Proceedings of the 2007 IEEE
International Conference on Robotics and Biomimetics
December 15 -18, 2007, Sanya, China

A Hybrid HMM/SVM Classifier for Motion Recognition Using μIMU Data*

Weiwei Wan and Hong Liu
State Key Laboratory of Machine Perception
Peking University
Shenzhen Graduate School
{weiweiwan, hongliu}@pku.edu.cn

Lianzhi Wang
Department of Computer Science
China Agricultural University
Beijing, China
ndjsj862@cau.edu.cn

Guangyi Shi and Wen J. Li
Center for Micro and Nano Systems
The Chinese University of Hong Kong
Hong Kong, China
{gyshi, wen}@mae.cuhk.edu.hk

***Abstract* - This paper describes a novel approach for human motion recognition via motion features extracted from sensor data. The classification process consists of two phases. The first one is a preprocessing of raw signals. Median Filter is used to filter pulse noise while Vector Quantization is used for Gaussian noise and reducing dimensions in this phase. The second one consists of a hybrid HMM/SVM classifier. Outputs from the first phase will be estimated by different pre-trained HMMs, and the results of the likelihood will be classified by the SVM classifier to identify the motion. With data collected from the μIMU equipment, falling-down motion can be told from non-falling-down motions with a correct recognition rate better than 99%. When the SVM training samples are labeled carefully and chosen bias, 100% correct recognition rate can be reached. The algorithm proves robustness and accuracy.**

Index Terms – Human motion recognition, HMM, SVM, μIMU.

I. INTRODUCTION

It is well known that the world is facing an increasingly aging population. With this increase, the proportion of frail and dependent elderly is also likely to increase significantly. This shift in demographic pattern will lead to an exponential increase in the number of elder individuals suffering from injury of falls, i.e., falls and fall-induced fractures are very common among the elderly.

Hip fractures account for most of the deaths and costs of all the fall-induced fractures. It can result in significant psychological trauma and lead to self-imposed restrictions of activity which can compromise the quality of life of the individual. Hip protectors are protective devices made of hard plastic or soft foam and are placed over the greater trochanter of each hip to absorb or shunt away the energy during mechanical impact on the greater trochanter. However, the compliance of the elderly to wear them is very low, due to discomfort, wearing difficulties and some other problems. Wen J. Li's group developed a novel hip protector with smaller dimensions and greater comfort for the elderly using their uIMU device based on MEMS sensors [1]. When the falling-down motion is detected by the MPU in the μIMU device, a compressed CO2 cartridges will be triggered to inflate the airbag to protect the elderly.

With three gyroscopes and two accelerators in the μIMU device, three dimensional angular rates and accelerations are measured, denoted by *Gx, Gy, Gz, Ax, Ay* and *Az*. Dimensions are defined as follows. Axis of *x* is from left hip to right hip, axis *y* is from back to the front and axis *z* is vertical to the ground. 200 sequences are collected and there are ten different motions, including 100 times of lateral falls which start with standing and fall in a vertical plane (30 fast, 30 slow and 40 normal) and 100 other normal motions including 10 times of jumping (high and low), 10 times of running, 20 times of sitting to chair (10 sitting down and 10 standing up), 10 times of squatting and 10 times standing up from squat, 20 times of walking including fast, normal and slow, 10 times of upstairs and 10 times of downstairs. With these data, a novel approach is tested, in which falling-down motion can be told from others accurately.

Human motion recognition has many potential applications in human computer interaction, human character animation and human activity recognition. Reference [2] gives a review of vision-based algorithms including a lot of traditional methods. After feature extraction, sensor-based recognition is almost the same as vision-based ones. Generally, these algorithms are divided into two different types. The first one takes motion signals static while the other one takes them time-varying. The first type involves division of feature space, such as linear or non-linear discriminating functions (SVM [3]). The second type makes classification according to time series, such as DTW [4] methods and HMM [5]. In recent years, more and more institutes begin to pay attention to multi-layer classifiers [6-7], and in this paper a two-layer classifier is realized.

HMM is chosen as we take motion sequences time-varying. However, it is of bad correct recognition rate [5]. The ambiguous results from HMM can be classified again to achieve better performance. Since the results are static, methods from the first type are under consideration. SVM is proved best in these classifications [3] since it is a realization of Statistical Learning Theory [3]. In this paper, both HMM and SVM are introduced to build the two-layer classifier.

First, data are collected from the μIMU device to train classifiers. Then experiments are carried out using the pre-trained classifiers. The whole classification process consists of

* This is a collaboration work between the Chinese University of Hong Kong and Peking University. For questions regarding recognition algorithms, please contact hongliu@pku.edu.cn, for questions regarding uIMU, please contact: wen@mae.cuhk.edu.hk

978-1-4244-1758-2/08/$25.00 © 2008 IEEE.

two phases. The first one is pre-processing of raw signals. Median Filter [8], which is widely used in image processing, is introduced to remove pulse noise. Since abrupt change will occur when one falls, Gaussian smoothing [8] is not adopted. Vector Quantization [9] is introduced for Gaussian noise and to reduce dimensions in this part. Outputs of the first phase are indices to the codebook, so enough information can be reserved. The second phase is a hybrid HMM/SVM classifier. Outputs from the first phase will be estimated by the 10 pre-trained HMMs, and the 10-dimensional likelihoods results will be classified by the SVM classifier to identify the motion.

This paper is organized as follows. HMM and SVM are introduced in Section II. In section III, the training algorithm and the recognition algorithm are given separately. Experiments are shown in section IV before the conclusion and future work which are presented in section V.

II. HMM AND SVM

As mentioned before, Median Filter, VQ, HMM and SVM are used in this paper. They are introduced briefly in this section.

A. Preprocessing Phase

1) Median Filter: Median Filter is a non-linear digital filtering technique which can protect the edge of signals when filtering. It is often used to remove noise from images or other signals. The idea is to examine a sample of the input and decide if it is representative of the signal. This is performed using a window consisting of an odd number of samples. The values in the window are sorted into numerical order; the median value, the sample in the centre of the window, is selected as the output. The oldest sample is discarded, a new sample acquired, and the calculation repeats.

Median filtering is a common step in image processing. It is particularly useful to reduce speckle noise and salt and pepper noise. Its edge-preserving nature makes it useful in cases where edge blurring is undesirable. This helps a lot in reducing noises caused by human movements when training samples are collected.

2) Splitting Method Based VQ: Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. It is a fixed-to-fixed length algorithm. The VQ design problem can be stated as follows. Given a vector source with its statistical properties known, a distortion measure and the number of code vectors, find a codebook (the set of all red stars) and a partition (the set of blue lines) which will result in the least average distortion. In 1980, Linde, Buzo, and Gray (LBG) proposed a VQ design algorithm based on a training sequence. The use of a training sequence bypasses the need for multi-dimensional integration. VQ designed using this algorithm is referred to in literatures as an LBG-VQ. LBG is used in this paper for quantizer design.

To convert a vector into a single dimensional parameter can both hold all the information and remove Gaussian noise. 64 vectors are produced after Vector Quantization. There are six elements in each vector, namely x_1, x_2, x_3, x_4, x_5 and x_6. Changes of indices to these vectors can be observed easily when sorted by the sum of all the six dimensions. The key problem of VQ is to choose the initial codebook. Splitting method helps to sample averagely, and features are distributed into the quantized vectors. By using splitting method, the system will be extensible and more kinds of motions can be introduced easily.

For i = 1, 2, ...,N, set

$$C_i^{(0)} = (1+\varepsilon)C_i^* \tag{1a}$$

$$C_i^{(1)} = (1-\varepsilon)C_i^* \tag{1b}$$

C_i denotes the initial centroid, and it splits into two parts with a small ε each time. After 6 splits, 64 vectors are generated. The reason of choosing 64 is that with a longer codebook, there will be more noises, while with a shorter codebook, there will be too much loss.

B. Classification Phase

1) HMMs Based On Indices: Indices of sorted code vectors are used to train HMMs. Similar to speech recognition with paragraphs, initial point and end point must be found, namely endpoint detection. In speech recognition, this problem can be solved using breaks between words, energy or frequency; however, the same methods are not fit to classification of motions. A man who is claiming the stairs may fall down. A sliding window method is introduced to avoid these confusion, and the result shows high accuracy.

Baum-Welch algorithm [5] is used to train HMMs, while forward and backward algorithm is used to supervise the testing sequences.

2) SVM: The Support Vector Machine is a new technique in the field of statistical learning theory [10]. Originally, SVM was developed for classification problems. It was then extended to regression estimation problems, i.e., to problems related to finding the function: $y = f(x)$, $y \in R$, $x \in R^N$, given by its measurements y_i with noise at some (usually random) vector x_i, (y_1, x_1), …, (y_l, x_l). In SVM, the basic idea is to map the data X into a high-dimensional feature space f via a nonlinear mapping Φ, and to do linear regression in this space [4].

$$f(x) = (\omega \cdot \Phi(x)) + b, (\Phi : R^N \to F, \omega \in F) \tag{2}$$

where b is a threshold. Thus, linear regression in a high dimensional (feature) space corresponds to nonlinear regression in the low dimensional input space R^N. Note that the dot product in formula (2) between ω and $\Phi(x)$ would have to be computed in this high dimensional space which is computationally expensive. A more efficient technique is to use a kernel that eventually leaves us with dot products that can be implicitly expressed in the low dimensional input space R^N. Since Φ is fixed, we determine ω from the data by minimizing the sum of the empirical risk $R_{emp}[f]$ and a complexity term ||ω||2, which enforces flatness in feature space:

$$R_{reg}[f] = R_{emp}[f] + \lambda\|\omega\|^2 = \sum_{i=1}^{l} C(f(x_i) \cdot y_i) + \lambda\|\omega\|^2 \tag{3}$$

where l denotes the sample size $(x_1, \ldots, x_l)$, $C(.)$ is a loss function and λ is a regularization constant. For a large set of loss functions, formula (3) can be minimized by solving a quadratic programming problem, which has a unique solution.

It can be shown that the vector ω can be written in terms of the data points:

$$\omega = \sum_{i=1}^{l} (a_i - a_i^*)\Phi(x_i) \tag{4}$$

with α_i, a_i^* being the solution of the aforementioned quadratic programming problem. α_i and α_i^* have an intuitive interpretation as forces pushing and pulling the estimate $f(x\mathrm{i})$ towards the measurements y_i. Taking formula (2) and (4) into account, the whole problem can be rewritten in terms of dot products in the low dimensional input space

$$\begin{aligned} f(x) &= \sum_{i=1}^{l} (a_i - a_i^*)(\Phi(x_i) \cdot \Phi(x)) + b \\ &= \sum_{i=1}^{l} (a_i - a_i^*) K(x_i, x) + b \end{aligned} \tag{5}$$

where α_i, α_i^* are Lagrangian multipliers, and x_i are support vectors.

III. Two-layer Classifier

With the mathematical models introduced in section II, a two-layer classifier is realized, which is a composition of HMM and SVM.

A. Training Algorithm

1) Hidden Markov chains

Before training, raw signals from sensors should be preprocessed, including scaling and filtering. Training samples for HMM are generated after this process. Then, Baum-Welch algorithm is introduced for training of the Hidden Markov chains.

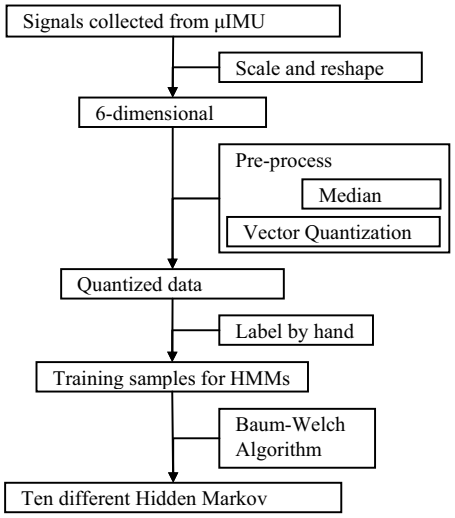


Fig.1 The process of training HMM

Given the training samples, Hidden Markov chains are generated as below.

Step 1-1. Samples are used to generate 64-codebook *C* with LBG-algorithm.

Step 1-2. Each sample is quantized according to *C*, and the results are sequences with index numbers pointing to items in *C*.

Step 1-3. S_0, S_1, S_2 ... are introduced to denote these sequences. 50-frame motion clips are cut from each *Si*, with labeled starting and ending points.

Step 1-4. Train the corresponding 10 Hidden Makov chains with these 50-frames until the logarithm likelihood between two succeeding iterations is less than 0.01.

Step 1-5. End.

Fig.2 Algorithm for HMM training

2) Support Vector Machine Classifier

The pre-trained Hidden Markov chains are used in SVM classifier training. For a specific sample, its corresponding motion is known. With 10 different Hidden Markov chains generated from the last process, 10 logarithm likelihoods will be produced. They can be recognized as a point in a 10-dimensional feature space. To train a SVM classifier is to find an optimal hyper plane in such a space with points (samples) belonging to different classes known. To fix the idea, refer to Section 4.

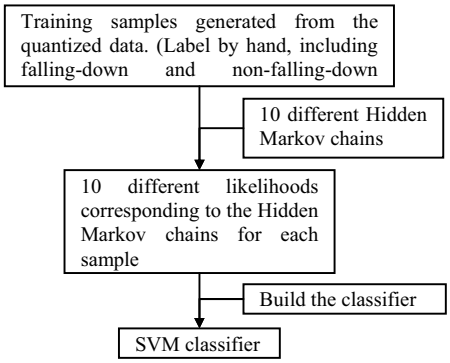


Fig.3 The process of training SVM classifier

B. Recognition Algorithm

The whole classification process consists of two phases. The first one is pre-processing of raw signals while the second phase is a hybrid HMM/SVM classifier.

1) Pre-processing and Sliding Window Method

As done in the training process, before classification, raw signals from sensors should be preprocessed first, including scaling and filtering.

Since real-time motion recognition should be carried out with a man wearing the μIMU device, starting points of motions cannot be labeled by hand. Sliding window method is introduced to solve this problem, as follows.

Step 2-1. Scale the raw signals.

Step 2-2. Specify a window with length *w*, and set the counter count *w*.

Step 2-3. Starting from the count frame, the previous w frames are cut off.

Step 2-4. Filter the w frames with a Median Filter to reduce pulse noise and quantize them with the pre-trained codebook *C*.

Step 2-5. Make classification with the hybrid HMM/SVM classifier and predict the motion.

Step 2-6. Set *count* = *count* + 1, and go to Step 2-2.

Step 2-7. End.

Fig.4 Recognition Algorithm

Step 2-2, 2-3 and 2-6 in Fig.4 composes the concept of the sliding window method.

2) Hybrid HMM/SVM Classification

Details of Step 2-5 in Fig.4 are shown in the following figure.

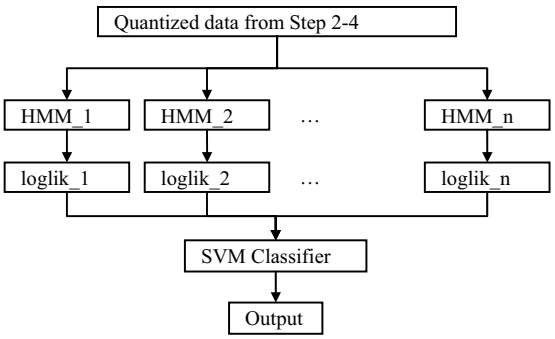


Fig.5 Motion classification using the hybrid model

With the quantized output window data of step 2-4, 10 logarithm likelihoods to 10 different Hidden Markov chains are calculated first. These logarithm likelihoods are taken as a point in a 10-dimensional feature space, and the process of SVM classification is to find the space (class) that the point belongs to using the pre-trained hyperplane (SVM classifier). Output in Fig.5 shows the kind of the motion.

IV. Experiments

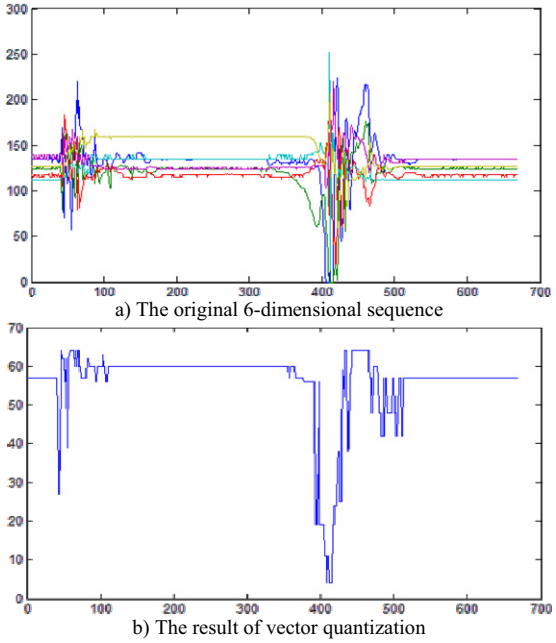
Experiments are carried out according to the algorithm introduced previously, and they will be introduced in this section. The experimental environment consists of an Intel Centrino 1.5GHz computer with 512MB memory and a MATLAB software platform.

10 different motions (200 data sequences, refer to section 1) are collected from the μIMU device, and 10 different Hidden Markov chains are trained corresponding to these motions. Then 100 falling-down motions and 150 non-falling-down motions are labeled manually to train the SVM classifier, as follows.

A. Median Filter and Vector Quantization

Median Filter is introduced to reduce pulse noise, and 7 is chosen as the length of the filter window. Gaussian noise can be removed by Vector Quantization.

After trying codebooks of length 32, 64 and 128, we found 64 is the best in our application. Enough information can be preserved as well as reducing most of the Gaussian noise. When the codebook *C* is generated using LBG-algorithm, quantization can be performed according to it. Code vectors are sorted by the sum of different dimensions to show the trends of independent motions. The results of quantization and the position where a motion happens can be observed easily after this sort. Fig.6 shows the result of quantization of the first sequence compared with the original 6-dimensional sequence.



a) The original 6-dimensional sequence

b) The result of vector quantization

Fig.6 Original data and quantized data

The curves of color blue, green, red, cyan, purple and gold in Fig.6 a) show the changes in value *Gx, Gy, Gz, Ax, Ay* and *Az*. And in Fig.6 b) a sequence of scalars are shown. It is the result of the quantized 6-dimensional sequence.

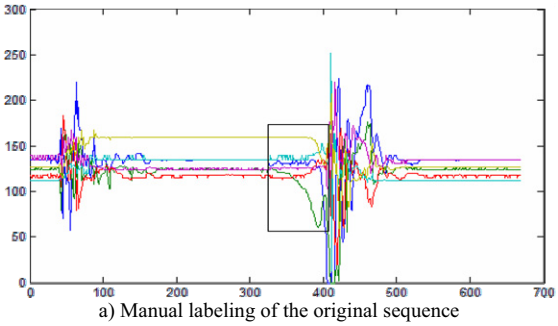
The same codebook *C*, which is trained in the preprocessing phase, is used in the motion classification phase.

B. Labeling By Hand

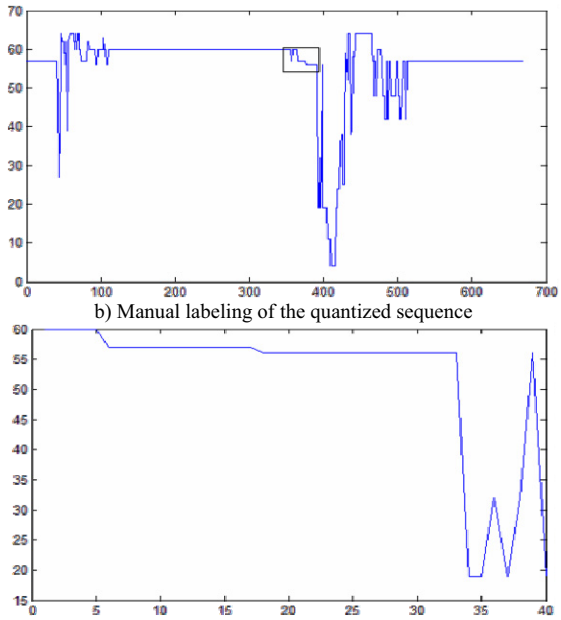
Endpoint labeling is a labor-intensive and error-prone process and only feasible for a limited set of motion data. It costs a lot of time and will not be easily done automatically by computers. In this way, the correctness depends a lot on the man who labels. Two different labeling of the 200 sequences are performed to show the robust of our algorithm in which different window length is adopted.

50 is set to *w* in the first labeling while 40 in the second time. Fig.7 illustrates a 40-length labeling.

After these manual labeling, 200 training samples for Hidden Markov chains are generated.



a) Manual labeling of the original sequence



b) Manual labeling of the quantized sequence

c) Result of labeling

Fig.7 Labeling by hand

C. *Classification with Hidden Markov Models*

Correct recognition rate can be improved greatly when the SVM layer is added. However, experiments are also carried out with only a single HMM layer for comparison. With the two different labeling results, recognition rates are only 61% and 69% respectively. Single layer classification with only HMM is of bad recognition rate in our experiments.

When estimated with the 10 different models, ambiguous results are usually achieved. Fig.8 shows the result of a falling-down sequence recognition. The whole process is the same as the algorithm introduced in Section III without the second SVM layer.

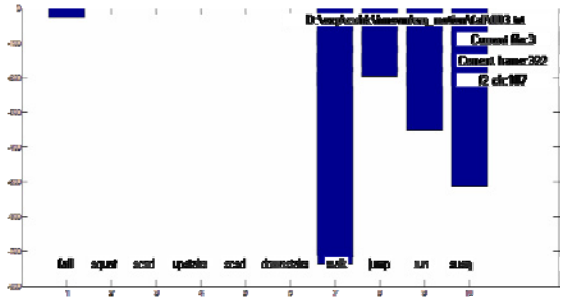


Fig.8 Ambiguous result of single HMM classifier

It is the 3rd sequence of the falling-down motion data in Fig.8. Sliding window is used to cut off every 40 or 50 frames to make classification. As shown in the figure, frame 322 is where the falling-down motion happens, but single layer HMM classifier found it at frame 107. Ambiguity happens at this frame. It is recognized as fall, walk, jump, run and stand up from squat respectively. We can find the logarithm likelihoods differ from motion to motion although any of them is the possible kind. By adjusting some parameters to construct a simple second criterion will do great help in improving the recognition rate. After parameter adjusting, the correct recognition rate can be improved to 100% [11]. Unfortunately, these empirical adjustments have no mathematical background. If not changed, the same criterion will do poorly in the second experiment, only a correct recognition rate of 93% is achieved. In fact, with different labeling, different parameters should be used. This is one motivation of our SVM layer.

D. *Hybrid HMM/SVM Classification*

SVM is introduced as the second layer to make further classification of the ambiguous results from the HMM layer. The recognition rates are satisfying, as follows.

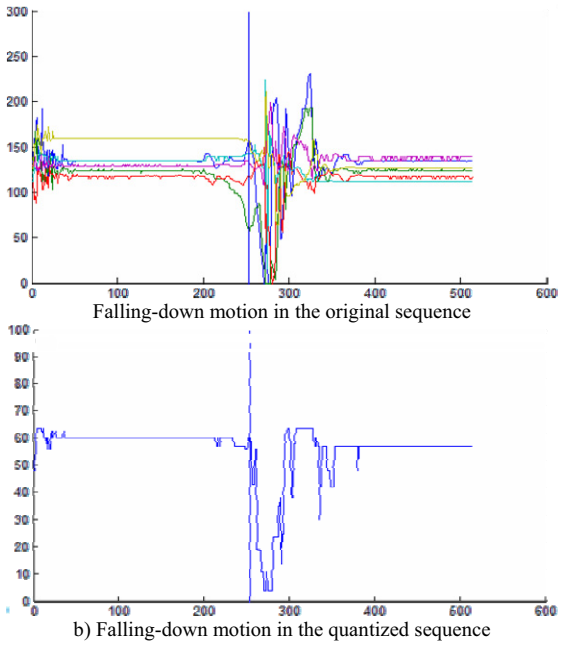
TABLE I

CORRECT RECOGNITION RATE WITH DIFFERENT METHODS

Count	HMM Classification 1	HMM Classification 2	Hybrid HMM/SVM Classification
Exp_1	69%	100%	100%
Exp_2	61%	93%	99%

*HMM Classification 2 adjusted the criterion empirically. Both experiments took the same criterion.

The simulator can detect falling-down motion before a man hits to the ground (when the gold curve in Fig.9 suddenly spikes), so enough time can be preserved for the DSP to trigger the airbag.



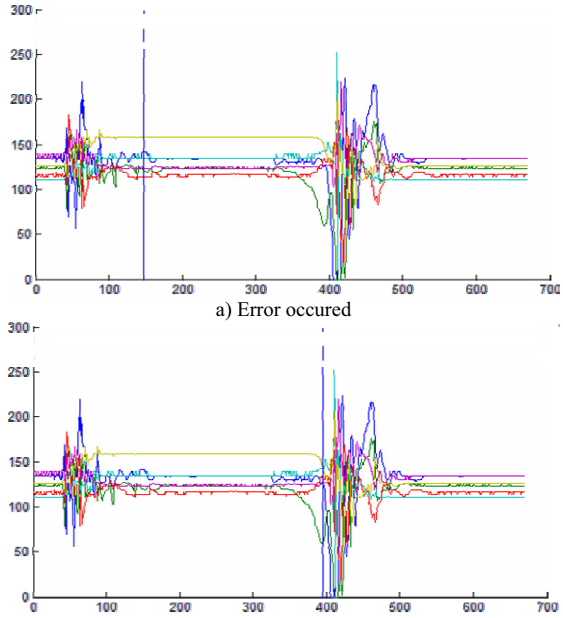
Falling-down motion in the original sequence

b) Falling-down motion in the quantized sequence

Fig.9 Hybrid HMM/SVM classification

Experiments are also carried out to find out the reason for the error in the second hybrid classification experiment. The error point (the first sequence) is at the edge of the two-class spaces, that is, it is between the support vectors. 50 more training samples, which are produced by hand, are added to

generate a better hyperplane. These manual samples are all between the support vectors to avoid this kind of confusion. 100% recognition rate can be achieved with the new SVM classifier trained by these 100 falling-down points and 200 non-falling-down points.



a) Error occured

b) Error fixed after modification

Fig.10 Make modification to the SVM classifier

V. CONCLUSIONS

This paper presented a novel method for human motion recognition using μIMU data. Hybrid HMM/SVM methods are introduced, where SVM is used to tell the ambiguous results generated from the HMM classification. Experiments are carried out with data collected from the equipment. With two groups of experiments, the hybrid model is proved robust and accurate. Falling-down and other motions can be separated with a 100% correct recognition rate with different manual labels. Future work consists of quantizing the data according to certain features and labeling them automatically. We are also looking forward to realizing the classifier on the μIMU device.

ACKNOWLEDGMENT

This project is funded by the Hong Kong Innovation and Technology Commission (Project ITF-GHP-029-06) and National Natural Science Foundation of China (NSFC 60675025) and the National High Technology Research and Development Program of China (863 Program, No.2006AA04Z247).

REFERENCES

[1] Guangyi Shi, Cheung-Shing Chan, Yilun Luo, Guanglie Zhang, Wen J. Li, Philip H.W. Leong and Kwok-Sui Leung, *Development of Human Airbag System for Falling Protection Using MEMS Motion Sensing Technology*. in Proceeding of Intelligent Robot and System 2006 IEEE (IROS'06), pp 4405-4410, 2006

[2] Ying Wu and Thomas S. Huang, *Vision-based gesture recognition: A Review*. in Proceedings of the International Gesture Recognition Workshop, pp. 103--115, 1999

[3] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[4] Eammon J. Keogh and Michael J.Pazzeni, *Derivative Dynamic Time Warping*, in First SIAM International Conference on Data Mining (*SDM'01*), 2001

[5] Lawrence R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, in Proceedings of the IEEE, pp 257-286, Vol.77, No.2, 1989

[6] Jianjun Ye, Hongxun Yao, Feng Jiang, *Based on HMM and SVM Multilayer Architecture Classifier for Chinese Sign Language Recognition with Large Vocabulary*, in Proceedings of the Third International Conference on Image and Graphics (*ICIG'04*), pp 377-380, 2004

[7] Aravind Ganapathiraju, Joseph Picone, *Hybrid SVM/HMM architecture for speech recognition*, in Proceedings of the International Conference on Spoken Language Process (*ICSLP'2000*), pp 504-507, 2000

[8] Fengjun Lv, *Introduction to Image Processing, Algorithms and Programming*, Tsinghua University Press, ISBN 7-302-03523-7/TP.1928, 1999

[9] R. M. Gray, *Vector Quantization*, IEEE ASSP Magazine, pp 4-29, 1984

[10] V. N. Vapnik, *Statistical Learning Theory: Inference from Small Samples*, New York: Wiley, 1998

[11] Weiwei Wan, Guangyi Shi, Hong Liu, Wen J. Li, *Real-time Recognition of Multi-category Human Motion Using uIMU Data*, in Proceedings of the International Conference on Mechatronics and Automation (*ICMA'07*)