MicaPen: A Pen to Write in Air Using Mica Motes

Amitangshu Pal
Computer and Information Sciences, Temple University, Philadelphia, PA 19122
E-mail:{amitangshu.pal@temple.edu}

***Abstract*—This paper demonstrates a technique to write in air using Micaz motes. The application, named MicaPen is low-cost and can be used by the disabled who does not have fingers or limbs. Patients can wear the mote as a wrist watch or bracelet and move their hands based on what character they want to write. The mote detects the movement pattern using the accelerometer of MTS310 sensorboard and displays the characters on the screen. The application can also be extended in several applications including virtual touchscreen or remotely interact with different controlled devices.**

***Index Terms*—Write in air, accelerometer, hand-gestures, alphabets, human-sensor interaction.**

I. Introduction

A. Motivation

Many disabled patients often loose finger or limbs. These people cannot write or type on keyboards as a normal person. Many of them practice writing using their toes, which is hard for many people and take time to learn. But they are capable of showing some gestures using their hands, which are often known as sign-languages. Thus it will be useful to develop some hand-worn device that understand their hand-gestures and convert their gestures to a character and print of the computer screen.

The above example is only a representative application of a broad categories of human-sensor applications. For example, an user wearing such device can remotely control the television, game consoles, powerpoint presentations, flying drones, robots or can record some useful input texts to any connected devices remotely. While searching for a movie with a conventional remote control, entering the movie name becomes a monumental task, which results in poor user experiences. An hand-word character recognition remote control will certainly enrich the user experience and interaction. In surgical environments, hand gesture systems can help the doctors to manipulate the patients medical record, related digital images using certain hand gestures instead of touching the screens or keyboards [1].

B. Key contributions

This paper introduces MicaPen, a low-cost and east-to-learn device that will be useful to human-sensor hand gesture related applications. The solution uses the in-built accelerometer sensor of the MICAz sensor node [2], mounted on the user's hand to detect the hand gestures, and then interpret this for English character and number recognition (although the application is generic for identifying general gesture recognition). The key idea is to interpret the accelerometer readings to recognize different combination of hand *strokes* and then match these sequences with a predefined set of stroke sequence corresponding to different characters. Although the idea is tested on MICAz sensor nodes, the concept is applicable to any in-built accelerometer equipped, hand-worn (or finger-worn) devices such as smartwatches, wristbands, rings etc.

Although sounds simple, such solutions suffers different real-world challenges, including the inherent noise of the accelerometers, involuntary hand movements etc. The current work is an early step towards solving this over difficult problem. The key contribution is to explore the viability of using cheap accelerometer sensors that are ubiquitously available in different hand-worn devices to recognize different hand strokes and then represent them in distinct human-readable characters. The solution and stroke recognition technique proposed is made to be deliberately simple and lightweight to be implemented in cheap sensing devices. The feasibility of the proposed technique is thoroughly tested with real sensing devices.

C. Paper organization

The paper is organized as follows. Detailed related works for hand gesture detection are discussed in section II. Section III describes overall system design and implementation of MicaPen. This section also shows the principle behind identifying different hand strokes, and map them into different readable characters, along with some experimental results of MicaPen. Finally, the paper is conclusion in section IV along with the relevant discussions and future works.

II. Related Work

Hand movement detection using different sensors is well researched, which can be categorized into vision based, RF-based, and motion sensor based approaches. Below I discuss the existing works for each of these categories.

A. Vision based gesture recognition

Vision based gesture recognition has also been studied in the literature. TinyMotion [3] captures images from the camera and uses the image processing techniques for detecting certain movements, like horizontal, vertical, rotation etc. "Write in air" [4] is another similar project initiated by Microsoft that also used computer vision to recognize air written alphabets. In [5], [6] the authors have used color gloves or hand markers to measure the position, angles and hand orientations to recognize the hand gestures. A review of visual interpretation of hand-gestures are provided in [7].

B. RF based gesture recognition

RF based gesture recognition is presented in [8] where the authors have used the idea of Doppler shift to detect user's hand gestures. AllSee [9] and WiGest [10] perform WiFi-based gesture recognition using RSSI and CSI values from the off-the-shelf devices. RF-IDraw [11] uses commercial off-the-shelf RFID readers and tags to accurately track the trajectory of the hand movements based on RF-based positioning. WiDraw [12] uses the angle-of-arrival based hand motion tracking system without any wearable, and still achieves accurate trajectory.

C. Motion sensor based gesture recognition

Several approaches have been presented in the context of accelerometer/gyroscope based gesture detection applications. Wiimote [13] is a hand movement tracking remote that uses 3-axis accelerometer, gyroscope, and optical sensors to recognize different arm motions. Logitech Air Mouse [14] provides mouse like functionalities on air, by using sensor like accelerometers and gyroscopes. PyAcceleREMOTER [15] is also an accelerometer based remote controller MPlayer for GHU/Linux, where the tilting of a N95 phone is detected to implement different functions of a player such as play, stop, fast-forward, rewind etc. In [16] the authors have proposed a wearable energy-harvesting ring that can track the user's finger movements and gestures. The ring uses a finger-tendon pressure-based solution to detect touch, and an audio based solution for detecting finger motion. In [17] the authors have introduced an arm-band that can recognize different arm and hand gestures. The band mainly detects the electrical activity in the muscles for recognizing such gestures. Authors in [18] have presented the motion energy measurements of from a smartwatch and exploit this for finger writing. PhonePoint Pen [19] uses the accelerometer of the mobile phones to detect English characters using geometric strokes.

MicaPen falls under the category of motion sensor based hand recognition approaches. Although such approaches have been well-studied in the past, this is the first work that test the feasibility of hand gesture detection using MICAz sensors, with simple straight line based hand strokes.

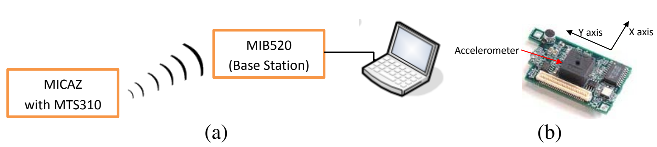

Fig. 1. (a) The system model of MicaPen. (b) MTS310 sensorboard.

III. System Design and Implementation

A. System architecture

Our system model is shown in Fig. 1(a) that includes MICAz motes and one MTS310 sensorboard [20]. Our main objective is to use the two-axis accelerometer ADXL202JE (shown in Fig. 1(b)) of the MTS310 to recognize different hand gestures. As shown in Fig. 1(a), a MICAz mote attached with MTS310 sensorboard is fastened on someone's hand with a wrist watch. The person moves his hand based on what character he wants to type. The accelerometer stores the accelerator value and sends this to the basestation attached to a computer. Then the basestation recognizes the character that the person wants to write based on his hand movements and displays that on the screen.

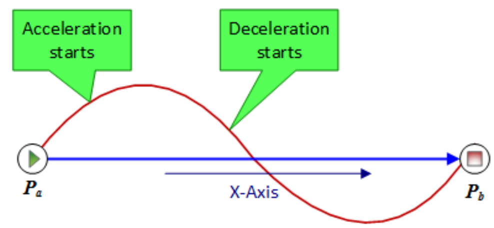

Fig. 2. The variation of acceleration while the sensor node is moved P_a to P_b in the +x-axis, which represents the stroke $\rightarrow$.

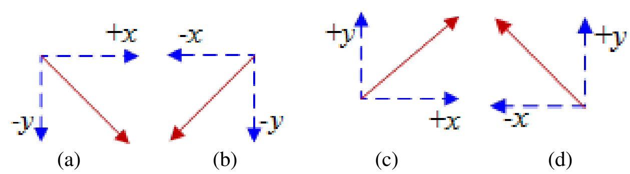

Fig. 3. The decomposition of movement patterns $\searrow$, $\swarrow$, $\nearrow$ and $\nwarrow$ in horizontal and vertical axis respectively.

B. Key concepts used in identifying strokes

MicaPen mainly detects eight sets of hand movement, denoted as $\rightarrow$, $\leftarrow$, $\uparrow$, $\downarrow$, $\searrow$, $\swarrow$, $\nearrow$ and $\nwarrow$. Let us call these patterns as our set of *strokes*. These eight strokes can be categorized into two categories: (a) vertical/horizontal strokes (i.e. $\rightarrow$, $\leftarrow$, $\uparrow$), and (b) non-vertical/horizontal or diagonal strokes (i.e. $\searrow$, $\swarrow$, $\nearrow$, $\nwarrow$). The key challenge then is to identity the strokes using the two-axis accelerometer of MTA310. Let's see the physics behind recognizing these two types of strokes using an accelerometer readings.

Vertical/horizontal stroke detection: The stroke $\rightarrow$ represents a hand movement from left to right, i.e. in the positive x-axis. Thus when the hand starts moving the x-axis acceleration starts increasing and when the hand stops, the acceleration decreases and comes back to zero. Fig. 2 shows this phenomenon while the sensor node is moved from P_a to P_b. The y-axis acceleration remains same as there is no movement in the y-axis.

This phenomenon is tested with the actual accelerometer sensor minted on MTS310 sensorboard. The results is shown in Fig. 4(a), which shows the similar behavior of x-axis acceleration variation, whereas y-axis ADC readings remains almost constant. Fig. 4(b) shows the movement in –x-axis (i.e. $\leftarrow$) which shows the opposite phenomenon. Fig. 4(c)-(d) show the movement in +y-axis and –y-axis which shows that the similar phenomenon applies for the y-axis ADC values as well.

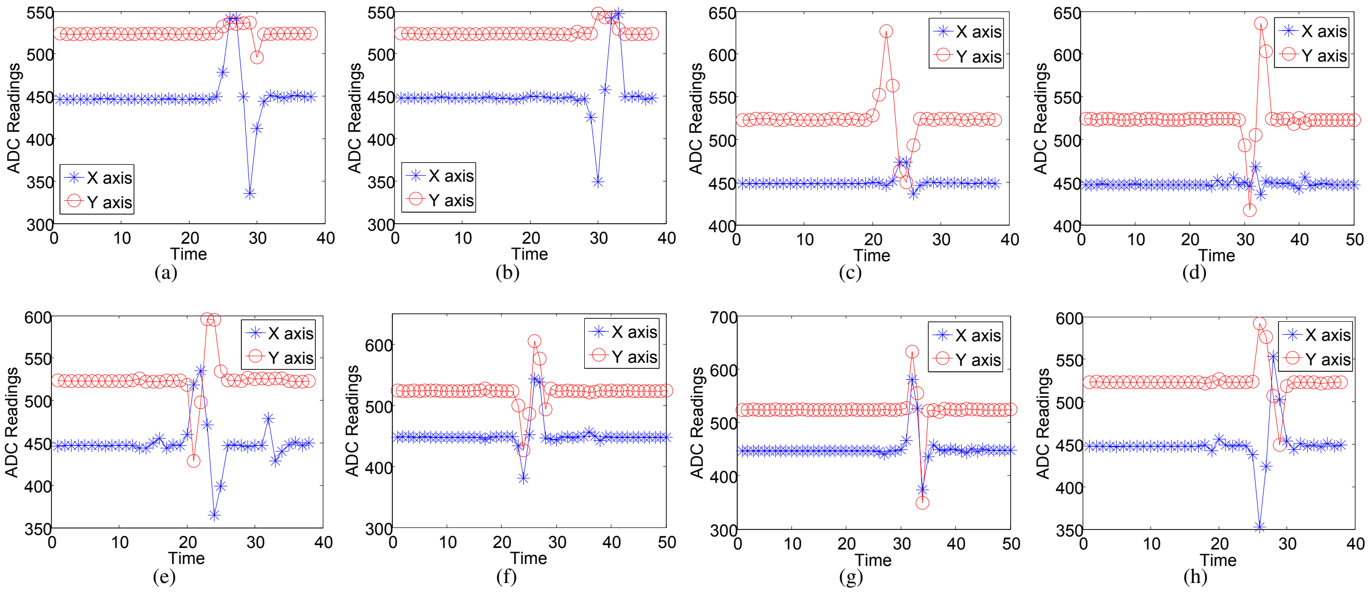

Fig. 4. ADC readings for the movement patterns →, ←, ↑, ↓, ↘, ↙, ↗ and ↖ respectively.

Diagonal stroke detection: In case of diagonal movements like ↘, can be decomposed into ↓ and → in x-axis and y-axis respectively as shown in Fig. 3(a). This is equivalent to moving the sensor node in $+x$-axis and $-y$-axis, which is reflected in the ADC readings of Fig. 4(e). Thus the ↘ stroke in Fig. 4(e) represents the superposition of Fig. 4(a) and Fig. 4(d). Similarly the stroke ↙ can be decomposed into ← and ↓ respectively as shown in Fig. 3(b). This is also confirmed from the corresponding ADC readings of Fig. 4(f). The decomposition of strokes ↗ and ↖ is shown in Fig. 3(c)-(d), along with their ADC readings in Fig. 4(g)-(h) respectively.

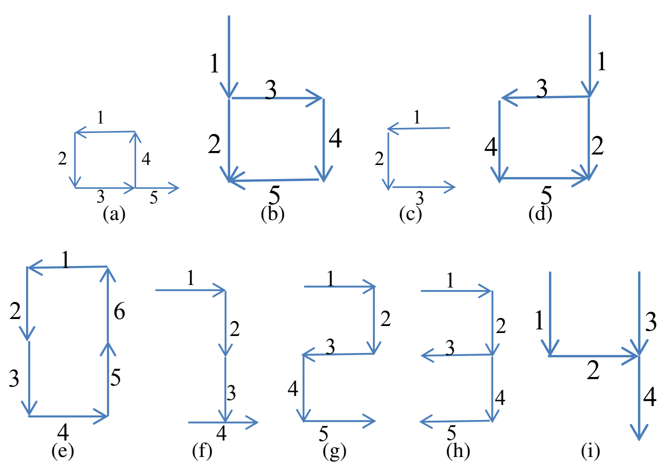

Fig. 5. The writing patterns of alphabets and numbers in MicaPen.

C. Mapping the strokes into representative characters

In MicaPen the alphabets (a-z) as well as all numbers (0-9) have to be written by following a set of hand strokes as shown in Fig. 5. For examples the letter 'c' can be represented by using three strokes ←, ↓, and → respectively. Similarly the number '2' can be represented using →, ↓, ←, ↓ and → respectively. Table I summarizes the set of strokes used in MicaPen to detect different characters ('a'–'z') and numbers ('0'–'9'). These set of strokes are unique for individual characters, and thus there is no ambiguity.

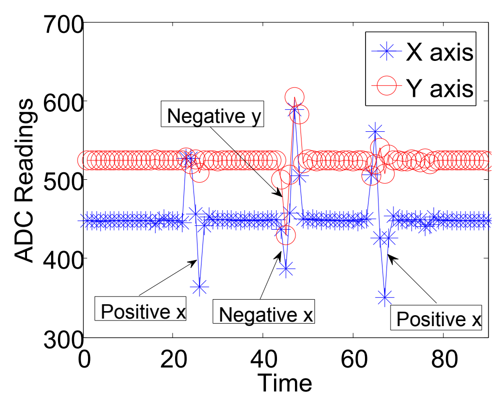

Fig. 6. ADC readings while writing 'z', which consists of the set of strokes are →, ↙ and → respectively.

Now let us see one example of how these strokes can be identified for the alphabets and numbers recognition in MicaPen. Suppose we want to write 'z'; first we need to move our hands in the positive x direction (→), then pause a little bit, then move like ↙, pause a bit and move again in positive x direction. These pauses are necessary so that the system can detect the end of one stroke and the start of the next one. The ADC readings corresponding to writing 'z' is shown in Fig. 6. Similarly all the characters can be recognized by detecting these eight set of strokes.

TABLE I
DIFFERENT CHARACTERS AND THEIR CORRESPONDING STROKES FOR IDENTIFYING THEM

Character	Strokes	Character	Strokes	Character	Strokes	Character	Strokes
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**	(→, ↙, →)	**0**	(←,↓,↓,→,↑,↑)	**1**	(→, ↓, ↓, →)
2	(→,↓,←,↓,→)	**3**	(→, ↓, ←, ↓, ←)	**4**	(↓,→,↓,↓)	**5**	(←, ↓, →, ↓, ←)
6	(←,↓,↓,→,↑,←)	**7**	(→, ↓, ↓)	**8**	(←,↓,→,↓,↓,←,↑)	**9**	(←, ↓, →, ↓, ↓)

D. Other details

When an user finishes writing a character, he needs to give some indications so that the basestation can understand the end of one character. While writing on a paper, we use a gap between two characters. In this paper the photo sensor (the photo sensor is part of the MTS310 sensorboard) is used to take that indication from the user. When an user finishes one character, he puts his hand on the photo sensor. It blocks the light on the photo sensor, thus its value goes down. The basestation detects this and understands the end of one character and prints it on the screen. Also if there is some typing errors occur in this process, the person can delete the previously written character by using the strokes ← and then insert the right character. The implementation gives accurate character/number recognition as far as the individual strokes are detected correctly.

IV. DISCUSSIONS, CONCLUSION AND FUTURE WORKS

A. Discussions

MicaPen has the potential to enable fine grained gesture wrist and finger recognition, which opens many new avenues for human-sensor interactions with smart environments. This paper explore MicaPen only for character recognition, however, the potential of MicaPen can be extended for any hand gesture recognition based applications. There are several directions on which the current application can be improved. For example, in this paper each English letter or number is represented by a set of unique strokes, however, there may be multiple ways a character can be represented. This feature can be easily included into the current application, by incorporating multiple patterns corresponding to any character. Another point to note is that in this paper, the characters are represented using eight different strokes for simplicity. In general the characters can be written in cursive manner, which will be different from recognizing strokes. Such increments can also be incorporated into the MicaPen application by pre-training a machine learning model or dynamic time warping based pattern matching model for character recognition techniques.

B. Conclusion and future works

This paper demonstrates a hand-gesture recognition application using MICAz mote and mts310 accelerometer for writing anything on air. The application is easy to learn and is low-cost that makes is suitable for real application for disabled. Although in this work, the idea of MicaPen is demonstrated for character and number recognition, the concept can be also used for various hand gestures and sign-language recognition for real world applications.

REFERENCES

[1] B. D. Itkowitz, S. Dimaio, and T. Zhao, "Method and apparatus for hand gesture control in a minimally invasive surgical system," 2010.

[2] "Micaz Datasheet," http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf.

[3] J. Wang and J. F. Canny, "Tinymotion: camera phone based interaction methods," in *CHI*, 2006, pp. 339–344.

[4] "Techfest: Write in the air," http://www.youtube.com/watch?v=WmiGtt0v9CE.

[5] R. Cipolla, Y. Okamoto, and Y. Kuno, "Robust structure from motion using motion parallax," in *International Conference on Computer Vision*, 1993, pp. 374–382.

[6] J. Davis and M. Shah, "Recognizing hand gestures," in *European Conference on Computer Vision*, J.-O. Eklundh, Ed., 1994, pp. 331–340.

[7] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, 1997.

[8] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *ACM MobiCom*, 2013, pp. 27–38.

[9] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *USENIX NSDI*, 2014, pp. 303–316.

[10] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *IEEE INFOCOM*, 2015, pp. 1472–1480.

[11] J. Wang, D. Vasisht, and D. Katabi, "Rf-idraw: virtual touch screen in the air using RF signals," in *ACM SIGCOMM*, 2014, pp. 235–246.

[12] L. Sun, S. Sen, D. Koutsonikolas, and K. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *ACM MobiCom*, 2015, pp. 77–89.

[13] "Nintendo," https://www.nintendo.com/switch/.

[14] "Logitech mx air rechargable cordless air mouse," https://www.youtube.com/watch?v=rlurAy_jnP8.

[15] "Pyacceleremoter: N95 remote control for mplayer," https://www.youtube.com/watch?v=fz8OKG4FI2Q.

[16] J. Gummeson, B. Priyantha, and J. Liu, "An energy harvesting wearable ring platform for gestureinput on surfaces," in *ACM MobiSys*, 2014, pp. 162–175.

[17] "Thalmic labs myo gesture control armband for presentations," http://www.obitko.com/tutorials/genetic-algorithms/.

[18] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *HotMobile*, 2015, pp. 9–14.

[19] S. Agrawal, I. Constandache, S. Gaonkar, R. R. Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *ACM MobiSys 2011*, 2011, pp. 15–28.

[20] "MTS/MDA sensor board users manual," http://www.xbow.com.