MagiText : Around Device Magnetic Interaction for 3D Space Text Entry in Smartphone

Rajkumar Darbar, Debasis Samanta
IIT Kharagpur, West Bengal, India
rajdarbar.r@gmail.com, dsamanta@iitkgp.ac.in

***Abstract*—This paper presents MagiText that expands the text entry space beyond physical boundaries of a device, to overcome the problems of touch screen input. This approach uses the mobile phone's inbuilt magnetometer sensor to recognize 3D space handwritten character gestures. The key idea is to influence the magnetic sensor by writing character gestures (i.e. Graffiti or EdgeWrite) in front of the device using a properly shaped magnet taken in hand. The movement of this magnet changes the magnetic flux pattern around the device and it is sensed and registered by the magnetometer sensor. Then, we classify this flux pattern using Support Vector Machine (i.e. SVM) classifier. The experimental result shows that our prototype can achieve 81.2% accuracy with Graffiti, whereas EdgeWrite provides 89.4% accuracy.**

***Index Terms*—Interaction Technique, Mobile Devices, Text Input, User Interfaces, Signal Processing, Machine Learning.**

I. Introduction

Mobile and tangible devices such as smartphones, tablets, smartwatches, bracelet, and portable music players, have become a rich text input space due to its tremendous affordability, flexibility, accessibility, and portability. With the proliferation of electronic industry, these devices are becoming smaller in size with increasing computational capabilities day by day. The traditional way of entering text on these devices is virtual keyboard and speech. But, each modality has its limitations which act as a barrier to smooth and efficient text entry for these small display devices.

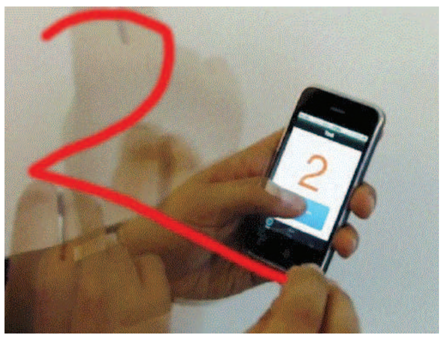
For example, typing with QWERTY softkeyboard suffers from fat finger problem and demands long visual search-time [4]. On the otherhand, speech based text input technique suffers from environmental noise and it demands server based architecture for automatic speech recognition [5].

To overcome the limitations of touch screen input, researchers proposed 3D space text entry technique that expands the text entry space beyond the physical boundaries of small screen handheld devices.

For this purpose, researchers have used phone's inertial sensors such as accelerometer, gyroscope, magnetometer and so on. For instance, in paper [1], S. Agrawal et al. proposed PhonePoint-Pen where a user holds the phone like a pen and writes short messages in the air (Fig. 1(a)). The acceleration due to hand gestures are translated into geometric strokes and compares the sequence of strokes against a grammar tree to identify the airwritten English alphabets. This process needs no training and recognizes characters with 83% accuracy. In an another research, T. Deselaers et al. [2] presented GyroPen that also allows users to hold the smartphone like a pen and write on any surface (Fig. 1(b)). The angular trajectory of the phone's corner is reconstructed from the phone's embedded gyroscope sensor and it is used as input to the online handwriting recognition system to identify the desired English words. In the experiment, novice participants took 37 seconds to write a word, whereas experienced users were able to write it within 3-4 seconds with a character error rate of 18%. However, both types of users feel that holding the phone as a pen is little bit awkward for writing long words. Recently, paper [3] proposed a stepwise lower-bounded dynamic time warping algorithm to recognize user-independent real-time 3D handwritten English alphabets using phone's built-in gyroscope sensor. This approach achieves 91.1% accuracy and it is computationally faster (in terms of memory and CPU time) than other conventional methods.

(a)

(b)



(c)

Fig. 1. (a) PhonePoint-Pen prototype [1] (b) GyroPen prototype [2] and (c) MagiWrite prototype [12].

The 3D space handwritten recognition methods presented in paper [1][2] and [3] considered phone as a pen. But, in our work, we used a different framework called Around Device Interaction (ADI) technique. This ADI commonly deals with different types of sensory inputs such as magnetic field [10], camera [6], electromagnetic field [9], and infrared distance sensor [7]. Among all these sensory inputs, magnetic field based ADI is much simpler as (1) its hardware (i.e. magnetometer sensor) is already available in the current mobile devices; (2) it consumes very less power; (3) it does not suffer from illumination variation and occlusion problems like camera based ADI; (4) it can pass through many materials i.e. it supports in-pocket interaction. Considering all these benefits, H. Ketabdar et al. [12] introduced MagiWrite, which supports 3D space digit (i.e. 0 - 9) entry in smartphones using magnetic field based ADI technique (Fig. 1(c)). In this approach, user draws digit shape gestures in front of the device using a properly shaped magnet taken in hand. This magnet movement changes temporal pattern of magnetic flux around the device and it is sensed and registered by the magnetometer sensor. Then, they applied Dynamic Time Warping (DTW) algorithm to recognize a particular digit from this unknown magnetic flux pattern. Similarly, here we propose MagiText which is an extension of this work [12] by looking at three different perspectives i.e. (1) it recognizes 26 English alphabets; (2) we compare the character recognition performance on the basis of Graffiti and EdgeWrite input gestures; (3) we have implemented the MagiText system architecture as an Android application and analyze the resource consumption profile in terms of CPU and memory usage.

II. MagiText Framework

The MagiText system architecture fundamentally comprises three parts: (1) input from magnetometer sensor (2) feature extraction from sensor data (3) modeling of multi-class Support Vector Machine (i.e. SVM) classifier. The working of the said three modules are discussed in the following subsections.

A. Input From Magnetometer Sensor

MagiText supports the effective use of 3D space around the device for handwritten English alphabets recognition. The underlying principle of our MagiText approach is to influence the phone's embedded magnetic sensor by drawing character gesture around the device along 3D trajectories using a proper shaped (may be ring, disk or rod type) magnet mounted on the finger. Here, we used Graffiti [14] (Fig. 2) and EdgeWrite [13] (Fig. 3) as character gesture and users have to press a button to indicate the starting and ending it. The temporal pattern of the magnetic flux's deformation is sensed by capturing the sensor values on x, y, and z coordinates. Note that, range of this output changes from device to device. For instance, in Samsung Galaxy S4, the value range is $\pm 200\mu$T, whereas, it is $\pm 128\mu$T for iPhone 3GS. The magnetic sensor can be affected by Earth's magnetic field and to eliminate this magnetic noise, we apply a time derivative operator on the output signals of the magnetic sensor.

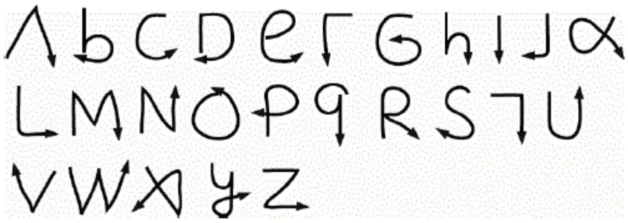


Fig. 2. Graffiti Characters [14].

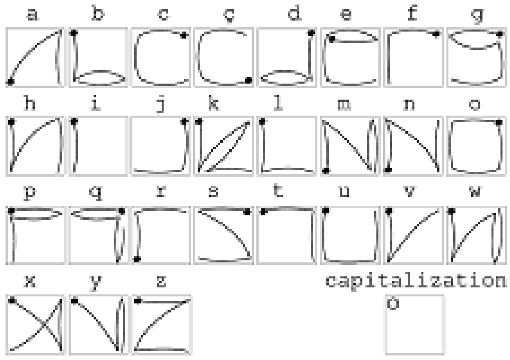


Fig. 3. EdgeWrite Characters [13].

B. Features Extraction

The next step in MagiText is feature extraction from recorded magnetometer signal. Here, we extract features over samples in an interval labeled with the starting and end of the gesture.

To capture the temporal pattern of the character gesture in a more detailed way, we divide the gesture interval into three equal length windows, extract a feature vector from each window and finally, concatenate three feature vectors to form a new feature vector that to be used for the gesture classification step. Features used in this study are listed in Table 1. These features yield 32 elements feature vector for each window and

TABLE I
Selected Features of Magnetometer Signal

Type	Feature Name	Coef.
Time Domain	Mean and S.D. of magnetic field strength along x, y, and z	6
	Mean and S.D. of Euclidian norm of magnetic field strength along x, y, and z	2
	Piecewise correlation between magnetic field strength along xy, yz, and zx	3
	Zero Crossing Rate (ZCR) along x, y, and z	3
Frequency Domain	Mean and S.D. of magnetic field strength along x, y, and z	6
	Max. and Min. of magnetic field strength along x, y, and z	6
	Kurtosis and Skewness of magnetic field strength along x,y, and z	3

Notes: S.D.: Standard Deviation; Coef.: Coefficients altogether three windows form 96 elements feature vector for each character gesture.

C. SVM Classifier Modeling

The extracted feature vector is used as input to multi-class Support Vector Machines (SVM) to classify different character gestures. There are two approaches to build multi-class SVM : (a) one-versus-one and (b) one-versus-all. Here, one-versus-all approach is used [16]. The non-linear kernel function namely radial basis function (RBF) is applied in our experiment and it is defined as

$$RBF : K(x, x') = \exp^{(-\frac{\|x-x'\|_2^2}{2\sigma^2})} \quad (1)$$

Where x and x' represent feature vectors in input space. $\|x - x'\|_2^2$ denotes squared Euclidean distance between the two feature vectors. σ is a kernel parameter and set $\sigma = 10$.

III. Experiments

A. Apparatus

We used inbuilt magnetometer sensor of Samsung Galaxy S4 GT-I9500 android smartphone and placed N54 grade, disk-shaped (10mm $\times$ 3mm in diameter and height respectively) neodymium magnet on the finger using a small strip of Velcro. This sized magnet provides an effective range of about 10cm.

B. Data Collection

To collect training and testing dataset, we tried for four days in the lab environment. We invited 3 users (2 Male + 1 Female), aged between 22 and 26 years. All were regular smartphone users, right-handed, and spend on an average 4 hours per day. But, none had used Graffiti and EdgeWrite earlier. Each participants received a demonstration before experiments and practiced each alphabets at least 15 times. For quick learning, a printed copy of two unistroke character sets was visible during trials.

Fig. 4. Experimental setup of MagiText.

TABLE II
Character Recognition Performance of MagiText System

Characters	Recognition Rate(F-Score)	
	Graffiti	EdgeWrite
A	0.764	0.924
B	0.719	0.841
C	0.733	0.866
D	0.845	0.929
E	0.825	0.931
F	0.862	0.908
G	0.759	0.972
H	0.792	0.861
I	0.828	0.848
J	0.814	0.839
K	0.836	0.954
L	0.802	0.850
M	0.786	0.849
N	0.774	0.852
O	0.882	0.938
P	0.806	0.912
Q	0.844	0.938
R	0.783	0.880
S	0.852	0.929
T	0.863	0.914
U	0.794	0.908
V	0.786	0.889
W	0.828	0.861
X	0.830	0.916
Y	0.848	0.894
Z	0.834	0.851
Average	**0.812**	**0.894**

Each user holds the phone in one hand and move the magnet mounted finger of other hand in front of the device (Fig. 4) . They were asked to repeat each gesture 50 times. We developed an android application to record the magnetic signals at 100 Hz sampling frequency. In this way, we collected total 7800 samples (i.e. 50 times $\times$ 3 users $\times$ 26 alphabets $\times$ 2 types gesture set). Then, features are extracted from those signals as described earlier. The extracted features are used for classification using SVM classifier.

C. Classification Results

We perform two separate experiments - one for Graffiti and another for EdgeWrite. The Graffiti and EdgeWrite dataset contain 3900 samples each. We split each dataset into 10 subsets for carrying out a ten-fold cross validation test. The number of samples in training, validation, and test sets are in 6:2:2 ratio. To analyze the classifier performance, we used F_1 score [11]. Let, the amount of true positives is TP, the amount of false positives is FP and the amount of false negatives is FN. Then, $Precision$ is defined as $P = \frac{TP}{TP+FP}$. $Recall$ is defined as $R = \frac{TP}{TP+FN}$ and F_1 is $\frac{2PR}{P+R}$. The F_1 score ranges from 0 to 1. Table 2 represents the overall performance of MagiText system. From our result, it is observed that MagiText system based on EdgeWrite character set can achieve 89.4% accuracy, whereas Graffiti can distinguish characters with 81.2% recognition rate. This is because user can easily map the EdgeWrite character's corners with the phone's four corner-points at the time of drawing gestures. As a result, EdgeWrite gesture input is less ambiguous than Graffiti.

IV. Resource Profile : CPU, Memory & Power Consumption Benchmark

We measure the CPU, memory, and power consumption footprints of MagiText with the help of 'Power Tutor' and 'OS Monitor' apps, available at Google Play Store. The CPU usage is less than 4% during idle state and on average of 16% at the processing time. The memory consumption is $\sim$7.8MB during silence and reaches $\sim$20.54MB during running period. The average power consumption for sensor reading is less than 18.76mW.

V. MagiText Android App

We developed MagiText system as an Android application in Samsung Galaxy S4. The main components, as described in 'MagiText Framework' section, were fully implemented in Java SE7 and successfully running on Android smartphone. To build this prototype, we used libgdx library [8] to perform a fast Fourier transform (FFT) on incoming sensor signals in real time. We also used SMO WEKA [15] machine learning library for Support Vector Machines (i.e. SVM) classifier. Figure 5 shows the app's user interface (UI). By selecting 'Start' and 'Stop', user enters character gestures and collects magnetic sensor readings and 'Del' button is used to delete recorded magnetic signal. Finally, by pressing 'Recognize Character' button, it extracts features from signal and recognizes particular character from input gesture. On Samsung Galaxy S4, the duration of overall process is $\sim$35.23 seconds.

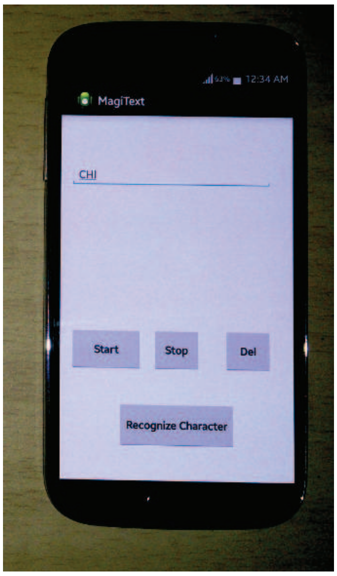


Fig. 5. GUI of MagiText.

VI. Conclusion & Future Work

We presented MagiText, based on drawing character gesture in the space around the device using a magnet. We also compared the character recognition accuracy on the basis of Graffiti and EdgeWrite gestures. This approach can be particularly suitable for very small mobile and tangible devices and we can use it as an efficient text entry mechanism, instead of touch screen keyboard, for taking short messages or notes quickly.

In future, we will focus on (1) studying sophisticated signal processing and advanced machine learning algorithms, so that MagiText should consume very less time to recognize a character; (2) user independent character recognition; (3) automatic identification of gesture starts and ends; (4) continuous character i.e. word recognition instead of single character input; (5) word prediction and completion; (6) a large scale user study to understand the acceptability of MagiText system.

References

[1] S. Agrawal, I. Constandache, S. Gaonkar, R. R. Choudhury, K. Caves, and F. DeRuyter, Using mobile phones to write in air. In MobiSys 2011, Pages 15-28.

[2] T. Deselaers, D. Keysers, J. Hosang, and H. A. Rowley, GyroPen: Gyroscopes for Pen-Input with Mobile Phones. In IEEE Transactions on Human-Machine Systems, 2014, Pages 1-9.

[3] D. Kim, J. Lee, H. Lim, J. Seo, and B. Kang, Efficient dynamic time warping for 3D handwriting recognition using gyroscope equipped smartphones. In Expert Systems with Applications 41 (2014), Pages 51805189.

[4] P. O. Kristensson, Five Challenges for Intelligent Text Entry Methods, AI Magazine, Vol 30, No 4, 2009, Pages 85-94.

[5] A. R. H. Fischer , K. J. Price, and A. Sears, Speech-Based Text Entry for Mobile Handheld Devices: An Analysis of Efficacy and Error Correction Techniques for ServerBased Solutions, International Journal of Human-Computer Interaction, 19:3, 2005, Pages 279-304.

[6] J. Song, G. Srs, F Pece, S. R. Fanello, S. Izadi, C. Keskin, and O. Hilliges, In-air Gestures Around Unmodified Mobile Devices, In UIST '14, Pages 319-329.

[7] S. Kratz and M. Rohs, HoverFlow: Expanding the Design Space of Around-Device Interaction. In MobileHCI'09.

[8] Libgdx libraries, http://www.java2s.com/Code/Jar/g/Downloadgdxaudiojar.htm

[9] M. L. Goc, S. Taylor, S. Izadi, and C. Keskin, A low-cost transparent electric field sensor for 3d interaction on mobile devices. In CHI'14, Pages 3167-3170.

[10] C. Harrison and S. E. Hudson, Abracadabra: Wireless, High-Precision, and Unpowered Finger Input for Very Small Mobile Devices. In UIST'09, Pages 121-124.

[11] Powers, D. M. W. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. Journal of Machine Learning Technologies 2, 1 (2011), Pages 3763.

[12] H. Ketabdar, M. Roshandel, K. A. Yksel, MagiWrite: Towards Touchless Digit Entry Using 3D Space Around Mobile Devices. In MobileHCI'10, Pages 443-446.

[13] J. Wobbrock and B. Myers, Trackball text entry for people with motor impairments. In CHI'06, Pages 479-488.

[14] http://en.wikipedia.org/wiki/Graffiti_(Palm_OS)

[15] http://www.cs.waikato.ac.nz/ml/weka/

[16] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, SVM and kernel methods matlab toolbox, 2005.