Handwritten Character Recognition Using Orientation Quantization Based on 3D Accelerometer

Shiqi Zhang [1, 2]   Chun Yuan [1]   Yan Zhang [2]

[1] Information Sci. and Tec. Division
Graduate School at Shenzhen
Tsinghua University
Nanshan, Shenzhen, Guangdong, China

[2] Dept. of Electronic and Info. Eng.
Shenzhen Graduate School
Harbin Institute of Technology
Nanshan, Shenzhen, Guangdong, China

zhangshiqihit@yahoo.com.cn   yuanc@sz.tsinghua.edu.cn   ianzhang@hit.edu.cn

ABSTRACT

This paper presents an online handwritten character recognition system. The whole system includes three parts: acceleration signal detection, signal processing and recognition by Hidden Markov Model (HMM). In hardware aspect, a mini-board with a three-dimensional accelerometer and a microcontroller is used to get real time acceleration values and send them to a terminal continuously. After effective section extraction and lowpass filtering, different quantizing methods based on acceleration orientation are used to quantize numerous data into small integral vectors. At last, we use HMM to do the recognition. For the experiments with 10 Arabic numerals, this system shows a high Recognition Rate (R.R.) of 94.29% in the database of 42 models for every Arabic numeral. This system could be used to reduce the size of handheld devices by discarding number keys and make human computer interaction more convenient and interesting.

Keywords

Handwritten Character Recognition, Accelerometer, Hidden Markov Model, Quantization based on Orientation, MEMS

1. INTRODUCTION

Ubiquitous (Pervasive) computing requires “anytime, anywhere, anyhow access of computing for anybody”. Ubiquitous computing devices should be tiny, effortlessly portable and constantly available, and it is better to be embedded in the environment as a distraction-free system. Many new kinds of technologies on Human-Computer Interaction (HCI) are thriving these years, such as speech recognition, vision-based gesture recognition and tablet types of devices, and they are much more effective and efficient than the traditional manner [1]. Thus, users could interact with computing devices more freely and directly without keyboards. However, there exist some innate drawbacks in them. For example, it is impolite to communicate with a computer in some quiet public places for speech recognition users, and firmly fixing a camera outside a user is very inconvenient too. Gesture recognition based on inertial devices is an original, direct human-computer interacting way, because the motion of limbs is driven by the force from muscles and the inertial devices could detect the outcome of the force, say acceleration and angular velocity, directly, so people could even use them without any distraction all the time.

2. RELATED WORK

Generally, the whole process of gesture recognition based on inertial devices could be divided into three parts, original data sampling, feature extraction and recognition algorithm. For every single part, there are several methods or models.

To sample acceleration and angular velocity, inertial devices like accelerometers and gyroscopes are widely used in online system [1], [2]. However, if gyroscopes are used, there must be a large number of sine/cosine and coordinate transform operations. It would bring a heavy computational burden to the processor. Ubiquitous computing should be low-cost and avoid heavy calculation. So if a task could be completed only by accelerometers, gyro shouldn’t be used. An online character recognition system means that not the trajectory but the acceleration or velocity is considered and during the sampling process the system must be able to record the real-time status by time. Correspondingly, the offline system needs only the trajectory of a character.

To extract features from the gestures, we could use the trajectory by integrating the acceleration twice, the velocity by integrating the acceleration one time or use the acceleration directly. From [3], we know that recognition based on acceleration shows a high recognition rate, and the poor performance of the other two methods come from the accumulated offsets during integrations. In [4], Eun-Seok Choi extracts features from acceleration curves by detecting feature points, which are defined as local minima or local maxima points where the signal values are minimal or maximal within some time interval. In vision-based gesture recognition, Mahmoud Elmezain has calculated the orientation between gravity center of the trajectory and each point in a pure path. Then the orientation is quantized to give a discrete vector [5]. In this paper, we select the latter to quantize the acceleration signal.

Finally, many algorithms have been used to recognize gestures like HMM, Bayesian Networks (BN) and Dynamic Time Warping (DTW). Most researchers use Discrete HMM to find the most probable activity state [5], [6], [7], which doesn’t involve much math calculation. In [3], both Continuous Hidden Markov Model (CHMM) and DTW are implemented and they used HMM Tool Kit (HTK) to do recognizing operation. HTK from Cambridge has perfect performance in Speech Recognition; however, it costs

MobiQuitous 2008, July 21-25, 2008, Dublin, Ireland.
Copyright 2008 ICST ISBN 978-963-9799-27-1

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3898
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3898

much hardware resources for handheld embedded applications. In [1], [4], BN is used to recognize Arabic Numerals. In this paper, we choose Discrete Hidden Markov Model (DHMM) as the recognition algorithm and a software named HMMSim [8] is used to assist the processing.

Acceleration-based Gesture Recognition has been widely used to improve the quality of Human-Computer Interaction (HCI), such as sign language recognizing, medical monitoring, handwritten character recognizing and many other applications [10]. In [9], Duy Bui has successfully recognized all the postures in Vietnamese Sign Language by six accelerometers. If this design is combined with the technology of speech synthesis, it would be quite helpful for the deaf and dumb to “speak”. Moreover, in recent years, many researchers have developed devices to assist persons with abnormal acting habits in developing normal pattern of action [2], [6].

Reminder of this paper is organized as follows. First, the hardware of this system is introduced. After that is the main part, feature extraction, which includes effective section extraction, coordinate translation, filter algorithm, and quantization based on orientation. Then we describe the approach of HMM including training and the testing process of handwritten character. Experimental results are discussed before the final part, a brief conclusion.

3. HARDWARE SYSTEM

The 3D accelerometer MMA7260T we used is made by Freescale Semiconductor, which has selectable measuring range (1.5g/2g/4g/6g), a high sensitivity (800mV/g @ 1.5g) and a very small Quad Flat Package (QFP) of 6mm×1.45mm. MC9S08QG8 is an 8-bit microcontroller made by Freescale Semiconductor too. It has many excellent features, such as 20-MHz working frequency, 8 Kbytes FLASH, 512 bytes RAM, 8-channel 10-bit analog to digital converter and a serial communication interface module. A snapshot is shown as figure 1.

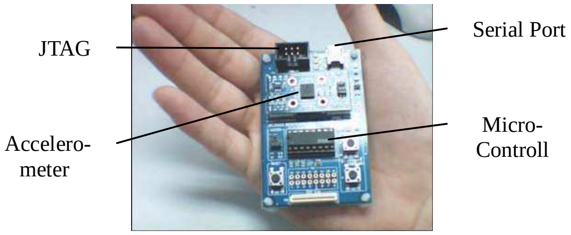


Figure 1. A snapshot of hardware system

Both the accelerometer and the microcontroller are fixed on a small Printed Circuit Board (PCB), which is size of 47×72 (mm). The mini-board is connected to a computer with a serial port line which is used to transmit acceleration signals and power the board at the same time. It is just a demo board and if this system is implemented in practical products, the board could be made much smaller because the components like JTAG, Serial port and the buttons will be discarded or laid inside the PCB.

4. ALGORITHM

The aim of feature extraction in character recognition is to get as much information as possible from the original signal, which could be transferred through serial port line continuously, and try to find a balance between quantization quality and algorithm complexity. Considering that no matter in which direction the user draws, every handwritten character is drawn in a 2-dimensional plane, we could discard the acceleration in z-axis. Thus all the operations could be simplified from 3D to 2D. The flow chart of this system in software aspect is shown as figure 2.

4.1 Effective Section Extraction

Before extracting acceleration information, the data stream should be divided into many single units that contain the information of one handwritten character, so effective section extraction module is needed to be executed before the others.

To detect the beginning of a handwritten character, a threshold acceleration value θ is set according to experience. If the acceleration change in x-axis or y-axis surpasses this threshold, it would represent a beginning of a handwritten character. The end of a character means no motion for this character any more, so we define the end to be the accelerations’ keeping invariable in both x-axis and y-axis for a short time. Following equations present the method to detect the beginning and end points.

$$\begin{cases} \left|a_x(T+1)-a_x(T)\right|<\theta \quad and \quad \left|a_y(T+1)-a_y(T)\right|<\theta \quad T<T_0 \\ \left|a_x(T+1)-a_x(T)\right|>\theta \quad or \quad \left|a_y(T+1)-a_x(T)\right|>\theta \quad T=T_0 \end{cases} \tag{1}$$

$$\begin{cases} \left|a_x(T+1)-a_x(T)\right|<\phi \\ \left|a_y(T+1)-a_x(T)\right|<\phi \end{cases} \quad for \quad T=T_{end},T_{end}+1,\cdots,T_{end}+N \tag{2}$$

where, T_0 is the sequence number of beginning, T_{end} is the sequence number of the end, θ is the acceleration threshold at the beginning, Φ is the acceleration threshold of the end and N denotes the holding time of the end. In this way, a matrix A could be constructed and it has and only has all the acceleration data of a handwritten character in both x-axis and y-axis.

$$A=\begin{bmatrix} a_x(T_0) & a_x(T_0+1) & \cdots & a_x(T_{end}) \\ a_y(T_0) & a_y(T_0+1) & \cdots & a_y(T_{end}) \end{bmatrix} \tag{3}$$

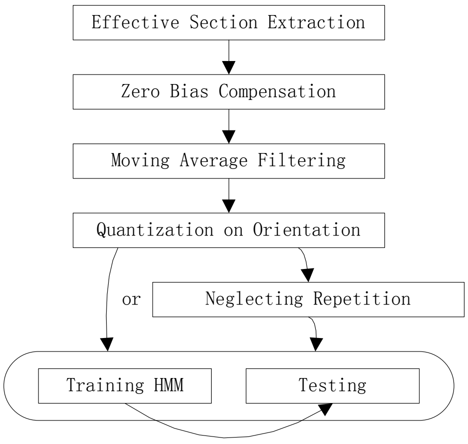


Figure 2. Flow chart of algorithm

4.2 Zero Bias Compensation and Filtering

The output of an accelerometer is a constant voltage called zero bias when it is stationary in every direction. The constant is a

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3898
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3898

summation of the gravity acceleration and a drift. There are many unexpectable environmental factors make drift keep changing all the time, like temperature and electric field. The drift could be regarded to be random as a function of time but constant within a short and certain period of time. So the module of zero bias compensation (coordinate regularization) is used to discard the zero bias of acceleration data. In following equations, a_0 is the average acceleration during the time from T_0 to T_{end}, and the values in x-axis and y-axis are not distinguished here for their similarity.

$$a_0 = \frac{1}{T_{end} - T_0} \sum_{i=T_0}^{T_{end}-1} a_r(i) \tag{4}$$

$$a(i) = a_r(i) - a_0 \quad i = T_0, \cdots, T_{end} \tag{5}$$

where, $a_r(i)$ is the original acceleration value in either x-axis or y-axis and $a(i)$ is the value after zero bias compensation.

The accelerometer output is a serial of discrete acceleration values. Due to accelerometer structures and random vibrations, there exists some noise in the output. Average Filtering is one of the simplest methods of filter algorithm.

$$a(j) = \frac{1}{M} \sum_{i=j}^{j+M-1} a(i) \quad j = T_0, \cdots, T_{end} - M \tag{6}$$

where, $a(j)$ is the acceleration value in either x-axis or y-axis, M is a parameter to adjust the strength of filtering.

For the ten Arabic Numerals written in the way as figure 3 shows, acceleration value curves are drawn as a function of time in figure 4. Green curve (light) is the acceleration in x-axis and blue (dark) is for y-axis.

4.3 Quantization Based on Orientation

Feature extraction is a very important part in our method to recognize the handwritten characters. In vision-based gesture recognition, researchers use location, velocity and orientation as the main features. Previous researchers [5] have concluded that orientation feature has the best performance for vision-based gesture recognition. So we get the information that orientation is the most important information from a trajectory curve. Although the acceleration isn't integrated twice into location, we could still draw a trajectory curve by acceleration information directly. Therefore, in this section, we regard the sequence of acceleration vectors as a serial of points in 2D coordinate space, and an acceleration trajectory curve is got by connecting the points. The x-axis is the acceleration in x-direction and the y-axis is for y-direction. In this way, figure 4 is redrawn into figure 5 by changing the meanings of abscissa and ordinate.

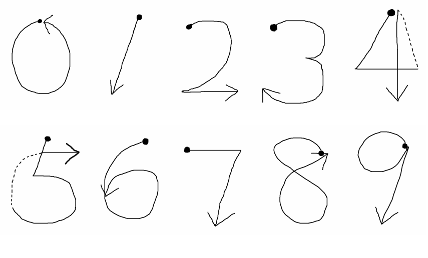
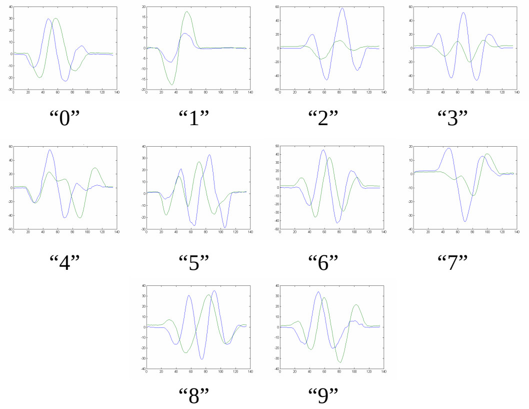


Figure 3. Trajectories of Arabic numerals

To extract the orientation information, it is necessary to divide the 2D coordinate space into some parts. There are theoretically quite a large number of ways to finish this task; however, there exist some constraints at the same time as follows. First, the space couldn't be divided into too many parts, because the sampling rate of accelerometer is a constant value, and if so, the quantization would be imprecise. Second, the number of divisions can't be too small, or the acceleration information would be largely lost at this step. Third, the method to divide the space must reflect the features of curves. For example, as figure 5 shows, there are many circles in the curves, so we should use the angle value instead of the distance in quantization method, if all the points are put into polar coordinates system. Finally, considering the range of accelerometer, all points in these trajectories must be in a rectangle, so the task is to find out a way to divide this rectangle. Figure 6 represents two methods of quantization, 4-division and 6- division, based on orientation in polar coordinates system. When the trajectory curve passes through a division, it would produce one digit (without iteration) or a serial of one digit (with iteration) and details of quantization methods will be presented in section 5.

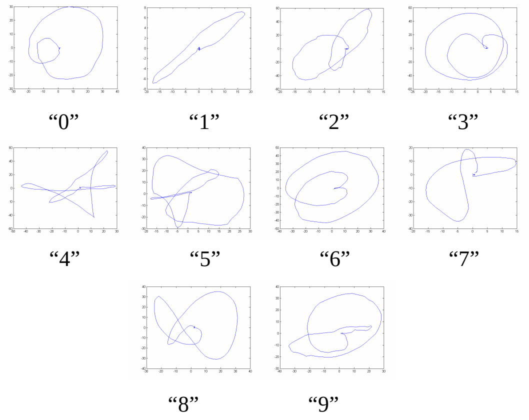


"0" "1" "2" "3"

"4" "5" "6" "7"

"8" "9"

Figure 4. Acceleration curves in x-axis and y-axis by time



"0" "1" "2" "3"

"4" "5" "6" "7"

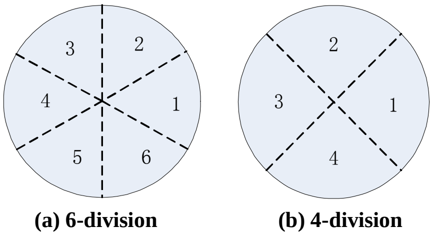
"8" "9"

Figure 5. Acceleration trajectory curve

In this way, for every handwritten character we could extract a serial of numbers according to the acceleration trajectory. The numbers contain the angle information of the handwritten

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3898
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3898

character in polar coordinates. In following parts, these two kinds of quantization would be compared. Besides, it is noted that there are many repetitions for the numbers after quantization, so we would also prove that the repetitions have much important information and are helpful for recognition.



(a) 6-division **(b) 4-division**

Figure 6. Quantization in polar coordinates system

4.4 Classification Based on Discrete HMM

Discrete HMM (DHMM) is a lightweight solution for classification compared with Continuous HMM (CHMM), DWT and other methods. In this part, we use software, Discrete Hidden Markov Model Simulation (HMMSim) to simplify the calculation. In [8], the author of this tool presents the incentive and objective of writing it and gives an introduction about the operation.

4.4.1 Hidden Markov Models

Markov model is a mathematical model of stochastic process where these processes generate a random sequence of outcomes according to certain probabilities [10], [11]. It is trainable and the underlying stochastic process is unobservable, so we call it hidden markov model. Both CHMM and DHMM are widely used in Pattern Recognition and many other fields. For DHMM, the observable states and invisible states are all discrete.

A HMM is a collection of finite states $S = \{S_1, S_2 \dots S_N\}$ interconnected by transitions. Each state has a number of distinct observation symbols $V = \{v_1, v_2 \dots v_M\}$ corresponding to the physical output of the system [2]. A HMM can be specified by the following notation:

$$\lambda = (A, B, \pi) \tag{7}$$

where, the state transition probability distribution $A = \{a_{ij}\}$, and a_{ij} denotes the probability of transition from S_i to S_j. The observation symbol probability distribution at the given state S_i, $B = \{b_i(x)\}$. The initial state probability distribution vector $\pi = \{\pi_i\}$.

A is an N by N matrix, B is an N by M one and π is a 1 by N vector. All of them should be initialized by experience at first. Number N represents the number of states in the model and number M is of distinct observation symbols per state. In [5], HMM are divided into three topologies: Fully Connected (Ergodic) model, Left-Right model and Left-Right Banded model. In this paper, we use the first one, both N and M are set to 10 and all the units in A, B and π are set to 0.1. Although these values are imprecise initially, they would keep approaching a perfect outcome in the process of numerous iterative calculations.

4.4.2 Classification

There are three problems for HMM to solve.

- First, consider the problem where we have a number of trained HMMs describing different systems, and a sequence of observations. The problem is to find which HMM most probably generated the given sequence, or calculate out the probability of every HMM for the observations.
- The second problem is to find the most probable sequence of hidden states given some observations.
- The last one is to generate a HMM from a sequence of observations.

By HMMSim, we could find the parameters of the HMM which maximize the probability to happen upon certain sequences of observations (*HMM Problem 3*) and estimate the probability to happen upon a particular sequence of observations (*HMM Problem 1*). Figure 7 is the flow chart of HMM training and testing in detail.

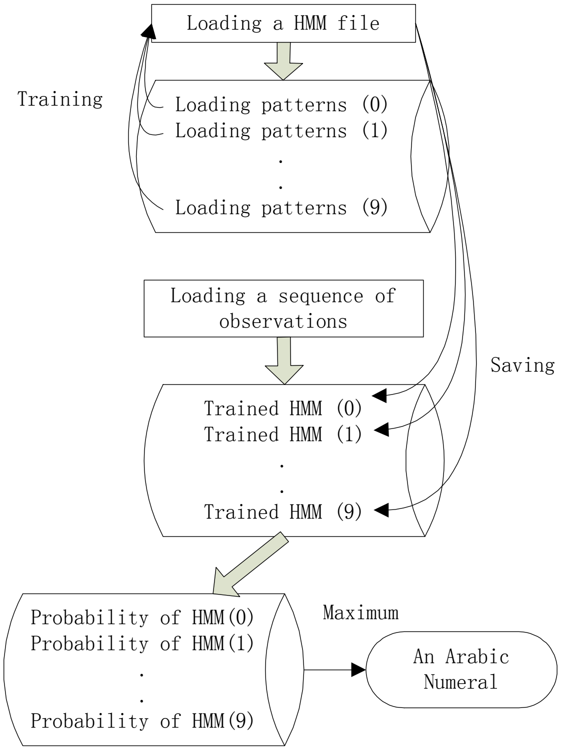


Figure 7. Flow chart of classification by HMM

At the beginning, a HMM file is loaded, which contains matrix A, B and vector π in it. This is both an initialization file and the basis of all the iterations. Then for every Arabic numeral, there are several patterns sampled already to train HMMs. After the step of training and saving, we get ten trained HMMs. For every trained model, using Forward Algorithm (FA) we can calculate out a probability value of an observation sequence and this is just the HMM problem 1. The final step is to compare all the values of probability and the Arabic numeral corresponding to the biggest value is the outcome of classification.

5. EXPERIMENTAL RESULTS

To test the performance of this system, large numbers of experiments have been carried out. Most efforts are made to

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3898
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3898

evaluate different kinds of quantization based on orientation. The methods of quantization are implemented in Matlab Language and because both Matlab and HMMSim could convert programs into C++ language, it is convenient to compile this system into executable file on Windows.

For every Arabic numeral, we record 42 times of input motion samples as figure 3 shows. Black points represent the beginning locations and the arrows represent the end locations and the directions of trajectories. The dashed lines mean that for paper-based handwritten characters they should be skipped over, but for acceleration-based ones they are recorded all together with the other strokes and have effect on the results of recognition too. For the characters having more than one stroke, it is difficult to separate a trajectory into several single parts, because detecting a beginning or an end point during a motion without any halt is not an easy task and the error-detected would have great bad effect on the recognition rate too. The ways to draw these Arabic numerals in the air are all the same as the ordinary way. Compared with the self-defined gesture shapes [1], [13], although some trajectories are complex and would add extra difficulties in recognition, they are more user-friendly and easier to be accepted. Half of the samples are used to train HMMs and the other half are used to test the performance. To compare the two methods of quantization in section 3.3, we use both of them in experiments.

At first, we choose the simplest quantization method, which is 4-division without iteration. The reason to discard the iterations is to test whether they are helpful to HMM recognition and contains useful information or not. Quantization of 4-division has only four feature numbers and the calculation would be much simpler than 6-division one, so the first step is to test the performance of the 4-division without iteration and the results are shown in table 1. Then as table 2 shows, the test results by quantization of 6-division without iteration are presented. Finally, test results of the most complex quantization way are shown in table 3. In the tables, the rows marked "none" mean that the probability values of all the ten numerals are zero and the input can't be recognized by any one of all the trained HMMs.

In table 1, we can see that for "0" and "4" more than half the outcomes are incorrect, and moreover "0" and "6" are almost undistinguishable. In this case, the numerals with complex trajectories are difficult to recognize because the number of quantizing classification is too small to extract enough information for recognition. In table 2, the problem is that both "2" and "3" have high rates to be incorrectly recognized as "0". By the way, from some point of view, being unable to recognize a numeral is better than an incorrect recognition, because the latter may result in incorrect output without the user noticing it.

Both Table 1 and Table 2 prove that the quantization of 4-division loses too much information that some digits couldn't be told apart from each other. Considering the drawbacks of the above two test results, we implement the quantization of 6-division with iteration. As table 3 shows, the test results have already got great improvement and to a certain extent the error-detected are homogeneously distributed among the ten numerals.

In summary, we draw a histogram as figure 8, which presents every Arabic numeral's recognition rate of all the three quantization methods. The recognition rates of 4-div., 6-div. without iteration and 6-div. with iteration are 77.14%, 84.76% and 94.29%.

Table 1. Test results by quantization of 4-division without iteration

In Out	0	1	2	3	4	5	6	7	8	9
0	11	0	0	0	6	0	9	0	0	0
1	0	16	0	0	0	0	0	0	0	0
2	0	0	21	0	0	0	0	2	0	0
3	0	0	0	21	0	0	0	0	0	0
4	0	0	0	0	9	0	0	0	0	0
5	0	0	0	0	0	16	0	0	0	0
6	19	1	0	0	5	0	12	0	0	0
7	0	1	0	0	0	0	0	17	0	0
8	0	1	0	0	0	0	0	0	18	0
9	0	0	0	0	0	4	0	0	3	21
None	1	2	0	0	1	1	0	2	0	0

Table 2. Test results by quantization of 6-division without iteration

In Out	0	1	2	3	4	5	6	7	8	9
0	20	0	7	11	0	0	0	0	0	0
1	0	19	0	0	0	0	0	0	0	0
2	0	0	12	0	0	0	0	0	0	0
3	1	0	2	10	0	0	0	0	0	3
4	0	1	0	0	20	0	1	0	0	0
5	0	1	0	0	0	20	0	0	0	0
6	0	0	0	0	0	0	18	0	0	0
7	0	0	0	0	0	0	0	20	0	0
8	0	0	0	0	0	0	0	0	21	0
9	0	0	0	0	0	0	2	0	0	18
None	0	0	0	0	1	1	0	1	0	0

6. CONCLUSIONS AND FUTURE WORK

In this paper, we present a system to recognize handwritten Arabic numerals using orientation quantization. This is an online system and the recognition is based on acceleration information instead of trajectory. Acceleration signals are sampled by a high performance 3-dimensional accelerometer from Freescale Semiconductor. In algorithm aspect, three kinds of quantization based on orientation are compared by a large number of experiments and the quantized acceleration data are recognized by Discrete HMM. From the test results, a conclusion is drawn that the quantization method of 6-division with iteration is better than the other two and is proved to be feasible in practice. Moreover, the quantization method introduced in this paper is not complex and it can be optimized to

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3898
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3898

get a higher recognition rate in succession.

This paper also proves that mobile and handheld devices could run without any number key (including physical keys and the ones on touch-screen which detect pressure from pens or fingers); in this way large space could be saved for their small size. If this system is combined with some other gesture recognition technologies like “swaying” or “inclination”, which could be defined as “Enter”, “Quit” and so on, a keyless mobile phone or PDA could be realized. And if the character recognition could be developed into sign language recognition without too many sensors, it would be greatly helpful for normal person to understand sign language. Moreover, if sign language recognition could be combined with speech synthesis technology successfully, maybe the deaf and dumb could “speak” someday like normal human beings.

Table 3. Test results by quantization of 6-division with iteration

In Out	0	1	2	3	4	5	6	7	8	9
0	21	0	0	0	0	0	0	0	0	0
1	0	19	0	0	0	0	0	0	0	0
2	0	0	20	0	0	0	0	0	0	0
3	0	1	1	21	0	0	0	0	0	0
4	0	1	0	0	19	0	0	0	0	1
5	0	0	0	0	0	19	0	0	0	0
6	0	0	0	0	0	0	20	0	0	0
7	0	0	0	0	0	0	0	21	0	0
8	0	0	0	0	0	0	0	0	21	0
9	0	0	0	0	0	0	0	0	0	17
None	0	0	0	0	2	2	1	0	0	3

In the future work, we would try to recognize simple sign language samples and introduce more sensors like gyroscope and temperature sensors to recognize human daily activities and health condition. Besides, another goal is to reduce the number of sensors and get a balance between system performance and hardware requirement.

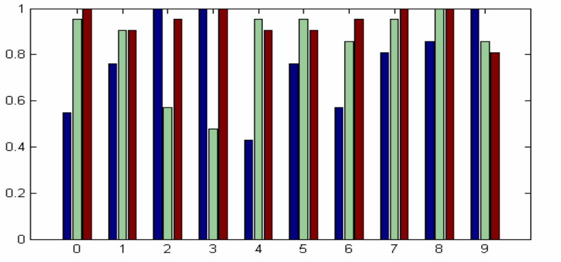


Figure 8. Histogram of Recognition Rates
From left to right, blue, green and red represent 4-div. without ite., 6-div. without ite. and 6-div with ite.

7. REFERENCES

[1] Sung-Jung Cho, Jong Koo Oh and Won-Chul Bang, “Magic Wand: A Hand-Drawn Gesture Input Device in 3-D Space with Inertial Sensors”, Proceedings of the 9th Int’l Workshop on Frontiers in Handwriting Recognition, 2004

[2] Meng Chen, Bufu Huang and Yangsheng Xu, “Human Abnormal Gait Modeling via Hidden Markov Model”, Proceedings of the 2007 International Conference on Information Acquisition, Jeju City, Korea, July 9-11, 2007

[3] Sung-Do Choi and Alexander S. Lee, “On-Line Handwritten Character Recognition with 3D Accelerometer”, Proceedings of the 2006 IEEE International Conference on Information Acquisition, Weihai, Shandong, China, August 20-23 2006

[4] Eun-Seok Choi, Won-Chul Bang, “Beatbox Music Phone: Gesture-based Interactive Mobile Phone using a Tri-axis Accelerometer”, IEEE International Conference on Industrial Technology, pp. 97-102, Dec. 14-17, 2005

[5] Mahmoud Elmezain and Ayoub Al-Hamadi, “Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models”, IEEE Int. Symposium on Signal Processing and Information Technology, 2007

[6] Jian Kang Wu, Liang Dong, and Wendong Xiao, “Real-Time Physical Activity Classification and Tracking using Wearable Sensors”, 2007 6th Int. Conference on Infor., Comm. and Signal Processing, Singapore, pp. 1-6, Dec. 2007

[7] Jin He, Huaming Li and Jindong Tan, “Real-time Daily Activity Classification with Wireless Sensor Networks using Hidden Markov Model”, Proceedings of the 29th Annual International Conference of the IEEE EMBS Cite International, Lyon, France, August 23-26, 2007

[8] J. F. Vélez, A. Sánchez and A. B. Moreno, “HMMSim: An Interactive Simulation Tool for Teaching Discrete Hidden Markov Models”, 4th International Symposium on Computer Science Education, Vigo, Spain, November 22, 2002

[9] Duy Bui and Long Thang Nguyen, “Recognizing Postures in Vietnamese Sign Language With MEMS Accelerometers”, IEEE Sensors Journal, Vol. 7, NO. 5, May 2007

[10] Sushmita Mitra and Tinku Acharya, “Gesture Recognition: A Survey”, IEEE Transactions on System, Man, and Cybernetics-Part C: Applicatoins and Reviews, Vol. 37, NO. 3, May 2007

[11] N. Liu, B. C. Lovell and P. J. Kootsookos “Model Structure Selection & Training Algorithms for a HMM Gesture Recognition System”, International Workshop in Frontiers of Handwriting Recognition, pp. 100-106, October 2004

[12] Guanglie Zhang and Guangyi Shi, “Towards an Ubiquitous Wireless Digital Writing Instrument Using MEMS Motion Sensing Technology”, Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, USA, 24-28 July, 2005

[13] Sanna Kallio, Huha Kela and Jani Mantyjarvi, “Online Gesture Recognition System for Mobile Interaction”, IEEE International Conference on System, Man and Cybernetics, pp. 2070-2076 vol.3, 5-8 Oct. 2003

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3898
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3898