MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition

Ruize Xu, Shengli Zhou, and Wen J. Li, *Fellow, IEEE*

***Abstract*—This paper presents three different gesture recognition models which are capable of recognizing seven hand gestures, i.e., *up*, *down*, *left*, *right*, *tick*, *circle*, and *cross*, based on the input signals from MEMS 3-axes accelerometers. The accelerations of a hand in motion in three perpendicular directions are detected by three accelerometers respectively and transmitted to a PC via Bluetooth wireless protocol. An automatic gesture segmentation algorithm is developed to identify individual gestures in a sequence. To compress data and to minimize the influence of variations resulted from gestures made by different users, a basic feature based on sign sequence of gesture acceleration is extracted. This method reduces hundreds of data values of a single gesture to a gesture code of 8 numbers. Finally, the gesture is recognized by comparing the gesture code with the stored templates. Results based on 72 experiments, each containing a sequence of hand gestures (totaling 628 gestures), show that the best of the three models discussed in this paper achieves an overall recognition accuracy of 95.6%, with the correct recognition accuracy of each gesture ranging from 91% to 100%. We conclude that a recognition algorithm based on *sign sequence and template matching* as presented in this paper can be used for nonspecific-users hand-gesture recognition without the time consuming user-training process prior to gesture recognition.**

***Index Terms*—Gesture recognition, interactive controller, MEMS accelerometer.**

I. INTRODUCTION

THE increase in human-machine interactions in our daily lives has made user interface technology progressively more important. Physical gestures as intuitive expressions will greatly ease the interaction process and enable humans to more naturally command computers or machines. For example, in telerobotics, slave robots have been demonstrated to follow the master's hand motions remotely [1]. Other proposed applications of recognizing hand gestures include character-recognition in 3-D space using inertial sensors [2], [3], gesture recognition to control a television set remotely [4], enabling a hand as a 3-D mouse [5], and using hand gestures as a control mechanism in virtual reality [6]. Moreover, gesture recognition has also been proposed to understand the actions of a musical conductor [7]. In our work, a miniature MEMS accelerometer based recognition system which can recognize seven hand gestures in 3-D space is built. The system has potential uses such as a remote controller for visual and audio equipment, or as a control mechanism to command machines and intelligent systems in offices and factories.

Many kinds of existing devices can capture gestures, such as a "Wiimote," joystick, trackball and touch tablet. Some of them can also be employed to provide input to a gesture recognizer. But sometimes, the technology employed for capturing gestures can be relatively expensive, such as a vision system or a data glove [8]. To strike a balance between accuracy of collected data and cost of devices, a Micro Inertial Measurement Unit (μIMU) is utilized in this project to detect the accelerations of hand motions in three dimensions.

There are mainly two existing types of gesture recognition methods, i.e., *vision-based* and *accelerometer and/or gyroscope* based. Due to the limitations such as unexpected ambient optical noise, slower dynamic response, and relatively large data collections/processing of vision-based method [9], our recognition system is implemented based on an inertial measurementunit based on MEMS acceleration sensors. Since heavy computation burden will be brought if gyroscopes are used for inertial measurement [10], our current system is based on MEMS accelerometers only and gyroscopes are not implemented for motion sensing.

Existing gesture recognition approaches include template-matching [11], dictionary lookup [12], statistical matching [13], linguistic matching [14], and neural network [15]. For sequential data such as measurement of time series and acoustic features at successive time frames used for speech recognition, HMM (Hidden Markov Model) is one of the most important models [16].It is effective for recognizingpatterns with spatial and temporal variation [17]. In this paper, we present three different gesture recognition models, which are: 1) sign sequence and Hopfield based gesture recognition model; 2) velocity increment based gesture recognition model; and 3) sign sequence and template matching based gesture recognition model. In these three models, in order to find a simple and efficient solution to the hand gesture recognition problem based on MEMS accelerometers, the acceleration patterns are not mapped into velocity, displacement or transformed into frequency domain, but are directly segmented and recognized in time domain. By extracting a simple feature based on sign sequence of acceleration, the recognition system achieves high accuracy and efficiency without the employment of HMM.

II. GESTURE MOTION ANALYSIS

Gesture motions are in the vertical plane (as defined by the x-z plane in Fig. 1(a)) or the projection of the motions is mainly

Manuscript received July 07, 2011; accepted August 17, 2011. Date of publication September 06, 2011; date of current version April 11, 2012. This work was supported in part by the Hong Kong Applied Science and Technology Research Institute (ASTRI). R. Xu performed this work while at The Chinese University of Hong Kong, N.T., Hong Kong SAR, China. The associate editor coordinating the review of this paper and approving it for publication was Prof. Okyay Kaynak.

R. Xu is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: rzxu@mit.edu).

S. Zhou and W. J. Li are with The Chinese University of Hong Kong, N.T., Hong Kong SAR, China (e-mail: slzhou@mae.cuhk.edu.hk; wen@mae.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2011.2166953

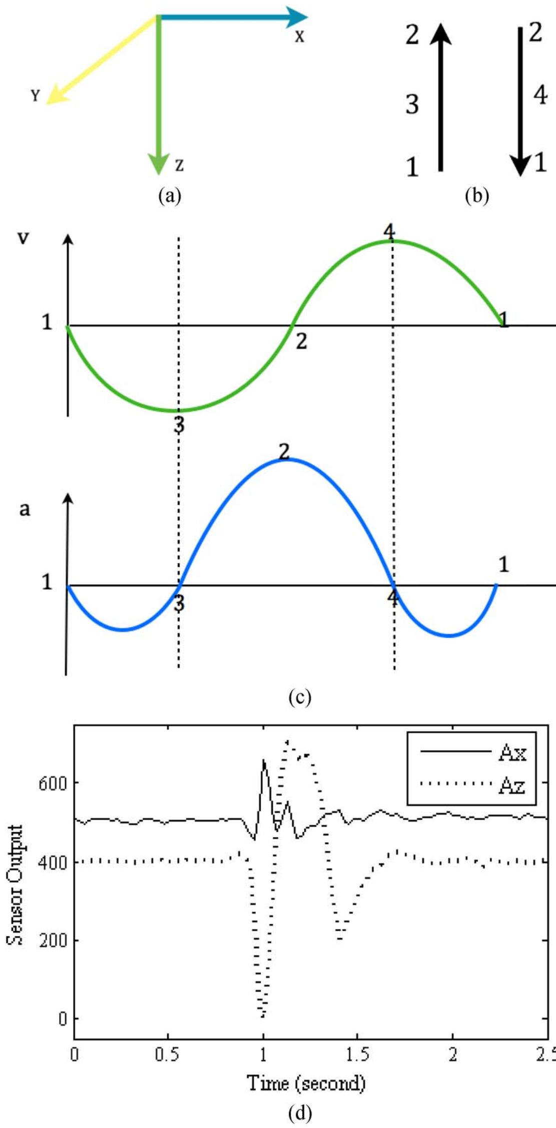


Fig. 1. Gesture *up* motion analysis. (a) Coordinate system. (b) Gesture *up* motion decomposition. (c) Predicted velocity and acceleration in the z-axis of the gesture *up*. (d) Real acceleration plot of the gesture *up*. Solid and dotted lines are accelerations on x- and z-axis, respectively.

in the vertical plane, so the accelerations on x- and z- axes are adequate to distinguish each gesture. Therefore, the acceleration on y-axis is neglected to reduce computational requirement.

We propose that the exact shape of the acceleration curves is not critical, but only the alternate sign changes of acceleration on the two axes are required to uniquely differentiate any one of the 7 gestures: *up*, *down*, *left*, *right*, *tick*, *circle*, and *cross*. This is the basis of the recognition algorithms discussed in this paper. For instance, the gesture *up* has the acceleration on z-axis in the order: negative —positive— negative (positive z direction points downward) and nearly has no acceleration on x-axis; for a *circle* gesture, on x axis: positive-negative-positive and on z-axis: negative-positive-negative-positive. Experiments showed that each of these gestures hasa special order of sign changes, and a kinematics analysis also proves this.

A kinematic motion a hand goes through in performing a gesture could nonintuitive at time. For example, a simple *up* gesture can be decomposed into several acceleration and deceleration periods. As shown in Fig. 1(b), an *up* gesture is actual consist of motion from point 1 to point 2, and then back to point 1. The velocity at the starting point 1, midpoint 2 and end point 1 are all zeros. For the convenience of analysis, point 3 is the point between point 1 and point 2 where acceleration changes sign, and point 4 is the point between point 2 and point 1 where acceleration changes sign. Then the acceleration changes can be described as:

$1 \rightarrow 3$: acceleration on z-axis is negative (since positive z direction is downward); velocity changes from zero to a maximum value at 3; acceleration at point 3 is zero.

$3 \rightarrow 4$: acceleration on z-axis is positive; velocity changes from negative to positive and is maximum at point 4, where acceleration becomes zero.

$4 \rightarrow 1$: acceleration on z-axis is negative; velocity changes from positive to zero. Also, acceleration and velocity become zero at point 1.

The analysis above is illustrated by Fig. 1(c). Fig. 1(d) is the real acceleration plot for the gesture *up*in which the dotted line is the acceleration on z-axis and solid line is the acceleration on x-axis. From Fig. 1(d), we note that noise exists from sensor measured data. However, the noise does not influence the trend of the acceleration curves, and hence, the analysis of gestures based on the above method still works without adding computational burdens on a CPU by using a noise-filtering algorithm. Comparing the predicted acceleration pattern in Fig. 1(c) with the real acceleration plot in Fig. 1(d), it is concluded that the trend of the real acceleration is the same with the prediction.

After analyzing the other gestures, it was found that they all have unique acceleration patterns for classification. Gesture *down* is similar to *up* but with changes in directions, *left* and *right*and also similar, but the changes in motion axes information. *Tick*, *circle* and *cross* are more complex since they have accelerations on both x- and z- axes simultaneously, but the accelerations on the two axes can be separated and decomposed, then the motion trend becomes similar to the above example. The uniqueness of each gesture trend makes the recognition algorithm possible, and the algorithms presented in this paper are based on this basic motion feature of the seven gestures.

III. Sensing System Overview

A. Sensor Description

The sensing system utilized in our experiments for hand motion data collection is shown in Fig. 2 and is essentially a MEMS 3-axes acceleration sensing chip integrated with data management and Bluetooth wireless data chips. The algorithms described in this paper were implemented and run on a PC. Details of the hardware architecture of this sensing system were published by our group in [19] and [20]. The sensing system has also been commercialized in a more compact form recently [21].

B. System Work Flow

When the sensing system is switched on, the accelerations in three perpendicular directions are detected by the MEMS sensors and transmitted to a PC via Bluetooth protocol. The gesture

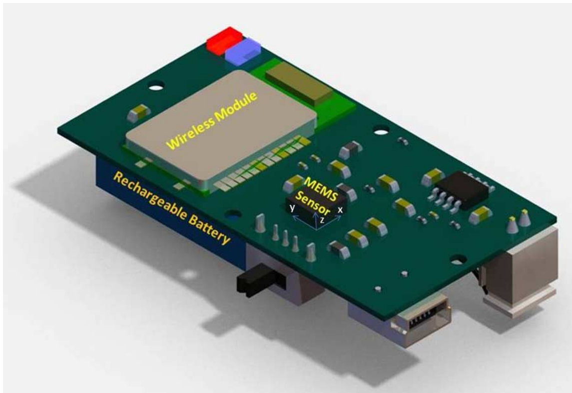


Fig. 2. Illustration of the components of the sensing system used for hand gesture recognition (the device shown as dimensions of $l = 6$ cm, $w = 4.5$ cm, $h = 2.5$ cm).

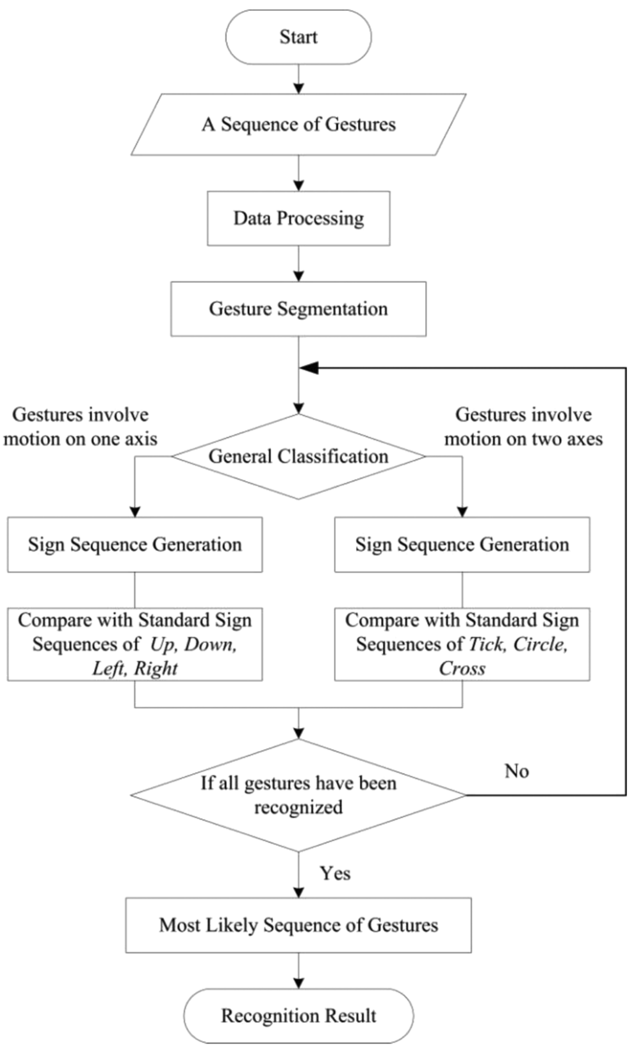


Fig. 3. Workflow of gesture recognition system using MEMS.

motion data then go through a *segmentation program* which automatically identifies the start and end of each gesture so that only the data between these terminal points will be processed to extract feature. Subsequently, the processed data are recognized by a *comparison program* to determine the presented gestures. The work flow of this system is shown in Fig. 3.

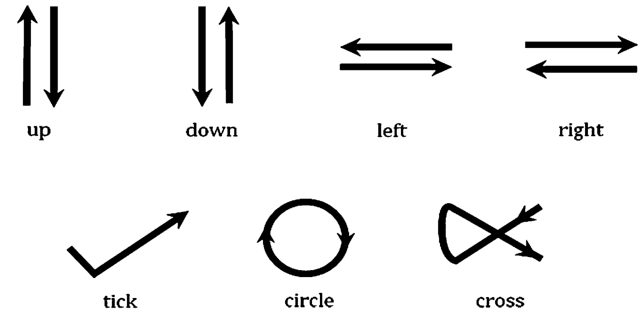


Fig. 4. Motions of seven gestures.

IV. Gesture Segmentation

A. Data Acquisition

To collect reliable hand gesture data for the sensing system, the experimental subject should follow guidelines below during the data acquisition stage:

- The sensing devices hould be held horizontally during the whole data collection process (i.e., the x-y plane of the sensor chip in Fig. 2 pointing towards the ground).
- The time interval between two gestures should be no less than 0.2 seconds so that the segmentation program can separate each one of the gestures in sequential order.
- The gestures should be performed as indicated in Fig. 4.

B. Gesture Segmentation

1) Data Preprocessing: Raw data received from the sensors are preprocessed by two 2 processes: a) vertical axis offsets are removed in the time-sequenced data by subtracting each data points from the mean value of a data set; hence, a data set shows zero value on the vertical axes when no acceleration is applied; b) a filter is applied to the data sets to eliminate high-frequency noise data.

2) Segmentation: The purpose of the *segmentation algorithm* is to find the terminal points of each gesture in a data set of gesture sequence. The algorithm checks various conditions of all the data points and picks out the most likely data points as the gesture termination points. The conditions of determining the gesture terminal points in our algorithm are a) amplitude of the points (y-coordinate value of a data point); b) point separation (the difference between the x-coordinates of the two points); c) mean value (mean of y-coordinates of points on left and right sides of a selected point); d) distance from the nearest intersection (quantifies how far is a selected point away from an "intersection point", i.e., a point where acceleration curve crosses from negative to positive or *vice versa*"); e) sign variation between two successive points. After examining all the points by checking these 5 different conditions, the terminal points can be generated for the motion data on each axis. Since, all these five conditions are checked separately on x- and z- axes acceleration data, two $2 \times n$ matrices are generated for each of gesture sequence data

$$\mathrm{a_x} = \begin{pmatrix} \mathrm{t_{11}} & \mathrm{t_{12}} \cdots \mathrm{t_{1n}} \\ \mathrm{t_{21}} & \mathrm{t_{22}} \cdots \mathrm{t_{2n}} \end{pmatrix} \quad \mathrm{a_z} = \begin{pmatrix} \mathrm{t_{11}} & \mathrm{t_{12}} \cdots \mathrm{t_{1n}} \\ \mathrm{t_{21}} & \mathrm{t_{22}} \cdots \mathrm{t_{2n}} \end{pmatrix}.$$

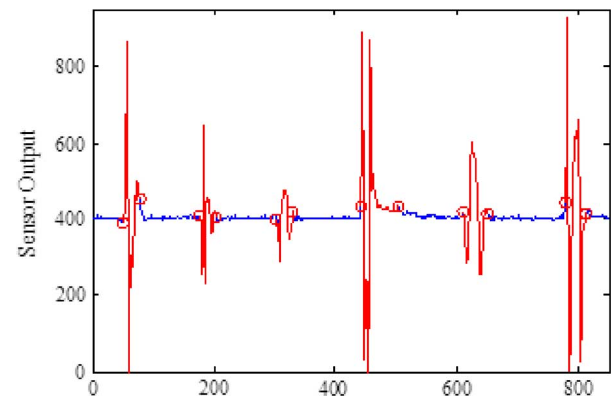


Fig. 5. Segmentation of a seven-gesture sequence in the order up-down-left-right-tick-circle-cross.

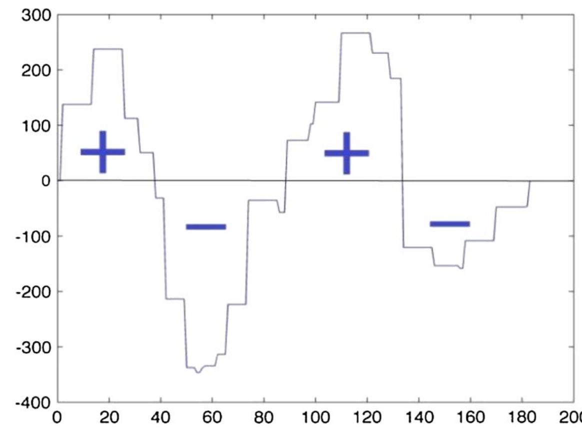


Fig. 6. Sign sequence generation.

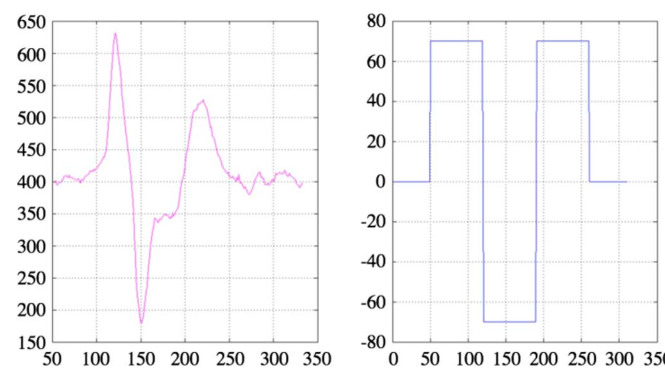


Fig. 7. Feature extraction transformation.

The element on the first row is the starting point of a gesture and the element in the second row in same column is the end points of the same gesture. Comparing the columns of the two matrices, if the pair of terminal points on one axis is close to a pair on the other axis, one pair of the terminal points will be eliminated. A final determination on if a given set of pairs of points are indeed terminal points is made by comparing the maximum acceleration between them with the mean value of the maximum accelerations between all pairs of points. If the former is too small, then that pair of points will be eliminated. As an example, the final terminal points for both x- and z- axes are denoted by circles in Fig. 5.

After obtaining the terminal points of each gesture, the number of gestures becomes obvious since every gesture has one starting point and one end point, i.e., the number of the columns of the final terminal points matrix is the number of gestures.

V. Model One: Gesture Recognition Based on Sign Sequence and Hopfield Network

A. Gesture Recognition

1) Feature Extraction: By comparing the maximum value and mean value of the acceleration of the same gesture on the x- and z- axes and setting corresponding flags, the gestures which only have motions on one axis (*up, down, left, right*) are separated from the gestures which involve 2-D motions (*circle, tick, cross*) to ease computational requirement.

To reduce influence of unstableness of a hand making the gestures, the algorithm uses the mean value of a certain number of acceleration points which is set dynamically according to the duration of the gesture to determine the sign sequence.

The feature extraction procedures are as follows: examine the sign of the first mean point of a gesture, store in gesture code, then detect the number of sign changes and store the alternate signs in sequence in the gesture code. Hence, for the gesture in Fig. 6, we get the code: 1, -1, 1, -1. The feature extraction process greatly reduces the data volume and Fig. 6 shows an example of sign sequence generation. The whole feature extraction process can also be illustrated by the transformation in Fig. 7.

2) Gesture Encoding: Before recognition, the obtained gesture code should be encoded first so that it can be restored later by Hopfield network. From our experiments, it was found that the maximum number of signs for one gesture on one axis is four, so if the x and z axes sign sequences are combined, there will be totally eight numbers in one gesture code. But, since the input for Hopfield network can only be "1" or "-1", we encoded the positive sign, negative sign and zero using the following rules:

- "1 1" represents positive sign;
- "-1 -1" represents negative sign;
- "1 -1" represents zero.

Hence, each gesture has a unique 16-number code. For instance, the first sign of the gesture in Fig. 6 is positive, so "1 1" is stored in the gesture code. The gesture data has three subsequent sign changes: from positive to negative, then from negative to positive, and finally from positive to negative; so s "-1 -1", "1 1", and "-1 -1" are stored to the gesture code; other numbers in gesture code should all be set to zeros, which can be represented by "1 -1".

3) Hopfield Network as Associative Memory: The involvement of Hopfield network as a recovery mechanism makes the recognition algorithm more fault tolerant. When part of the input is lost or wrong, the network can still retrieve the most likely pattern which has been stored previously. Hence, if there is a not serious deviation, the network will help to restore the gesture code to the correct pattern. To use Hopfield network as associative memory, a weight matrix should be constructed first; the

TABLE I
STANDARD PATTERNS FOR THE SEVEN GESTURES

	Gesture Codes															
Left	–1	–1	1	1	–1	–1	1	–1	1	–1	1	–1	1	–1	1	–1
Right	1	1	–1	–1	1	1	1	–1	1	–1	1	–1	1	–1	1	–1
Up	1	–1	1	–1	1	–1	1	–1	–1	–1	1	1	–1	–1	1	–1
Down	1	–1	1	–1	1	–1	1	–1	1	1	–1	–1	1	1	1	–1
Tick	1	1	–1	–1	1	1	–1	–1	1	1	–1	–1	1	1	1	–1
Circle	1	1	–1	–1	1	1	1	–1	–1	–1	1	1	–1	–1	1	1
Cross	–1	–1	1	1	–1	–1	1	–1	1	1	–1	–1	1	1	–1	–1

construction of the weight matrix is also the information storage process. The weight matrix is [18]

$$w = \Sigma_{p=1}^{P} s^p (s^p)^T - PI, \quad s^p \in \{-1, 1\}^n \tag{1}$$

where s^p is the pattern to be stored, P is the number of patterns to be stored and I is the identity matrix. The standard patterns for the seven gestures are listed in Table I.

Constructing weight matrix in this way guarantees the weight matrix is symmetric with zero diagonal elements, and according to the property of Hopfield network, the network will be stable and can always retrieve the closest standard pattern after a certain number of iterations. If the input to the network is s^q, then the retrieval is

$$v(0) = s^q, \quad u(1) = Wv(0) = \Sigma_{p=1}^{p-1} s^p (s^p)^T s^q - Ps^q \tag{2}$$

$$v(1) = sgn\,(u(n)) \ldots u(n) = Wv(n-1) \tag{3}$$

$$v(n) = sgn\,(u(n)) \tag{4}$$

where $v(n)$ is the output.

4) Gesture Comparison: After gesture code restoration, each gesture code is compared with the standard gesture codes. The comparison is made by calculating the difference between the two codes, i.e., the smallest difference indicates the most likely gesture and the recognition result is obtained.

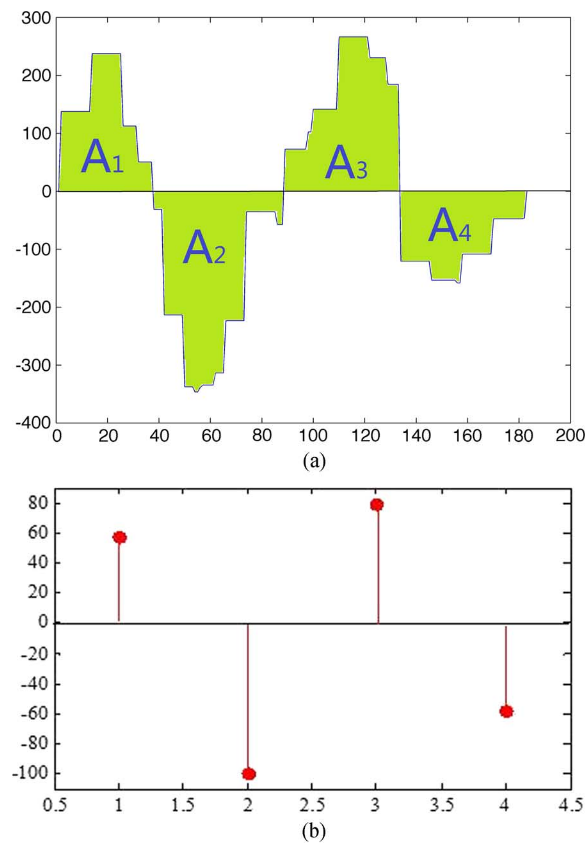


Fig. 8. Area sequence generation. (a) Acceleration partition and (b) area sequence generated from (a).

VI. MODEL TWO: GESTURE RECOGNITION BASED ON VELOCITY INCREMENT

The essence of this approach is to utilize a different feature which is the velocity increment or the area bounded by the acceleration curve and x-axis, to implement classification. The acceleration of a gesture on one axis is partitioned firstly according to the signs. As Fig. 8 shows, the acceleration pattern can be represented by areas in alternate signs. The physical meaning of these areas is the increase or decrease in velocity. Since the sign sequence may over reduce the information while the velocity increment or area sequence contains more information for discrimination, this approach is supposed to be a better method to deal with complex gestures.

Due to the intensity variance of each gesture, an area sequence should be normalized before stored as training data or compared with templates. Normalization is implemented using (5)

$$A_{\text{norm}} = \frac{A_{\text{original}}}{A_{\max}} \tag{5}$$

where A_{norm} is the normalized area, A_{original} is the original area, and $A_{\max}$ is the maximum area in a sequence. Since the comparison algorithm compares each pair of numbers separately, it is possible that misalignment happens due to noise. Hence, after normalization, the area sequences are not compared immediately but are processed by using an algorithm analogous to "center of mass," which is implemented by imagining the curve has mass and obtaining its coordinates of center of mass through calculation. Then, the two curves are

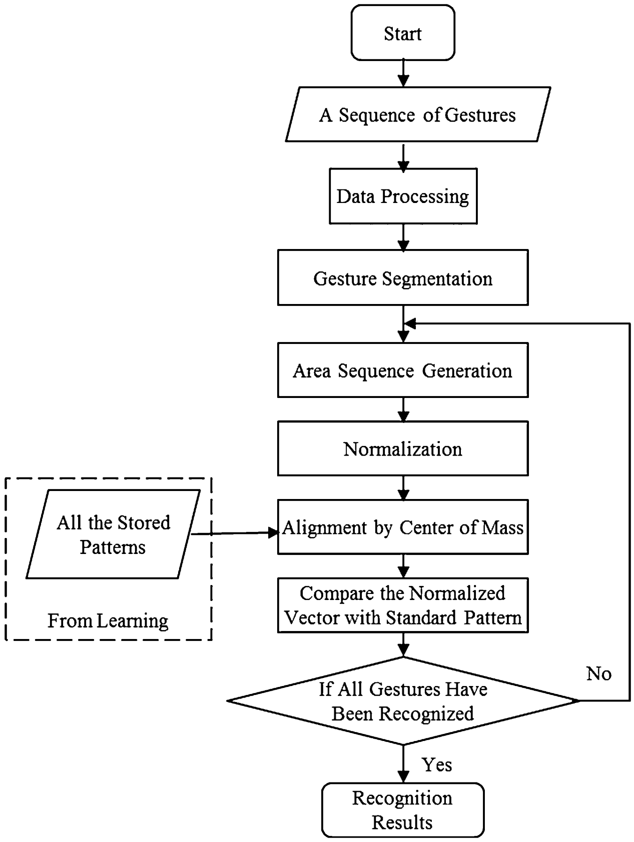


Fig. 9. Work flow chart of Model III.

aligned so that their centers of masses coincide. This algorithm is expected to reduce the possibility of misalignment. The final step is to compare the velocity increment sequence by subtracting two area sequence vectors

$$S_1 = [A_1, A_2, A_3 \ldots A_{n-1} A_n] \tag{6}$$

$$S_2 = \left[A_1' A_2' A_3' \ldots A_{n-1}' A_n'\right] \tag{7}$$

$$A_d = |A_1' - A_1| + |A_2' - A_2| + \cdots + |A_n' - A_n| \tag{8}$$

where S_1 and S_2 are the two area sequences to be compared and A_d is the comparison result. The gesture, which has the minimum value by comparing one area sequence with the training samples, can be recognized. The workflow of this model is shown in Fig. 9.

VII. Model Three: Gesture Recognition Based on Sign Sequence and Template Matching

The recognition algorithm of this model is very similar to that of model one, except that no Hopfield network is used, and hence encoding sign sequence into different combinations of -1's and 1's is not necessary. All the sign sequences are represented by -1, 1 and 0 as shown in Table II. The workflow of Model III is shown in Fig. 3. Since the algorithm is based on the feature of acceleration sign changes which is generalized from gesture motion analysis, it is not limited to specific users. Therefore, there is no requirement to train the system by specific users before using it.

TABLE II
Gesture Codes of Model Three

	X axis				Z axis			
Left	1	−1	1	0	0	0	0	0
Right	−1	1	−1	0	0	0	0	0
Up	0	0	0	0	1	−1	1	0
Down	0	0	0	0	−1	1	−1	0
Tick	−1	1	−1	0	−1	1	0	0
Circle	1	−1	1	−1	−1	1	−1	1
Cross	1	−1	1	0	−1	1	−1	1

TABLE III
Comparison of Gesture Recognition Accuracy (%) of Three Models

	Up	Down	Left	Right	Tick	Circle	Cross	Mean
Model I	95.0	86.0	91.0	84.0	64.0	75.0	61.0	79.0
Model II	87.0	19.0	63.0	94.0	25.0	0.0	88.0	54.0
Model III	94.8	91.1	96.7.0	100	94.4	97.7	94.4	95.6

VIII. Experimental Results

The experimental results of the three recognition models discussed above are listed in Table III. As shown in the table, Model III (*based on sign sequence and template matching*) achieves the highest accuracy among the three models, while the performance of Model II is the worst of the three. Besides, Model II is not as *robust* as the other two methods, i.e., variations of the input gestures are more likely to affect the outcome of the gestures recognized, so this model should not be preferred when MEMS accelerometers are used for gesture recognition. Since Model I and Model III have similar gesture encoding the recognition mechanism, only the evaluation result of Model III is provided in more detail in this paper.

The test results shown in Table III are based on 72 test samples, totaling 628 single gestures, i.e., each test sample consists of a sequence of input gestures in a particular order. They are collected in two kinds of gesture sequences: 1) in the order of *up—down—left—right—tick—circle—cross,* and 2) 10 same single gestures in onesequence, e.g., *circle—circle—circle—circle—circle—circle—circle—circle—circle—circle*. To increase data diversity and simulate variations in gestures made by different persons, gestures were made in different speeds and intensities; the trajectories of some gesture motions were made with some variation, e.g., an ellipse was made instead of a circle. Model III has an overall mean accuracy of 95.6%, with the recognition accuracy of each gesture above 90%. Table IV shows the detailed recognition results by using Model III, which shows the total number of input for each gestures and how many of the input are correctly recognized. We note here that, during experiments, some input gestures were not detected at all (i.e., due to loss of wireless transmission). For example, if the order of input gestures in one experimental sample is up-down-left-right-tick-circle-cross, the detected gestures may only be down-left-right-tick-circle-cross, i.e.,

TABLE IV
GESTURE RECOGNITION RESULTS FOR MODEL III

Experimental Results of Gestures		Recognized Gestures						
		Up	Down	Left	Right	Tick	Circle	Cross
Input Gestures	Up	**91**						
	Down	2	**82**			5		
	Left			**87**			1	
	Right			2	**90**			
	Tick	1				**85**	1	
	Circle						**85**	5
	Cross	2	8	1				**85**

TABLE V
COMPARISON OF GESTURE RECOGNITION ACCURACY (%) OF SEVERAL DIFFERENT ALGORITHMS

Algorithm	Average Accuracy (%)	No. of Testing Samples
DCT (Discrete Cosine Transform) [9]	94.6	370
DCT & Average Filtering [9]	87.6	370
DCT & Discrete Wavelet Transform [9]	90.5	370
Model I	79.0	628
Model II	54.0	628
Model III	95.6	628

only the last six gestures were detected. Moreover, sometimes a "ghost" gesture may be detected, i.e., due to environmental vibrations or unintended hand motions, the algorithm may "recognize" a gesture even though there was no intended gesture input. These "missing" or "ghost" gestures were not taken into account when "recognition accuracy" is determined, because they did not go through recognition process at all. We note here that the recognition performance using Model III is higher than the performance obtained by our group's prior work using HMM in [9]. A comparison of the results discuss in this paper and in [9] is provided in Table V. The experimental result proves that an algorithm based on sign sequence and template matching is efficient in recognizing gesture data from MEMS accelerometers without using a time consuming training process.

IX. CONCLUSION

This paper describes a nonspecific person gesture recognition system by using MEMS accelerometers. The recognition system consists of sensor data collection, segmentation and recognition. After receiving acceleration data from the sensing device, a segmentation algorithm is applied to determine the starting and end points of every input gesture automatically. The sign sequence of a gesture is extracted as the classifying feature, i.e., a gesture code. Finally, the gesture code is compared with the stored standard patterns to determine the most likely gesture.

Since the standard gesture patterns are generated by motion analysis and are simple features represented by 8 numbers for each gesture, the recognition system does not require a big data base and needs not to collect as many gestures made by different people as possible to improve the recognition accuracy. We note here, however, to enhance the performance of the recognition system, we will need to improve the segmentation algorithm to increase its accuracy in finding the terminal points of gestures. Moreover, other features of the motion data may be utilized for pattern classification, i.e., more recognition methods will be investigated in our future work.

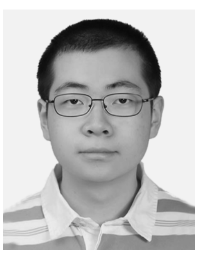
ACKNOWLEDGMENT

The authors would like to thank E. Q. Shan of the Hong Kong Applied Science and Technology Research Institute especially for her help on the segmentation program discussed in this paper.

REFERENCES

[1] T. H. Speeter, "Transformation human hand motion for telemanipulation," *Presence*, vol. 1, no. 1, pp. 63–79, 1992.
[2] S. Zhou, Z. Dong, W. J. Li, and C. P. Kwong, "Hand-written character recognition using MEMS motion sensing technology," in *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, 2008, pp. 1418–1423.
[3] J. K. Oh, S. J. Cho, and W. C. Bang *et al.*, "Inertial sensor based recognition of 3-D character gestures with an ensemble of classifiers," presented at the 9th Int. Workshop on Frontiers in Handwriting Recognition, 2004.
[4] W. T. Freeman and C. D. Weissman, "TV control by hand gestures," presented at the IEEE Int. Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, 1995.
[5] L. Bretzner and T. Lindeberg, "Relative orientation from extended sequences of sparse point and line correspondences using the affine trifocal tensor," in *Proc. 5th Eur. Conf. Computer Vision*, Berlin, Germany, June 1998, vol. 1406, Lecture Notes in Computer Science, pp. 141–157, Springer Verlag.
[6] D. Xu, "A neural network approach for hand gesture recognition in virtual reality driving training system of SPG," presented at the 18th Int. Conf. Pattern Recognition, 2006.
[7] H. Je, J. Kim, and D. Kim, "Hand gesture recognition to understand musical conducting action," presented at the IEEE Int. Conf. Robot & Human Interactive Communication, Aug. 2007.
[8] T. Yang, Y. Xu, and A. , Hidden Markov Model for Gesture Recognition, CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.
[9] S. Zhou, Q. Shan, F. Fei, W. J. Li, C. P. Kwong, and C. K. Wu *et al.*, "Gesture recognition for interactive controllers using MEMS motion sensors," in *Proc. IEEE Int. Conf. Nano/Micro Engineered and Molecular Systems*, Jan. 2009, pp. 935–940.
[10] S. Zhang, C. Yuan, and V. Zhang, "Handwritten character recognition using orientation quantization based on 3-D accelerometer," presented at the 5th Annu. Int. Conf. Ubiquitous Systems, Jul. 25th, 2008.
[11] J. S. Lipscomb, "A trainable gesture recognizer," *Pattern. Recognit.*, vol. 24, no. 9, pp. 895–907, 1991.
[12] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*. New York: McGraw-Hill, 1979.
[13] D. H. Rubine, "The Automatic Recognition of Gesture," Ph.D dissertation, Computer Science Dept., Carnegie Mellon Univ., Pittsburgh, PA, Dec. 1991.
[14] K. S. Fu, *Syntactic Recognition in Character Recognition*. New York: Academic, 1974, vol. 112, Mathematics in Science and Engineering.
[15] S. S. Fels and G. E. Hinton, "Glove-talk: A neural network interface between a data glove and a speech synthesizer," *IEEE Trans. Neural Netw.*, vol. 4, no. l, pp. 2–8, Jan. 1993.
[16] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. New York: Springer, 2006.
[17] T. Schlomer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a Wii controller," in *Proc. 2nd Int. Conf. Tangible and Embedded Interaction (TEI'08)*, Bonn, Germany, 2008, pp. 11–14.

[18] N. Popoviciu and M. Boncut, "On the Hopfield algorithm. Foundations and examples," *Gen. Math.*, vol. 13, no. 2, pp. 35–50, 2005.
[19] A. H. F. Lam, W. J. Li, Y. Liu, and N. Xi, "MIDS: Micro input devices system using MEMS sensors," presented at the IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), Oct. 2002.
[20] G. Zhang, G. Shi, Y. Luo, H. Wong, W. J. Li, P. H. W. Leong, and M. Y. Wong, "Towards an ubiquitous wireless digital writing instrument using MEMS motion sensing technology," presented at the IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM), Monterey, CA, Jul. 2005.
[21] Virtus Advanced Sensors [Online]. Available: http://www.virtusensors.com



Ruize Xu received the B.Eng. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK) in 2010. He is currently pursuing the M.S. degree in mechanical engineering at the Massachusetts Institute of Technology (MIT), Cambridge.

His research interests include microelectromechanical systems, vibration-to-electric energy harvesting, and pattern recognition.

Shengli Zhou received the B.Eng. degree in mechanical design, manufacturing, and automation from the School of Mechano-Electronic Engineering, Xidian University, China, in 2007, and the M.Phil. degree from The Chinese University of Hong Kong (CUHK) in 2009, with a research topic of human motion recognition. She is currently pursuing the Ph.D. degree at the Centre for Micro and Nano Systems (CMNS), CUHK, with a research topic of motion tracking with inertial sensor and webcam.

Wen J. Li (F'11) received the BSAE and MSAE degrees from the University of Southern California, Los Angeles, in 1987 and 1989, respectively, and the Ph.D. degree from the University of California, Los Angeles, in 1997.

He has been with the Department of Mechanical and Automation Engineering since September 1997 and currently heads the Centre for Micro and Nano Systems at The Chinese University of Hong Kong (CUHK). Before joining CUHK, he held R&D positions at the NASA Jet Propulsion Laboratory (Pasadena, CA), The Aerospace Corporation (El Segundo, CA), and Silicon Microstructures, Inc. (Fremont, CA).

Dr. Li has served as a guest editor for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, the IEEE/ASME TRANSACTIONS ON MECHATRONICS, and the IEEE SENSORS JOURNAL on special topics related to MEMS and Nanotechnology. He is currently the Editor-in-Chief of the *IEEE Nanotechnology Magazine*. His research group has published more than 270 technical papers related to MEMS and Nanotechnology and has won best conference paper awards from the IEEE Nanotechnology Council (IEEE-NANO 2003) and the IEEE Robotics and Automation Society (IEEE-ICRA 2003) for its work on micro cell-grippers and nano-sensors. He was elected IEEE Fellow in 2011 for his contribution in low power integrated carbon nanotube sensors.