., Mäntyjärvi, J., Seppänen, T., Tuulari, E.: Hand Gesture Recognition of a Mobile Device User. In: Proceedings of the International IEEE Conference on Multimedia and Expo., pp. 281–284 (2000)
5. Mäntyjärvi, J., Kela, J., Korpipää, P., Kallio, S.: Enabling fast and effortless customization in accelerometer based gesture interaction. In: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia (MUM 2004), October 27-29, pp. 25–31. ACM Press, New York (2004)
6. Mäntylä, V.-M.: Discrete Hidden Markov Models with Application to Isolated User-Dependent Hand Gesture Recognition. VTT publications (2001)
7. Hofmann, F.G., Heyer, P., Hommel, G.: Velocity profile based recognition of dynamic gestures with discrete hidden markov models. In: Wachsmuth, I., Fröhlich, M. (eds.) GW 1997. LNCS, vol. 1371, pp. 81–95. Springer, Heidelberg (1998)

8. Ravi, N., Dandekar, N., Musore, P., Littman, M.: Activity Recognition from Accelerometer Data. In: Proceedings of IAAI 2008, July 2005, pp. 11–18 (2005)
9. Frigo, M., Johnson, S.G.: The Design and implementation of FFTW3. Proceedings of the IEEE 93(2) (2005)
10. Joachims, T.: Making large-Scale SVM Learning Practical. In: Schöllkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning. MIT-Press, Cambridge (1999)
11. Christanini, J., Taylor, J.S.: An Introduction to Support Vector Machines and other Kernel-based Methods. Cambridge University Press, Cambridge (2000)
12. Mitra, S., Acharya, T.: Gesture Recognition: A Survey. IEEE Trans. Systems, Man, and Cybernetics, Part C 37(3), 311–324 (2007)
13. Quinlan, J.R.: Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
14. Moghaddam, B., Yang, M.-H.: Learning Gender with Support Faces, IEEE Trans. Pattern Analysis and Machine Intelligence 24(5), 707–711 (2002)
15. Osuna, E., Freund, R., Girosi, F.: Training Support Vector Machines: An Application to Face Detection. In: Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition, pp. 130–136 (1997)
16. Cho, S.-J., Choi, E., Bang, W.-C., Yang, J., Sohn, J., Kim, D.Y., Lee, Y.-B., Kim, S.: Two-stage Recognition of Raw Acceleration Signals for 3D-Gesture-Understanding Cell Phones. In: 10th International Workshop on Frontiers in Handwriting Recognition (2006)
17. Niezen, G., Hancke, G.P.: Gesture recognition as ubiquitous input for mobile phones. In: International Workshop on Devices that Alter Perception (DAP 2008), conjunction with Ubicomp 2008 (2008)
18. Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J., Vasudevan, V.: uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications. In: IEEE PerCom 2009 (2009)
19. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
20. Wilson, D.H., Wilson, A.: Gesture Recognition using the XWand, Technical Report CMU-RI-TR-04-57, CMU Robotics Institute (2004)
21. Apple iPhone, `http://www.apple.com/iphone`
22. Nintendo Wii, `http://www.nintendo.com/wii`
23. Hommel, G., Hofmann, F.G., Henz, J.: The TU Berlin High-Precision Sensor Glove. In: Proceedings of the WWDU 1994, Fourth International Scientific Conference, vol. 2, pp. 47–49. University of Milan, Milan (1994)
24. Kela, J., Korpipaa, P., Mantyjarvi, J., Kallio, S., Savino, G., Jozzo, L., Marca, D.: Accelerometer-based gesture control for a design environment. Personal Ubiquitous Computing 10, 285–299 (2006)