Hand-Written Character Recognition Using MEMS Motion Sensing Technology

Shengli Zhou[1, 2], Zhuxin Dong[1, 2], Wen J. Li[1, 2, *], and Chung Ping Kwong[2]

[1]*Centre for Micro and Nano Systems*

[2]*Department of Mechanical and Automation Engineering*

The Chinese University of Hong Kong

Shatin, N.T., Hong Kong SAR, China

**Contact author: wen@mae.cuhk.edu.hk*

Abstract—**In this paper, a Micro Inertial Measurement Unit (µIMU) based on Micro Electro Mechanical Systems (MEMS) sensors is applied to sense the motion information produced by characters written by human subjects. The µIMU is built to record the three-dimensional accelerations and angular velocities of the motions during hand-writing. (Here we write the characters in a plane, so only two accelerations and one angular velocity are taken from µIMU in processing the data discussed in this paper). Then, we compared the effectiveness of data processing methods such as FFT (Fast Fourier Transform) and DCT (Discrete Cosine Transform) by showing their corresponding experimental results. Subsequently, we gave an analysis of these two methods, and chose DCT as the preferred data processing method. For character recognition (26 English alphabets and 10 numerical digits), unsupervised network Self-Organizing Map (SOM) is applied to classify the characters and comparatively good results are obtained. Our goal is to show the feasibility of character recognition based on selected sensor motion information, and provide a potential technology for human-gesture recognition based on MEMS motion sensors.**

Index Terms—MEMS, µIMU, SOM, FFT, DCT.

I. INTRODUCTION

With the advent of computing technologies, specifically in the areas of improved computational speed and reduced system cost with additional functionalities, real-time and virtual interactions between humans and computers have gained significant research and commercial interests recently. Our work discussed in this paper focuses on human interactions with computing devices using characters and gesture recognition. There are two primary character recognition methods based on different inputs: one is Optical Character Recognition (OCR), which gets data information by scanning the printed text; the other is Dynamic Character Recognition (DCR), which recognizes the characters based on their motion information, such as acceleration, angular velocity and so on.

Since DCR method can realize interactions between a human and a computer much more effectively than OCR method, and together with the rapid development of advanced micro sensor technology, quit a few research groups have recently explored the possibility of realizing DCR using MEMS motion sensors. An example of using MEMS motion sensors to recognize human gestures is given in [1], where the researchers try to recognize Vietnamese sign language using MEMS accelerometers [1]. Examples of research in using MEMS sensors in input devices to provide accelerations and angular velocities for DCR are given in references [2-7].

Having obtained motion information from an input device, in order to recognize the characters, theoretically the trajectory of the input device can be reconstructed by integrating its accelerations twice to get its position [8]. Also, the characters could theoretically be recognized directly just base on accelerations and angular velocities. However, due to various noise signals, which may include significant intrinsic drift of the sensors, circuit thermal noise, random noise etc. [9], practically it is extremely difficult to reconstruct the characters either directly or by double integration.

For example, we wrote two characters *"M"* and *"N"* using a MEMS sensor based input device on a transparent plane. At the same time, sequential images of the motion of the input device were recorded by a high speed camera while characters were written directly above (experimental set up is described in [9]). Then, we acquired the motion data information of each character in two forms: one is angular velocities and accelerations of the device measured by the MEMS sensors; the other is the motion trajectory of the input device recorded by the camera. We then obtained the trajectory of the input device by combining the sequential pictures (red line in Fig. 1) and compared it to position trajectory obtained using the MEMS sensors by double-integration of acceleration signals (green line in Fig. 1). In this Figure, the characters *"M"* and *"N"* are recognizable by the red lines, but are unrecognizable by the green lines, i.e., the MEMS acceleration sensors could not provide enough sensitivity and stability to give position trajectories as simple as *"M"* and *"N"*.

In addition, another method was explored to recognize the *"M"* and *"N"* characters. We wrote 10 times of each character respectively on a plane as mentioned above. And then, based on the accelerations and angular velocities obtained by the sensors in the experiment, the classification was done after the data were pre-processed. We find these two characters could be classified completely, that is, 100% correct rate was obtained. With this encouraging result, we extended our work to recognize all 26 English alphabets in addition to the numerical digits from *"0"* to *"9"*. The results of this work are discussed in this paper.

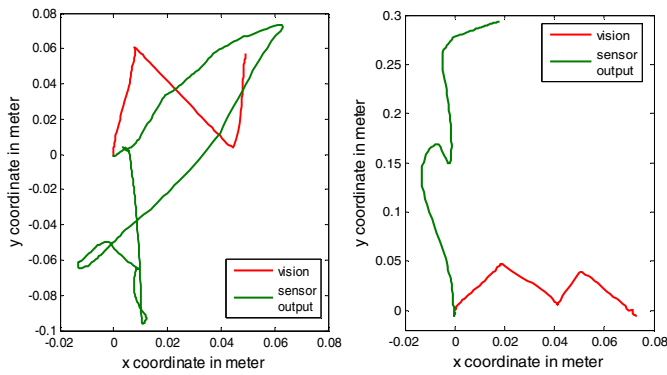In fact, similar experiments were performed to realize

Fig. 1 Comparison between original letters and reconstructed ones.

character recognition base on acceleration sensors' motion information by S. D. Choi *et al* [10]. In that paper, ten hand-written numbers were classified based on the 3D accelerations and two popular dynamic recognition methods HMM and DTW were applied to realize the character recognition. Different from their work, we employ an unsupervised network Self-Organizing Map (SOM) to carry out character recognition. As we will discuss in this paper, SOM gives relatively good results in recognizing some English alphabets and numerical digits.

This paper is organized as follows: In Section II, the experimental devices, data acquisition, data processing will be presented first, followed by a brief introduction to SOM. And then, detailed experimental results of motion based characters recognition will be discussed in Section III.

II. EXPERIMENTAL SETUP

A. Experimental device

As mentioned before, a Micro Inertial Measurement Unit (μIMU) based on Micro Electronic Mechanical System (MEMS) sensor is applied to measure the motion information produced by the input device when characters are written.

The experimental setup of our experiment is shown in Fig.2. Fig.3 is the hardware structure of our input device. Accelerations and angular velocities can be detected by the 3 accelerometers and 3 angular rate gyroscopes, respectively.. The output signals of the accelerometers (AX, AY) and the rate gyros (GXY) are measured directly with an A/D converter inside the microcontroller [11]. The sensor is connected to a computer through a USB interface when the characters are written on a smooth plane. The transformed digital signal is transmitted to a computer for off-line analysis through a USB port.

B. Data acquisition

50 trials were conducted for each character from *"0"* to *"9"*, and *"A"* to *"Z"*. In general, we write *"0" (the numerical digit)* and *"O" (the alphabet)* in a same way, so here we replace *"0"* for *"O"*, i.e., only 35 characters are written.
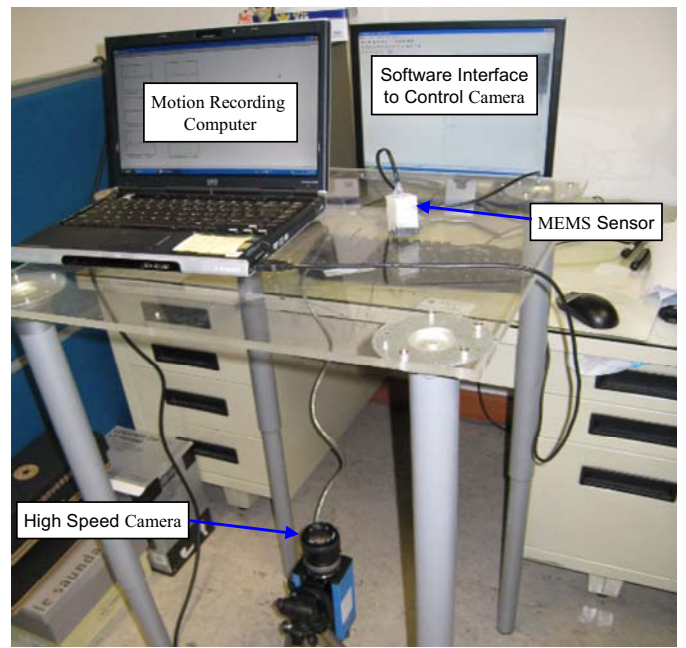

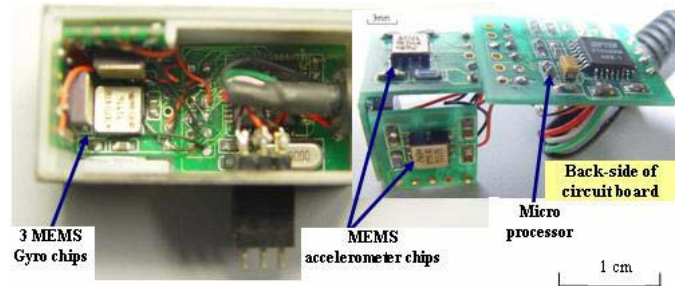Fig. 2 Experimental setup.

Fig. 3 Illustration of the 3D Motion Sensing System.

There are some basic rules that we followed when writing the characters during the experiments. Firstly, since the cross section of our μIMU is about 25 mm×25 mm, the characters written were all about 65mm long and 65mm height. Secondly, when writing the characters, the attitude of the μIMU was kept consistent, and there should be no pause during each character writing (i.e., each character should be completed in a continuous stroke), otherwise, the data could be hard to recognize. Lastly, the tip of the input device was always kept in touch with the horizontal writing plane during the writing motions, i.e., to ensure that the writing motion is in a two dimensional plane as much as possible.

We recorded 1750 experimental results consisting of 500 numbers and 1250 capital letters, which form a database of these characters. Each result includes 3 arrays measured by 3 corresponding sensors respectively. For example, Fig.4 shows original motion data recording of character *"H"* including two accelerations and one angular velocity. GXY is the angular velocity on the plane. AX and AY are accelerations in X, Y directions.
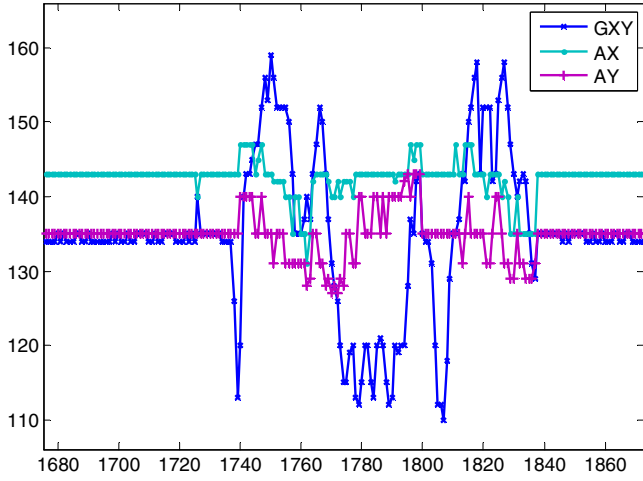
1419

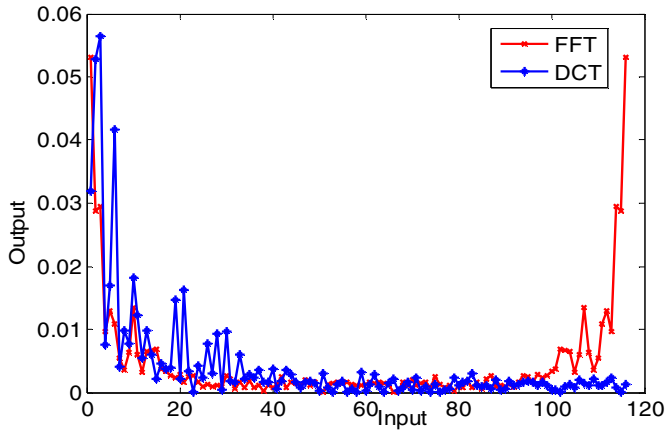Fig. 4 Original motion data of character "*H*".

Fig. 5 Comparison of data processed by FFT and DCT.

C. Data processing

We performed data pre-processing to filter noise and reduce dimension. FFT (Fast Fourier Transform) is the fundamental method to compress data and transform data from time domain to frequency domain, and it has been widely used in various areas. Its transforming algorithm is shown as follows:

$$X_{kfft} = \sum_{n=0}^{N-1} e^{-2\pi ik.(n/N)} x_n = \sum_{n=0}^{N-1} x_n (\cos(\frac{2\pi kn}{N}) - j\sin(\frac{2\pi kn}{N})) \quad (1)$$

Where N is the number of points in FFT, X_{kfft} is the kth point of the output data.

At the beginning, we took FFT transform for 11 capital letters: *"A", "B", "C", "D", "E", "F", "G", "M", "N", "S", "W"*, and then imported the data into SOM workplace to classify them. Unfortunately the experimental results were unsatisfactory. As shown in Table I, the four characters *"M"* and *"N", "D"* and *"S"* were always mixed up though the correct rate of other characters is more than 66%.

In order to reduce this confusion, we try using another data compressing method, i.e., DCT (Discrete Cosine

Transform, hose transforming algorithm is shown below in Eq. (2) to compress these data, and then compared its experimental results with FFT:

$$X_{kdct} = \sum_{n=0}^{N-1} x_n \cos(\frac{\pi k}{N}(n+\frac{1}{2})) \quad (2)$$

Where N is the points in DCT, X_{kdct} is the kth point of the output data.

Table II shows the experimental result of these 11 capital letters pre-processed by DCT. Form Table II, we find there are not so many characters that were mixed up as compared to results shown in Table I, and the correct rate of these characters is more than 70%.Therefore, we have deemed that DCT is more appropriate for our character recognition experiments. In the later experiments, all other characters were pre- processed by DCT.

In fact, from their transformation formula we can see that DCT is similar to FFT, but using only real numbers. The data processed by FFT will have more coefficients than by DCT, so it is hard to ensure its precision. Though they are all periodic functions, data processed by DCT is continuous on the boundary point, whereas FFT is not. There may be much noise produced due to the discontinuity.

After normalizing the transformed data, we compared the FFT and DCT processed data as shown in Fig. 5. From this Figure, the data processed by DCT fluctuates more than FFT, and hence they are easier to recognize during the classification

TABLE I
EXPERIMENTAL RESULTS OF FFT

| | | Recognized Characters | | | | | | | | | |
		A	B	C	D	E	F	G	M	N	S	W
Written Characters	A	43				2	1	1			3	
	B		47					3				
	C			49				1				
	D		2		48							
	E					43					7	
	F	1				2	46				1	
	G					3		47				
	M						1		47	1	1	
	N								27	3		20
	S			4	43		1	2				
	W									16	1	33

TABLE II
EXPERIMENTAL RESULTS OF DCT

| | | Recognized Characters | | | | | | | | | |
		A	B	C	D	E	F	G	M	N	S	W
Written Characters	A	40				2	1	6		1		
	B		45		2	1	1	1				
	C			48							2	
	D		3	1	46							
	E					50						
	F					6	42			1	1	
	G	4				5	3	38				
	M								47	3		
	N					1	4		4	41		
	S			14		1					35	
	W						1	1	4			44

process. In addition, we find that if we take the first twenty orders of the transformed data, then FFT takes 70% of the total energy while DCT takes 96%. Hence the data information processed by DCT is more concentrated, and less information is lost during dimensional reduction.

D. Introduction to SOM

Kohonen's Self-Organizing Maps (SOM) is simple analogs of the human brain's way of organizing information in a logical manner. They can learn to classify input vectors according how they are grouped in the input space. SOM is a self-competitive network. It is a single-layer network with a winner-take-all layer using an unsupervised learning algorithm. During the self-organizing procedure the topologically close relationship of the organized information is maintained. The data are mapped from high dimensional to one or two dimensional feature maps.

In the training procedure, the network finds the neuron that is closest to the input data vector by calculating their Euclidean distance and declares the neuron with minimum Euclidean distance as the winning neuron. Then the weights of all the neurons in the vicinity of the winning neuron are updated, by an amount inversely proportional to the distance. The training process is terminated if a prescribed number of iterations are reached. The network does not give direct information on the actual physical classification of the data samples, but merely clusters and classifies the data set based on the set of attributes used [12].

SOM is well known for its ability to identify regularities and correlations, and perform automation clustering and categorization of input data [13,14], so it is one of the most fascinating topics in neural networks. Owing to its advantage in needing less data, simpler algorithm, and powerful in clustering analysis, SOM has been applied in many fields. In our project, we used SOM to realize character (26 English letters and 10 numbers) classification.

III. EXPERIMENTAL RESULTS

As mentioned before, each character from *"0"* to *"9" and "A"* to *"Z"* was written 50 times on a plane in our experiments, and the information we obtained is angular velocity on X-Y plane and accelerations in x and y directions. However, as noted earlier, we wrote *"0"* and *"O"* in a same way, therefore only 35 characters were written in our experiments, and the total samples of these characters are 1750. For each experimental result, we performed a DCT transform of the 3 sensor data arrays respectively, and then we took the first twenty coefficients of each DCT. After this transformation, we obtained a matrix having 1750 rows and 60 columns. Each row is a sample which consists of 3×20 elements in the sequence of GXY, AX, and AY.

Classification was done in two ways: one was to putting all these characters in a same group to recognize them; the other was to separate the alphabets from the numerical digits, and then classifying the numbers and letters respectively.

TABLE III
CHARACTER RECOGNITION RESULTS

Recognized Character	1				7			I		M		X	
Actual Input	0	1	6	U	7	C	X	F	I	M	N	5	X
Mixed Up Count	43	48	39	24	44	44	30	23	28	31	22	19	14

(a) Mix up Characters

Input	4	7	8	9	A	B	D	E	G	I
Correct Rate (%)	86	88	94	54	40	76	92	70	76	56
Input	K	L	M	P	S	T	V	W	Z	
Correct Rate (%)	64	44	62	64	52	52	92	86	92	

(b) Correct Rate of other Characters

TABLE IV
EXPERIMENTAL RESULTS OF NUMBERS

Experimental Results of Numbers		Recognized Characters									
		0	1	2	3	4	5	6	7	8	9
Input	0	47		1					2		
	1		49					1			
	2			44	1			2	1		1
	3		1		44					2	3
	4			3		39		7	1		
	5				1		45	2			2
	6	21	3	8				12			6
	7	1	1						48		
	8		1					1		46	2
	9			4	1			3			42

A. Recognize all the characters in a same group

In this case, the experimental results can be divided into three groups: mixed up group, bad classification group and other characters. Table III shows part of the recognition results of this original method.

From Table III (a), we find there are too many characters mixed up. For example, *"0", "1", "6",* and *"U"* are classified into a same group and recognized as number *"1"; "7", "C"* and *"X"* are also mixed up. Besides, there are some other characters such as *"2", "3", "5", "H", "J", "Q", "R" and "Y"* etc., which are almost evenly divided into other characters' group and recognized as other characters.

We list the correct rate of the other characters that have comparatively better recognition rate, as shown in Table III (b). From this table, we find their correct rates are also unsatisfactory.

Besides the poor performance, owing to the large input matrix, at least 40 minutes is required each time to train the network, so it is not an ideal method to classify the characters.

B. Separating numbers from letters before classification

We also tried another approach: separate the data set of the letters and numbers before using the network to classify them respectively. Table IV shows the recognition results of the numbers. From this table we see that most of these numbers can be recognized with a correct rate higher than 78% except for the number *"6"*. There are about twenty numbers *"6"* and forty-seven numbers *"0"* that got mixed owing to their similar writing stroke. It can be concluded that

1421

TABLE V
EXPERIMENTAL RESULTS OF CAPITAL LETTERS

		Recognized Characters																								
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Characters Written	A	36					1	5		1	1			1					1			4					
	B		36					1			1					10						2					
	C			45						1	1						2									1	
	D		2		45											2					1						
	E					40				10																	
	F					1	40		1	5	1						1	1									
	G				3			36		6	2															3	
	H						8		36		2			4													
	I				3		15		1	15	14															1	
	J								1	2	35		6				6										
	K				1	1			2	1		28		1					13			2				1	
	L				1								32			5	12										
	M								3					47													
	N								6					25	17										1		1
	O															50											
	P						3		1		1			7			1		5							3	29
	Q								1				1	1					7		2				3	19	16
	R	10				1	1		2		2	4		5					23		2						
	S			2					1								1			44						2	
	T			2			2		1	1			2				5				19	1				17	
	U													1		43	1						5				
	V								2							1	1				1		45				
	W								1				3	2										44			
	X																								50		
	Y					2		2		1		11	1	1			3	4								25	
	Z													1			1										48

the numbers can be recognized better this way by comparing the results listed in Table III and IV.

Then the same method was applied to classify the 26 capital letters, and its corresponding result is summarized in Table V. By analyzing the table we find there are also several characters which were mixed up, i.e., *"M"* and *"N"*, *"O"* and *"U"*, *"Q"* and *"Y"*, *"P"* and *"Z"*. However, the number of counts of mixed up occurrence is much less than in Table III. In addition, the correct rate of some characters, such as *"O"* and *"X"*, reaches 100%.

IV. CONCLUSION

In this paper, we present our latest work on character recognition based on the acceleration and angular velocity motion of the input device when writing characters in a 2D plane. A µIMU built using MEMS sensors is employed for detection of the motion associated with writing the 26 English alphabets and the numerical digits from *"0"* to *"9"*. Experimental data is transmitted to a computer via a USB interface for post-analysis. For data pre-processing, after comparing FFT and DCT results, we find that DCT is more proper for data compression. We also use SOM as a character recognition method for training and classifying data transformed by DCT. Due to the inherent MEMS sensor noises, which are impossible to totally eliminate, sometimes it is not feasible to reconstruct the exact shape of hand-written characters, so our work has the advantage of recognizing the characters based on acceleration and angular velocity information, rather than having have to reconstruct the character shapes by double-integration of acceleration to obtain position.

Experimental results show that it is feasible to recognize characters based on the acceleration and angular velocities with a considerable correct rate except for some characters. Therefore we can extend character recognition into gesture recognition in our future works. However, during the experiment, we also find that SOM has some limitations which can affect its classification results, such as its sensitivity to data input sequence and signal level. Therefore, we will try to improve this network in future.

REFERENCES

[1] The Duy Bui, and Long Thang Nguyen, "Recognition of Vietnamese Sign Language Using MEMS Accelerometers", *in Proc. IEEE International Conference on Sensing Technology*, pp. 118-122, November 21-23, 2005.

[2] Won-Chul Bang, Wook Chang, Kyeong-Ho Kang, Eun-Seok Choi, Alexey Potanin, and Dong-Yoon Kim, "Self-contained Spatial Input Device for Wearable Computers", *in Proc. the Seventh IEEE International Symp . on Wearable Computers*, pp. 26-34, 2003.

[3] A. D. Cheok, K. G. Kumar, and S. Prince, "Micro Accelerometer Based Hardware Interfaces for Wearable Computer Mixed Reality Applications", *in Proc. of the 6th IEEE International Symposium On Wearable Computers*, pp. 223-230, October. 7–10, 2002.

[4] H. Ishikawa, Y. Yonezawa, K. Itoh, and M. Hashimoto, "Extraction of Character Information from Acceleration of Handwriting Pen", *Trans. Society of Instrument and Control Engineers,* vol. J76D-I, pp. 541–543, 1993.

[5] D. Strickland, A. Patel, C. Stovall, J. Palmer, and D. McAllister, "Self Tracking of Human Motion for Virtual Reality Systems", *in Proc. SPIE Int. society of Instrument and Control Engineers,* vol. 2177, pp. 278–287, 1994.

[6] Lee, G.-J. Nam, J. Chae, H. Kim, and A. J. Drake, "Two-Dimensional Position Detection Systems with MEMS Accelerometer for Mouse Applications", *in Proc. of the 38th IEEE International Conference on Design Automation,* pp. 852–857, June 2001.

[7] B. Milner, "Handwriting Recognition Using Acceleration Based Motion Detection", *in Proc. IEEE Coll. Document Image Processing and Multimedia,* London, UK, pp. 5/1–5/6.

[8] Chi Chiu Tsang, Gary Chun Tak Chow, Guanglie Zhang, Yilun Luo, Zhuxin Dong, Guangyi Shi, and Wen J. Li, etc., "A Novel Real-Time Error Compensation Methodology for μIMU-based Digital Writing Instrument", *in Proc. of 2006 IEEE International Conference on Robotics and Biomimetics(IEEE-ROBIO 2006)*, pp. 678-681, Kunming, China, 17-20 December 2006.

[9] Zhuxin Dong, Guanglie Zhang, Yilun Luo, Chi Chiu Tsang, Guangyi Shi, and Wen J. Li , etc., "A Calibration Method for MEMS Inertial Sensors Based on Optical Tracking", *in Proc of 2007 IEEE International Conference on Nano/Micro Engineered and Molecular Systems (IEEE-NEMS2007),* Bangkok, Thailand, 16-19 January 2007.

[10] Sung-Do Choi, Alexander S. Lee, and Soo-Young Lee, "On-Line Handwritten Character Recognition with 3D Accelerometer", *IEEE International Conference on Information Acquisition,* August 20-23, 2006.

[11] Yilun Luo, Guangyi Shi, Josh Lam, Guanglie Zhang, Wen J. Li1, Philip H. W. Leong, etc., "Towards a Human Airbag System Using μIMU with SVM Training for Falling-Motion recognition", *IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO '05),* June 29, 2005.

[12] M. T. Taner, "Kohonen's Self Organizing Networks with 'Conscience'", *Rock Solid Images,* pp. 1-7, November 1997.

[13] A. M. Fanelli, *Recent Advances in Artificial Neural Networks: Design and Applications*. Boca Raton: CRC Press LLC, 2000.

[14] Xiaoyong Yang, "An Enhanced SOFM Method for Automatic Recognition and Identification of Digital Modulations", *in Proc. of the 2th IEEE International Workshop on Electronic Design, Test and Applications (DELTA'04),* January 2004.