Magic Wand: A Hand-Drawn Gesture Input Device in 3-D Space with Inertial Sensors

Sung-Jung Cho, Jong Koo Oh, Won-Chul Bang, Wook Chang,
Eunseok Choi, Yang Jing, Joonkee Cho, Dong Yoon Kim

*{sung-jung.cho, jong.oh, wc.bang, wook.chang, eunseok.choi,
jing.yang, handle.cho, kdy2891}@samsung.com
Samsung Advanced Institute of Technology
PO Box 111, Suwon, 440-600, Korea*

Abstract

*This paper presents a gesture input device, **Magic Wand**, with which a user can input gestures in 3-D space. Inertial sensors embedded in it generate acceleration and angular velocity signals according to a user's hand movement. A trajectory estimation algorithm is employed to convert them into a trajectory on 2-D plane. The recognition algorithm based on Bayesian networks finds the gesture model with the maximum likelihood from it. The recognition performance of the proposed system is quite promising; the writer-independent recognition rate was 99.2% on average for the database of 15 writers and 13 gesture classes.*

Keywords: Gesture input device, Handwriting recognition, Bayesian networks, Trajectory estimation, Inertial navigation system, Accelerometer, Gyroscope

1. Introduction

As the ubiquitous computing environment becomes widespread these days, the role of computers has changed from massive computing devices to assistant devices of our daily lives. Accordingly, more natural interaction methods beyond tradition keyboards and mouse have been studied. Speech recognition [1], vision-based gesture recognition [2] and on-line handwriting recognition [3] are popular examples. Among them, the online handwriting input method, which transcribes human hand movements into characters and gestures, has the advantage of natural and portable interaction. It is very natural because people have been accustomed to using pens and papers since childhood. Also a small writing surface of a few inches is enough for applying it.

Conventional on-line handwriting recognition systems aim at recognizing trajectories written on 2-D writing surfaces. The writing surface is enabled usually by tablet types of devices such as opaque tablets, touch pads, tablet PCs and web pads. When people write characters and gestures by these devices, pen positions on 2-D plane are digitized by sensing pressures [4] or electro-magnetic signals of tablets [5]. These devices have the advantage of high resolution and high sampling rate in digitization. However, they have the limitation that people should write only on tablets.

In order to extend the writing area beyond tablets, new types of sensors are employed such as optical sensors and ultrasonic waves. In the case of optical sensors, a camera mounted on the pen tip captures the image around it. Then the coordinate of the pen tip is calculated by analyzing unique image patterns [6] or comparing changes between consecutive input images [7]. In the case of ultrasonic waves, a pen emits ultrasonic waves and receivers in neighborhood compute distances from it [8]. Due to these sensors, writing surfaces are extended to any plat surfaces.

By employing inertial sensors, the writing area is further extended to 3-D space. Inertial sensors measure the inertia of objects; accelerometers measure accelerations and gyroscopes measure angular velocities. They are small enough to be embedded in handheld information devices. Moreover, they do not require any external reference devices. Therefore, users can input gestures in almost any place. This is a big advantage over the approach of employing external sensors such as ultrasonic receivers/emitters [9] and cameras [10], which require fixed installation space. However, they have the limitation that trajectories are not measured directly but calculated from sensor signals so that robust signal processing techniques are necessary.

This paper presents a gesture input device, *Magic Wand*, that recognizes trajectories of hand movements in 3-D space. When a user writes gestures in 3-D space with the wand, its inertial sensors, 3-axis accelerometers and 3-axis gyroscopes, convert hand movements into acceleration and angular velocity signals. Then, a trajectory estimation algorithm converts them into trajectories. Finally, a recognition algorithm matches the trajectories with Bayesian network-based gesture models.

2. System overview

2.1 System usage

In order to show the applicability of a 3-D input device in commercial products, we have made a prototype remote controller with only one button. Typical remote controllers have tens of buttons. Different commands are mapped to different buttons so that users control electronic appliances such as TV's and DVD players by pressing buttons. Many customers complain the difficulty to find the proper button among so many small buttons. However, the proposed system has only one button which activates sensor signals. Users control appliances by drawing gestures while pressing the activation button. Fig. 1 shows the picture of the proposed system. It looks noticeably simple and slim compared to conventional large remote controllers with a lot of buttons.

The typical scenario of its usage is as follows. When a user wants to issue a command, he draws the gesture shape corresponding to it while pressing the activation button. After the button is released, the gesture shape is recognized. Then, proper IR control codes are fetched and transmitted to a TV via its IR LED. By using the traditional IR codes, conventional TVs are controlled by gestures without any modification to them.

Fig. 1: *Magic Wand*: the proposed gesture input device as a form of a remote controller

2.2 Hardware components

The *Magic Wand* consists of accelerometers (1 XZ-axis, and 1 Y-axis chips), gyroscopes (1 X-, 1 Y-, 1 Z-axis chips), an analog-to-digital converter (ADC), a digital signal processor (DSP), a flash memory, an infrared (IR) LED, a serial port interface and a lithium-iron battery (Fig. 2). Accelerometers measure accelerations and gyroscopes measure angular velocity. The measured signals are converted into digital signals by ADC. DSP runs signal processing and recognition algorithms. Flash memory stores program codes and data. A serial port interface is used for transmitting sensor data to PC when collecting data and training gesture recognizers.

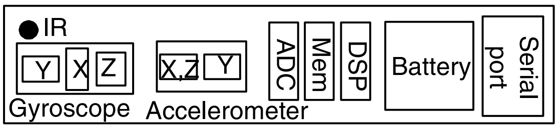

Fig. 2: Hardware components of *Magic Wand*

2.3 Software components

The gesture recognition system has the components of a trajectory estimation algorithm and a gesture recognition algorithm. The trajectory estimation algorithm gets the acceleration and angular velocity signals from sensors, and converts them into a hand-movement trajectory. The gesture recognition algorithm gets the trajectory and classifies it into one of predefined gesture classes. Fig. 3 shows the overall software components and data flow.

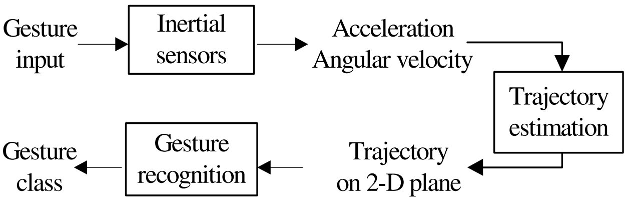

Fig 3: Software components of *Magic Wand*

We choose to classify trajectories instead of raw signals because of two reasons. First, signal variations can be greatly reduced in the trajectory domain compared to in the raw signal domain. Inertial sensor signals are sensitive to the variation of motion status and writers. Even a simple trajectory can be made by different ways of movements; some people tend to write slowly and other fast. Also the posture of the device, how it is oriented in the 3-D space, affects sensor signals. In trajectory domain, these motion and posture variations are removed. Second, in trajectory domain, we can apply traditional on-line handwriting recognition algorithms. They have been studied for decades so that reliable recognition performance is expected, provided that trajectories can be stably estimated.

3. Trajectory estimation algorithm

The trajectory estimation algorithm converts raw sensor signals into trajectories in 3-D space and finally projects it onto 2-D plane [11-13]. The first step to realize the proposed system is to identify physical motion properties, i.e., three dimensional position and orientation information, which falls into category of motion tracking. There are various kinds of motion tracking algorithms available and the proposed system utilizes the motion tracking method with inertial sensing technologies.

With three axis acceleration and angular rate measurements, the theory of inertial navigation system (INS) theoretically guarantees the possibility of computing position and orientation information of an object moving in the 3-D space [11].

To apply an INS theory, we defined the coordinate system of the body coordinate (**b**) and the navigation coordinate (**n**) in 3D space, as shown in Fig. 4. The navigation coordinate (**n)** is the stating point of trajectory external to the input device. x_n , y_n , and z_n axes are perpendicular to one another, where the direction of z_n is parallel to the direction of earth gravity (**g**). It does not change even when the device is in motion. The body coordinate (**b**) is fixed on the device. It is aligned with the axes of the inertial sensor chips (IMU: inertial measurement unit). x_b, y_b, and z_b axes are perpendicular to each other, where the direction of z_b axis is aligned with the core axis of the device. Therefore, it is changed according to the motion of the device body.

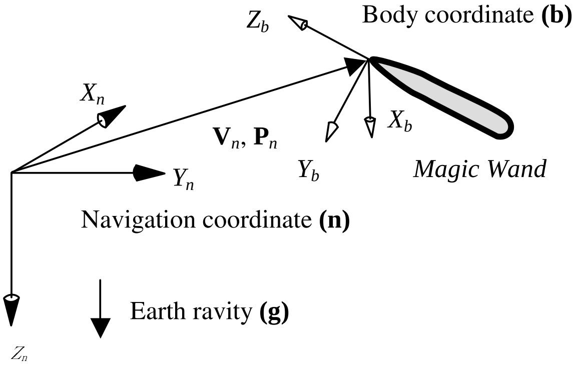

Fig. 4: Coordinate system: the body coordinate (b) and the navigation coordinate (n)

While a user is drawing a gesture, the IMU measures the acceleration $A_b = [A_{bx} \quad A_{by} \quad A_{bz}]^T$ and angular rate $\omega_b = [\omega_{bx} \quad \omega_{by} \quad \omega_{bz}]^T$ of each axis in the body coordinate (**b**). Then, using A_b and ω_b , the acceleration $A_n = [A_{nx} \quad A_{ny} \quad A_{nz}]^T$ in the navigation coordinate (**n**) is calculated by the state-space equation [11] of the device. By integrating A_n twice, we obtain the handwritten trajectory $P_n = [P_{nx} \quad P_{ny} \quad P_{nz}]^T$ in the navigation coordinate (**n**). The governing equations of motion tracking used in this paper are expressed as follows:

$$\begin{aligned}
\dot{P}_n &= V_n \\
\dot{V}_n &= C_b^n A_b - G \\
\dot{\theta} &= \omega_{by}\cos\phi - \omega_{bz}\sin\phi \\
\dot{\psi} &= \frac{\omega_{by}\sin\phi + \omega_{bz}\cos\phi}{\cos\theta} \qquad (1) \\
\dot{\phi} &= \omega_{bx} + (\omega_{by}\sin\phi + \omega_{bz}\cos\phi)\tan\theta
\end{aligned}$$

where the subscript n denotes the navigation coordinate, and b denotes the body coordinate, V_n is the rate of change of position, i.e., velocity shown in the navigation coordinate, G is the constant gravity vector shown in the navigation coordinate, $(\omega_{bx}, \omega_{by}, \omega_{bz})$ is the inertial angular rate vector shown in the body coordinate, $(\phi, \theta, \psi) = (\text{roll, pitch, yaw})$ are Euler angles. Here, the matrix $C_b^n = C_n^{bT}$ refers to the direction cosine matrix describing the rotation relationship between the navigation coordinate and body coordinate and is the function of Euler angles as shown in Eq. (2).

However, the above described algorithm can not be directly applied to the proposed system since the INS leads to an unbounded growth of error due to many integration steps involved. A typical INS uses periodic or aperiodic resetting procedure to remove the error growth, which is not a feasible solution for the small and low-cost systems including the proposed system. Fortunately, the solution of this problem has been detailed and solved in [12] and is called zero velocity compensation (ZVC).

After reconstructing 3D motion information, we project the recovered 3D trajectory into a 2D writing plane by finding the optimal writing plane in the sense of minimum distortion to the original point positions [12]. The purpose of this process is to reduce the writing plane variation of the estimated trajectories.

4. Gesture recognition algorithm

We apply the on-line handwriting recognizer based on Bayesian networks [14-15] for recognizing trajectories estimated from inertial sensor signals. The recognizer models dependencies between points and basic strokes explicitly. It showed favorably comparable recognition rates to conventional approaches based on template matching method and hidden Markov models in recognizing digits and Korean Hangul characters [14-15].

4.1 Introduction to Bayesian networks

A Bayesian network is a directed acyclic graph (DAG) whose nodes represent random variables and whose arcs relationships between them [16]. It efficiently encodes the joint probability distribution of a large set of random variables. When a Bayesian network S has N variables: $X_1, X_2, \cdots, X_N$ and $pa(X_i)$ denotes the random variables from which dependency arcs come to X_i , the joint probability of $X_1, X_2, \cdots, X_N$ is given as follows:

$$P(X_1, X_2, \cdots, X_N) = \prod_{i=1}^{N} P(X_i \mid pa(X_i)) . \qquad (3)$$

In this paper, the conditional probability is represented by the conditional Gaussian probability [14]. When a multivariate random variable X depends on $X_1, \cdots, X_n$, the conditional probability distribution is given as follows:

$$C_n^b = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & \cos(\theta)\sin(\phi) \\ \sin(\psi)\sin(\theta) + \cos(\psi)\sin(\theta)\cos(\phi) & -\cos(\psi)\sin(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \qquad (2)$$

$$P(X = x | X_1 = x_1, \cdots, X_n = x_n) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp[-\frac{1}{2}[x-u]^T \Sigma^{-1}[x-u]] \quad (4)$$

The mean μ is determined from the linear weight sum of dependant variable values $Z = [x_1^T, \cdots, x_n^T, 1]$ as follows:

$$u = WZ^T. \quad (5)$$

where W is a $d \times k$ linear regression matrix, d and k are the dimension of X and Z^T respectively.

4.2 Gesture model

A gesture is represented hierarchically by modeling its components and relationships among the components [14]. In the first level, a gesture model is composed of basic stroke models and their relationships. In this paper, a basic stroke denotes a nearly straight trace whose global direction is different from those of connected traces in writing order. In the second level, a basic stroke model is composed of point models and their relationships. Finally, a point is modeled with 2-D Gaussian distribution for its X-Y position.

A point model is represented by a 2-D Gaussian distribution for (x, y) coordinates of its corresponding points in 2-D plane. It corresponds to a single node in Bayesian networks.

A basic stroke model is composed of point models and their relationships, called as WSRs (within-stroke relationships). It is constructed by recursively adding mid point models and specifying WSRs. A mid point is the point at which the lengths of the left and the right partial strokes are equal. A WSR is represented as the dependency of a mid point from two end points of a stroke. Fig. 5 shows the recursive construction example of a basic stroke model. Fig. 5 (a) shows an example of basic stroke instances. At the first recursion ($d = 1$), IP_1 is added for modeling ip_1's with the WSR from EP_0 and EP_1 (Fig. 5 (b)). At $d = 2$, IP_2 and IP_3 are added for the left and the right partial basic strokes (Fig. 5 (c)). This recursion stops when the covariances of newly added point models become smaller than a predetermined threshold.

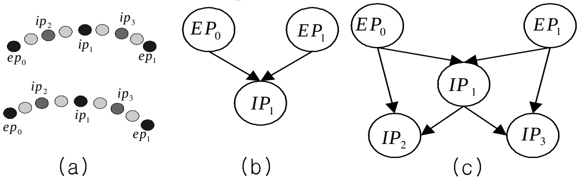

Fig. 5: Example of recursive construction of a basic stroke model

A gesture model is constructed by concatenating basic stroke models according to their writing order and specifying inter-stroke relationships (ISRs). ISRs are represented by dependencies among basic stroke end points. Fig. 6 shows a Bayesian network based gesture model with N strokes and the stroke recursion depth $d = 2$. EP_i's are the stroke end point models and $IP_{i,j}$'s are the internal point models of the i-th basic stroke. The right end point of the previous basic stroke is shared with the left one of the following basic stroke. ISRs are represented by the arcs between EP_i's, and WSRs are represented by the incoming arcs to $IP_{i,j}$'s.

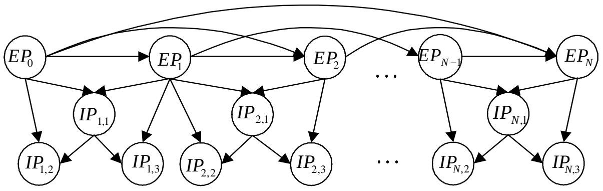

Fig. 6: Gesture model with N basic strokes and the stroke recursion depth of 2

4.3 Matching algorithm

Each gesture class m has a corresponding gesture model λ_m [14]. A gesture input, a trajectory point sequence of $O_1, \cdots, O_T$, is recognized by finding the gesture model λ^* which produces the highest model likelihood as follows:

$$\begin{aligned} \lambda^* &= \arg\max_m P(\lambda_m | O_1, \cdots, O_T) \\ &= \arg\max_m P(\lambda_m) P(O_1, \cdots, O_T | \lambda_m) \end{aligned} . \quad (6)$$

The model likelihood is calculated by matching stroke internal point models (IP's) and stroke end point models (EP's) of gesture models (Fig. 6) with the input point sequence. Because boundaries of basic strokes are not explicitly specified in the input point sequence, all the possible basic stroke segmentations should be searched. After a gesture input is segmented into basic strokes, basic stroke end points are matched to EP's. Then each basic stroke is recursively resampled into mid points and matched to IP's. When a gesture model G with N basic stroke models matches the input points $O_1, \cdots, O_T$, and one basic stroke segmentation instance is denoted as $\gamma_i = (t_0, t_1, \cdots, t_N)$, $t_0 = 1, \cdots, t_N = T$ and the whole set as Γ, the model likelihood is calculated as follows:

$$\begin{aligned} &P(O_1, \cdots, O_T | \lambda_m) \\ &= \sum_{\gamma_i \in \Gamma} \prod_{j=0}^{N} P(EP_j = O(t_j) | O(t_0), \cdots, O(t_{j-1})) \\ &\quad \prod_{j=1}^{N} \prod_{k=1}^{2^d-1} P(IP_{j,k} = ip_{j,k}(O(t_{j-1}, t_j)) | pa(IP_{j,k})) \end{aligned} \quad (7)$$

0-7695-2187-8/04 $20.00 © 2004 IEEE

In Eq. (7), $O(t_{j-1}, t_j) = \left(O(t_{j-1}), O(t_{j-1}+1), \cdots, O(t_j)\right)$ and $ip_{j,k}$ represents the k -th recursively sampled point of the j -th basic stroke input. The matching probabilities of EP 's can be interpreted as the probabilities of global stroke positions and those of IP 's as the probabilities of local stroke shape distortions.

5. Experimetal results

5.1 Data set

In order to evaluate the proposed gesture recognition system, we collected data from 15 writers. Among them, eight writers have some experience of using the device and the others do not have any. The input device was attached to a PC by using the serial port interface during data collection. Sensor signals were generated from the input device and transmitted and saved in the PC. Gesture labels and representative shapes were shown on PC screen. Writers drew gestures while looking at the representative shapes. Each writer wrote 13 classes of gestures by 24 times. They were instructed to hold the device in static position for a short time just before and just after writing.

Gesture shapes are iteratively designed for high recognition rates and convenience of writing. At first, we adopt gesture shapes from widely used graffiti on PDAs. Because of the limited accuracy of trajectory estimation algorithm and the lack of visual feedback to writers during writing, gestures with similar movement history were confused frequently (the gesture pair of 0 and 6, and 5 and 8). These confusions are resolved by appending a basic stroke to the end of the gesture 5 and 6. Fig. 7 shows the final gesture shapes[1].

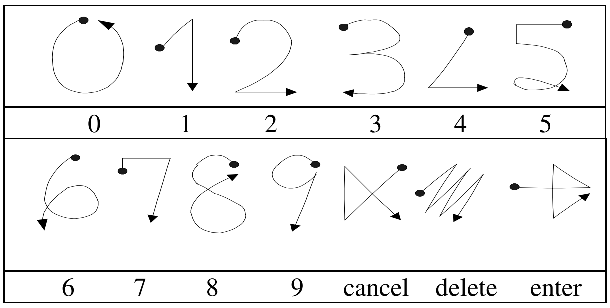

Fig. 7: Gesture shapes for experiments.

Fig. 8 shows an example of inertial sensor signals obtained when the gesture 2 is written. The first graph shows 3-axis acceleration signals and the other graph shows 3-axis angular velocity signals. The large signal changes from the time 25 to the time 75 suggests that the gesture was drawn during the interval. The other intervals indicate that the device was in a static position. It is observed that the acceleration is more sensitive to the motion of the device than angular velocity from its larger variation.

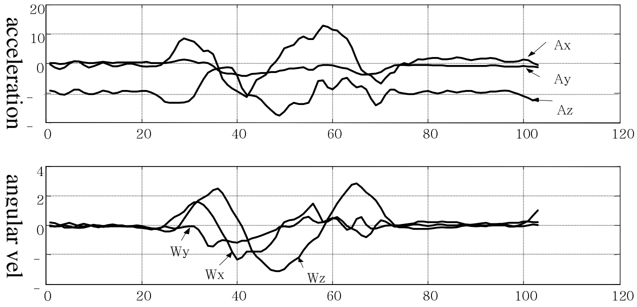

Fig. 8: Example of raw sensor signals when the gesture 2 is written

5.2 Results of trajectory estimation

Fig. 9 shows trajectories estimated from raw sensor signals. Texts on the left-top corner of trajectories represent guesture labels. The shapes look somewhat distorted from representative gesture shapes. Artificial hooks are observed in start and end parts of trajectories. The length ratios between basic strokes are not estimated reliably. It is caused by integration errors of inertial signals and also the lack of visual feedback of trajectories to writers. Nontherless, trajectory shapes look smooth and natural, and directions of partial trajectories are estimated reliably. Also, shapes of different classes look distinguishable among one another.

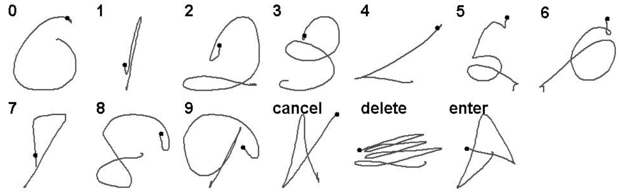

Fig. 9: Examples of trajectories estimated from raw sensor signals

5.3 Recognition results

The recognition performance was measured by dividing the data set of 15 writers into four partitions according to writers. First, the first three subsets were used for training and the other for testing. Second, the next three subsets were used for training and the other for testing. In this way, four different configurations of training and test sets were used for evaluating the writer-independent recognition rate.

Fig. 10 and 11 show recognition rates by writers and by classes. The average recognition rate of all the writers is 99.2%, and all the writers have recognition rates of more than 96%. However, large variations are shown among writers. Among gesture classes, the class 7 has the

[1] The shapes shown in this paper are mainly designed for testing the recognition performance of the device. We have developed another gesture set suitable for TVs.

lowest recognition rate because its shape is similar to that of the class 1. When it is considered that half of the writers have no experience in writing with the proposed device, the high recognitionrate indicates the reliability of the proposed input device.

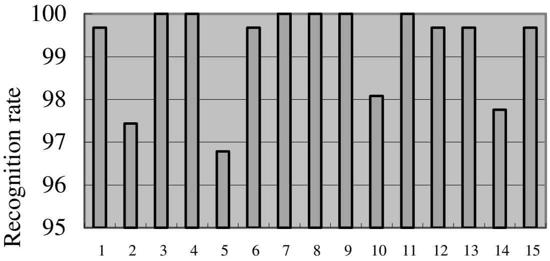

Fig. 10: Recognition rates by writers

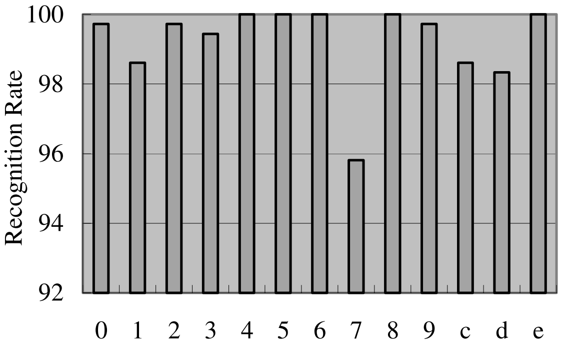

Fig. 11. Recognition rates by classes ('c': 'cancel', 'd': 'delete', and e: 'enter')

6. Conclusions

This paper introduces the gesture recognition system in 3-D space. The employment of inertial sensors enables users to draw gestures in almost any place because they do not require any external reference devices. In order to reduce the variations of movement histories and postures of the device, the trajectory estmiation algorithm based on inertial navigation system theory is employed to convert inertial signals to trajectories. Bayesian network based gesture recogntion algorithm is employed to recognize the estimated trajctories.

The proposed gesture recognition system showed a promising performance; the average recogntion rate of writer independent test was about 99.2% on the database of 15 writers and 13 classes of gestures. It is quite a promising result with the fact that the half of the writers have no experience with the device. The estimated trajectories look somewhat distorted from the original gesture shapes but shapes of different classes look distinguishable.

The future work is to enhance the preprocessing step such as hook removal of the estimated trajectories and to design gesture shapes more convenient to users.

Acknowledgement

Mr. Kyungho Kang implemented the proposed hardware prototype by designing PCB and mounting sensors. Samsung Software Center designed the aesthetic remote controller case.

References

[1] L. Comerford, et. al, "The IBM personal speech assistant," Proc. of IEEE ICASSP, Vol. 1, 2001, pp 7-11

[2] V. Pavlovic, et. al, "Visual interpretation of hand gestures for human-computer interaction: a review," IEEE PAMI, Vol. 19, No. 7, 1997, pp 677-695

[3] R. Plamondon, et. al, "Online and off-line handwriting recognition: a comprehensive survey," IEEE PAMI, Vol. 22 No. 1, 2000, pp 63 –84

[4] Mass Multimedia, Inc., "How does a touchscreen work," http://www.touchscreens.com/intro-anatomy.html

[5] Wacom Inc, "Technology," http://www.wacom-components.com/english/tech.asp

[6] Anoto Group, "The technologies behind Anoto functionality," http://www.anoto.com/?url=/technology/infrastructure/

[7] OTM technologies, "White paper on OTM technology," http://www.otmtech.com/

[8] Pegasus Technologies, "The PC NoteTaker," http://www.pc-notetaker.com

[9] VR Depot, "Logitech trackers," http://www.vrdepot.com/vrteclg.htm

[10] Vicon Motion Systems, "Vicon motion trackers for VR," http://www.vicon.com/Engineering/support/downloads /tech%20sheets/VMS066A_VR05.pdf

[11] G. Dissanayake, et. al, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," IEEE Trans. Robotics & Automation, vol. 17, Oct., 2001, pp. 731–747

[12] W.-C. Bang, et. al, "Self-contained spatial input device for wearable computers," in Proc., 7th IEEE Int. Symp. on Wearable Computers, 2003, pp. 26- 34.

[13] W. Chang, et. al, "A miniaturized attitude estimation system for a gesture-based input device with fuzzy logic approach," in Proc, 4th Int. Symp. on Advanced Intelligent Systems, Sep., 2003, pp. 616-619

[14] S.-J. Cho, et. al, "Bayesian network modeling of strokes and their relationships for on-line handwriting recognition," Pattern Recognition, V. 37, No. 2, 2004, pp 253-264

[15] S.-J. Cho, J.H. Kim, "Bayesian Network Modeling of Hangul Characters for On-line Handwritten Recognition," Seventh ICDAR, Aug., 2003, pp 207-211

[16] F. Jensen, An Introduction to Bayesian Networks, Springer, New York, 1996