HandMagic: Towards User Interaction with Inertial Measuring Units

Jules C. Calella, Francisco R. Ortega, Naphtali Rishe, Jonathan F. Bernal, Armando Barreto
Florida International University
Miami, FL
jcale004@fiu.edu, fortega@fiu.edu, ndr@acm.org, jbern102@fiu.edu, barretoa@fiu.edu

***Abstract*—We present a method of gesture recognition designed to be used with inertial measurement units (IMUs) for 3D user interaction. We provide a drone simulator environment to validate our work. It is important to use motion sensors for user interaction as other devices provide challenges not present in motion-sensors. It is also important since the current field of user interaction is moving towards a multi-modal paradigm and IMUs have been under studied for this field. This work presents the functionality of IMUs with a 6-axis model. We discussed our pilot study and its results, which were satisfactory.**

Keywords*—*IMU; Motion; Sensors; User Interaction

I. Introduction

Today we are faced with the emergence of many types of input devices that can improve user interaction, including multi-touch, vision-based devices (e.g., Kinect), and motion sensors (e.g., gyroscopes). However, motion sensors (e.g., inertial measurement units (IMU)) have been underutilized for user interaction. This is probably due to the inaccuracies and high price point of past IMUs [1], but technology has kept improving, bringing about better noise reduction filters and smaller packages. Other devices have received more research. For example, the use of vision-based recognition is becoming as pervasive as multi-touch interaction; however, it is not always the best choice (e.g., infrared does not work well outdoors). When it comes to devices used with the hand, flex sensors [2] have been popular but they are bulky and uncomfortable for long-term use. Our motivation is to move the state of the art for motion sensors used in 3D user interaction. We proposed a small IMU system, which we called HandMagic, that doesn't require additional input for gesture recognition and can be used with 3D real-time virtual environments (e.g., drone simulator). We believe that our pilot study results provide evidence as to why motion sensors should be an integral part of user interaction.

II. Background

Various advances in the area of hand gesture recognition have been made over the last two decades and more accurate sensors for their classification and recognition have been devised [3]. Some of these advancements have been propelled towards wearable-based technologies capable of interacting with virtual and augmented environments [4]. Hand-worn technologies have been built based upon three main approaches: (i) optical sensing, (ii) inertial sensing, and (iii) acoustic sensing. None of these techniques has become an all-in-one solution for all types of hand recognition applications [5]. More novel approaches exist too, such as the method used in [6] applying electromyography to detect muscle movement.

Our work is focused on the inertial sensing approach. Inertial sensing implementations have been limited to the exclusive use of gyroscopes and accelerometers, such as the work in [7]. Additionally, recent contributions, such as those in [8] have incorporated wireless communication capabilities. Other works have incorporated optical recognition techniques; however, these implementations have tried to move away from a glove-based approach to a hand-based approach [9].

Other recent works, although not built onto gloves, are being worn on extremities close to the hands, such as wrists and fingers as in [10]. These devices, although less invasive, restrict the set of gestures that could otherwise be recognized.

III. Sensor Hardware and configuration

The hardware for this approach was selected to be fast, low power, and be user-calibration-free. Calibration can be considered a difficult process for users [11] and should be avoided to make the experience more user friendly. Calibration can be avoided by not using devices whose measurands differ based on location or other factors. Calibration should not be confused with training as a device can benefit from learning how a user's gesture differs from someone else with different physical attributes.

A. Microcontroller

The PIC24FJ128GB202 microcontroller was selected for its high-speed, low power, and USB compatibility. The PIC24F family of microcontrollers are the 16-bit models that come with an array of useful hardware features, such as built-in oscillators, digital communication, among others. The internal oscillator is accurate enough for USB communication and does not use additional space. The internal clock was run at 32 MHz.

B. MEMS Inertial Measurement Unit

The LSM9DS1 is a 9-axis MEMS device by ST Microelectronics. It is comprised of an accelerometer that measures in 'g's', a gyroscope that measures in 'degrees per second', and a magnetometer that measures in 'gauss'. Each device provides a configurable measurand range. In this work, we selected a range of ±4g for the accelerometer and ±500 dps for the gyroscope; the magnetometer was switched off as it

978-1-4799-8287-5/16/$31.00 ©2016 IEEE

required calibration by the user. It can be interfaced with another device using either SPI or I2C. In this work, SPI communication was used for its high-speed data transmission; the data received is formatted to 16-bits per axis and represented in 2's complement.

C. Additional Components

Additional components are a voltage regulator to provide the 3.3V to both devices, resistors and capacitors, a USB mini-B socket, and an LED for debugging. All components were soldered onto a 2 in. by 2 in. perfboard. The entire board was secured inside a container; the case was attached to a glove. The final product is shown in Figure 1.

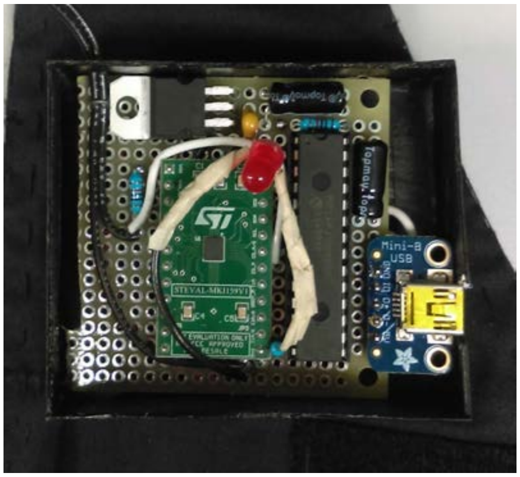

Fig. 1. Assembled hardware. *(Image shows top-side view of assembled hardware with container and glove.)*

IV. Methods: Gesture recognition

Several improvements and methods that allow more meaningful gestural tracking have been developed. Some of these methods involve determining the distance traveled by double integrating the acceleration values used as done in [12].

Our gesture recognition method uses raw data from the sensors and does not attempt to convert from acceleration to distance or degrees per second to an angle relative to an initial frame of reference. Instead, it uses a configurable number of, and values for, thresholds that determines when an action has been completed.

A. Operation

Initially, the data retrieved from the sensor is normalized to work with values between -1 and 1. This approach allows the values to be calculated with the same thresholds if desired, regardless of the sensor used. The premise of this method is to apply an N-based number system for different levels of sensitivity. If using a method with 'motionless', 'slow left', 'slow right', 'fast left', and 'fast right' as the five states, then a base-5 number system would be used. The states can all be assigned a value that explicitly represents only a single state, which can all be added together to give a number that represents a unique motion, which will be referred to as the gesture state.

For example, to label the states across the x-axis for the accelerometer, they could be given the numbers as follows: fast left (0), slow left (1), motionless (2), slow right (3), fast right (4). Since each number can only be used once (except for zero), reusing the numbers one to four for another axis or sensor would not be valuable for distinguishing the different movements. Instead, continuing with a sequence of new numbers is needed. Defining the values for the y-axis states retrieved from the accelerometer, the labels given will be: fast down (0), slow down (5), motionless (10), slow up (15), and fast up (20). When added together, the values for each motion will always be unique. A sum of 12 will always mean the sensor is motionless (2+10), whereas a sum of 9 will always represent moving down slowly while also moving to the right quick (4+5).

The pattern used for this example, with a total of five states, is different powers of five multiplied with a number from zero to four and summed together. This equation can be written in summation notation and be more generalized for a different number of states, as shown in Formula 1, where G_s is the result of the gesture state, L_n is the label value representing the state for that axis, B is the total number of states, and N is the number of axes to be measured.

$$G_s = \sum_{n=0}^{N-1} L_n \cdot B^n \tag{1}$$

Note that after determining the gesture state, the value is compared to the previous gesture state performed and ignored if they are the same. This prevents errors from different machines recording values at different rates. This correction also prevents gestures from being determined based on how long a single motion is being performed but instead whether or not it is performed at all.

After each gesture state is recorded, all previous gesture states are compared to a gesture recognition database to find a matching pattern. The comparison is started by searching for the most recent gesture state performed in the database. Upon finding a match, the next several gesture states recorded are compared to the next state in the database. The reason for using several samples is to account for non-ideal motions being performed. If a second match is found the same process is repeated, starting from the last matching gesture state and comparing several more gesture states until a match is found. This process continues until a full gesture is recognized or it is determined the gesture is not categorized. After a gesture is recognized, all previous recorded gestures states are erased. This process is continuous because it checks after each new gesture state is recorded. It differs from other algorithms that depend on a manual input of when the gestures begin and end, such as in [1] and [13]. The process is also configurable because the user may choose the thresholds for each state as well as how many samples are to be compared before determining the gesture is not categorized

V. User interaction and results

A six-degree-of-freedom pilot study was conducted by operating a drone in a custom simulation, which included three rotations (pitch, yaw, and roll) and three translations (up/down, left/right, back/forward). In addition, lift off was controlled with an additional button on the sensor. This pilot study was used to determine the feasibility of our sensor using the 3-axes from the accelerometer and the 3-axes from the gyroscope (in addition to the button). The five states mentioned were assigned to each of the 6 axes. Each gesture was programmed to compare up to 20 values to determine if a gesture exists.

The first gesture involved moving the hand from the palm facing the ground to the position where the palm faces the screen. The gesture was used in two separate programs. One of the programs was used for opening a web browser. The second was used in the 3D virtual environment and would pause and resume the game. These two tests helped determine that the gesture can be recognized when it is the only focus in a program as well as when used in a larger program with multiple tasks. The next two gestures began with holding the hand in the air with the pinky facing down and the thumb facing up. The hand could move in the direction the palm was facing for one gesture, or the opposite direction for the other. These two were implemented in a program that simulated pressing either the left arrow or the right arrow. When using a PowerPoint presentation, the user was able to change slides by simply moving their hand either left or right. While our initial pilot study was used to determine the feasibility of our sensor, most users were able to go collect the boxes (as shown in Figures 2 and 3) with a few of the subjects crashing. Overall, users stated their satisfaction with regards to their experience.

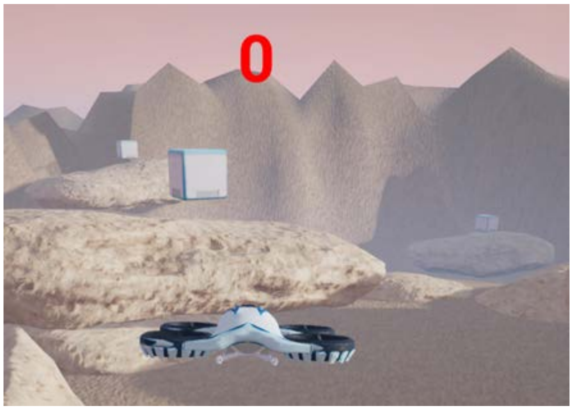

Fig. 2. In game footage. (*shown: the drone, collectable boxes, and score.)*

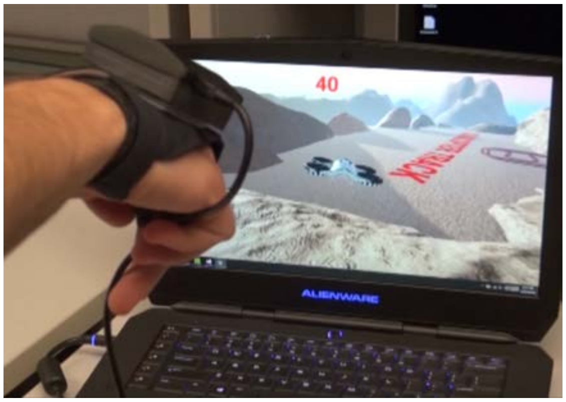

Fig. 3. User interacting with 3D virtual environment. *(User acquired four collectable boxes valued at 10 points each.)*

VI. Conclusion

We presented our sensor called HandMagic using a gesture recognition with raw data (no Kalman Filters) to demonstrate its use with a drone simulator. Our results are promising, showing that a real-time interactive 3D environment can be used with motion sensors. In future works, we expect to add additional sensors, such as Galvanic Skin Response sensor. A more important question is how to combine multiple devices, such as IMU, vision-based, speech, and multi-touch (and others) to provide a true multi-modal experience.

Acknowledgment

We acknowledge the support by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-1532061, OISE 1541472, MRI CNS-1532061, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, RAPID CNS- 1507611.

References

[1] Jing Yang et al., "A novel hand gesture input device based on inertial sensing technique," Industrial Electronics Society, 2004. IECON 2004. 30th Annual Conference of IEEE, 2004, pp. 2786-2791 Vol. 3.

[2] E. Jeong, J. Lee and D. Kim, "Finger-gesture Recognition Glove using Velostat (ICCAS 2011)," Control, Automation and Systems (ICCAS), 2011 11th International Conference on, Gyeonggi-do, 2011, pp. 206-210.

[3] H. Cheng; L. Yang; Z. Liu, "A Survey on 3D Hand Gesture Recognition," in IEEE Transactions on Circuits and Systems for Video Technology , vol.PP, no.99, pp.1-1

[4] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," in IEEE Computer Graphics and Applications, vol. 14, no. 1, pp. 30-39, Jan. 1994.

[5] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal", IEEE Comput. Grap. Appl., vol. 22, no. 6, pp. 24-38, 2002.

[6] Georgi M, Amma C, Schultz T, “Recognizing hand and finger gestures with IMU based motion and EMG based muscle activity sensing”, InProceedings of the International Conference on Bio-inspired Systems and Signal Processing 2015, pp. 99-108.

[7] Y. Kim, B. Soh and S. Lee, "A New Wearable Input Device: SCURRY", IEEE Trans. Ind. Electron., vol. 52, no. 6, pp. 1490-1499, 2005.

[8] M. Sama, V. Pacella, E. Farella, L. Benini and B. Ricco, "3dID: a low-power, low-cost hand motion capture device,"Proceedings of the Design Automation & Test in Europe Conference, Munich, 2006, pp. 6 pp.-.

[9] D. Kim et al., “Digits: Freehand 3D Interactions Anywhere Using a Wrist-Worn Gloveless Sensor,” User Interface Software and Technology Symposium, pp. 167-176, Oct. 2012.

[10] D. Ashbrook, P. Baudisch, S. White, “Nenya: Subtle and Eyes-Free Mobile Input with a Magnetically-Tracked Finger Ring,” Human Factors in Computing Systems. SIGCHI, 2011, pp. 2043-2046.

[11] P. Keir, J. Payne, J. Elgoyhen, M. Horner, M. Naef and P. Anderson, "Gesture-recognition with Non-referenced Tracking," 3D User Interfaces (3DUI'06), 2006, pp. 151-158.

[12] Shengli Zhou, Zhuxin Dong, W. J. Li and Chung Ping Kwong, "Hand-written character recognition using MEMS motion sensing technology,"2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Xian, 2008, pp. 1418-1423.

[13] Sung-Jung Cho et al., "Magic wand: a hand-drawn gesture input device in 3-D space with inertial sensors," Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on, 2004, pp. 106-111.doi: 10.1109/IWFHR.20