Real-Time Fingertip Gesture Recognition for Embedded Systems

Ji Kwang Kim, Jung Hwan Oh, Jun Hyeok Yang and Seung Eun Lee

Department of Electronic Engineering, Seoul National University of Science and Technology, Republic of Korea

Abstract

Augmented and virtual reality systems depend on accurate, real-time hand and fingertip tracking for seamless integration between real objects and associated digital information. In this paper, we propose a real-time fingertip gesture recognition system. We adopt microelectromechanical system (MEMS) Gyroscope device for tracking the movement of fingertip. The fingertip gesture is spotted and recognized based on cross-correlation and mean feature. Our system is implemented on Field-Programmable Gate Array (FPGA) and the experimental results shows the feasibility of our system with the average recognition rate of 86%.

Keywords: *Human-machine interface, Gesture recognition, Embedded system, Augmented Reality, Virtua Reality.*

I. INTRODUCTION

For a long time, humans have been interacting with computer by using keyboard and mouse. Nevertheless, people are still desiring more comfortable and efficient interactive device. The Human-Machine Interface (HMI) details how humans get interaction with machines [1]. In other words, the HMI means software or hardware that presents information to a client and delivers commands or control directions from the client [2]. Thanks to the advances in pattern recognition technology, new input methods using pattern recognition have been introduced [3-4].

HMI devices have been introduced in order to make it easier for the disabled people to interact with devices in daily life. T. Chakraborty *et al.* designed finger gesture recognition system for elderly people, using optical sensor array to detect combination of finger position [5]. P. Salunkhe *et al.* proposed a device which controls computer mouse cursor, exploiting human eyes tracking [6]. On the other hand, the HMI system has been proposed to bring out the potential of the various applications by improving the accessibility and usability. L. Long *et al.* proposed 3-dimensional computer-aided design (3DCAD) system using finger gesture recognition to assist 3D object manipulation and alignment [7]. Z. Yan *et al.* presented multi-touch gesture recognition system for virtual reality (VR) travel framework, mapping the finger gesture to the control of travel metaphors such as walking, Segway and surfboard [8]. S. Khare proposed security system using hand draw pattern [9]. Ultimately, The HMI is studied for the purpose of user friendly and intuitive communication with the machine [9-10].

The gesture recognition techniques are classified into two categories: Vision-based Gestures Recognition (VGR) and Sensor-based Gestures Recognition (SGR) [1]. Since the VGR methods recognize the user's gesture using an optical sensor without any attachment to the body, it is not under behavioral restriction. However, the VGR methods require various techniques such as the detection of the Region of Interest (RoI), edge detection, and feature extraction [11-12]. These techniques have heavy arithmetic operations and require high-performance resource for real-time operation. Also, the vision-based approach is sensitive to lighting condition and camera facing angles [13]. The SGR methods make use of sensors to acquire gesture information. For example, the electromyography (EMG) or microelectromechanical system (MEMS) sensor is attached to hands or arms in order to measure the muscle activation or physical movement accordingly [14-17]. Especially, the MEMS sensors are pervasive in consumer devices such as smartphone, wearable device and game controller [13]. It is useful for cost-sensitive embedded devices because of its affordable cost.

The gesture recognition technique that automatically detects the start and end points of gesture is called a continuous gesture recognition (CGR) technique. In order to realize the CGR, the sensor data successively coming from sensor device should be processed on-the-fly. In the embedded devices having resource constraints, an excessive algorithm would be a critical task that causes timing delays. Therefore, it is important to use appropriate algorithm with reasonable accuracy. In this paper, we propose real-time fingertip gesture recognition system, focusing on the extremely reduced complexity. Our system continuously detects easy-to-act fingertip gesture without any additional action for gesture spotting. We adopt only 3-axis gyroscope sensor, which is attached to fingertip. In order to detect fingertip gestures, we use the cross-correlation between sensor data and coefficients for each fingertip gesture. The proposed recognition system was realized on Field-Programmable Gate Array (FPGA) and the functionality was validated through the experiment. We expect that our system can be efficiently used to various electronic embedded devices having low-cost and tiny-hardware resource interacting with augmented and virtual reality systems.

The rest of this paper is organized as follows. In Section II, we present correlation based fingertip gesture recognition technique including the definition of gestures. The architecture of our system is described in Section III. Section IV presents the FPGA implementation details, demonstrating the feasibility of our fingertip gesture recognition system. Finally, we conclude in section V with summarizing our paper.

II. Correlation Based Gesture Reconition

In order to acquire the information of the fingertip movement, we use 3-axis gyroscope measuring angular motion. Figure 1 shows the directions of the rotational axis of gyroscope attached on an index finger. We extract the variation of rotation moment in terms of upward, downward, leftward, and rightward by using the gyro data on the *x* and *y* axis. The sensor device is programmed and controlled by accessing the internal register of the sensor device through Inter-Integrated Circuit (I2C). We can configure the sensor device by writing data in the register. Also, we obtain the gyro data by reading the register. We receive the sampled data with the 32*Hz* sampling rate, 500°/*s* full scale range and 16bit data width.

We have defined the four classes of gesture which indicate up, down, left, and right as shown in Figure 2. The gestures are defined as cyclical pattern, in which hand moves from a rest position and returns to its rest position. Exploiting this closed-form, the cyclical gesture can be recognized more correctly than typical gesture because of the periodicity and the phase of gesture [16]. Figure 3 shows the variation of angular velocity obtained by performing 50 iterations for each gesture. In the actual measurement, we found that the angular velocity on the *y*-axis moves most sensitively on the up and down movement, and the angular velocity on the *x*-axis moves most sensitively on the left and right movement. Also, because of the characteristic of the cyclical pattern, the variation of each angular velocity always has both start direction and reverse direction, which is opposite to the start direction. Therefore, this characteristic of the closed-form can be applied with the cross-correlation.

$$C_{ges}(n) = \int_0^W S_{dir}(n-\tau)R_{ges}(\tau)\, d\tau \qquad (1)$$

In our gesture recognition, the cross-correlation is defined as (1). When *n* denotes sample meaning discrete time, *C* represents correlation value for gesture, which is previously defined as *ges-up*, *ges-down*, *ges-right* and *ges-left*. *S* indicates sensor data on the direction that will be *x* or *y* axis. Each data set of angular velocity, previously obtained as shown in Figure 3, is used as correlation coefficient *R*. The cross-correlation *C* is used for referring to the correlation of sensor signal with gesture coefficients. When a correlation value has high positive value, it is considered that the sensor signal is correspond with the relevant gesture.

In our system, the correlation values for the four gestures are calculated for each sampling time. The corresponding gesture is activated when the correlation value for the gesture exceeds a given threshold. However, the correlation value would unintentionally exceed the threshold value when the sensor data becomes excessive for non-cyclical operation. To prevent this incorrect detection, we applied the mean feature. As mentioned above, the cyclical pattern has bi-directional movement. In this pattern, the mean of the signal energy for entire pattern period is approximated to zero. Therefore, when the mean value becomes larger than a given threshold, it is not considered to be a cyclical pattern, so the correlation value exceeding the threshold is ignored. In this way, it is possible to detect the correlation value only for the cyclical pattern.

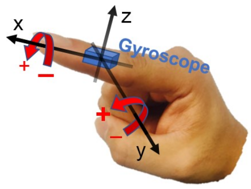

Figure 1. The directions of the rotational axis of gyroscope

Figure 2. The behavior of defined gestures

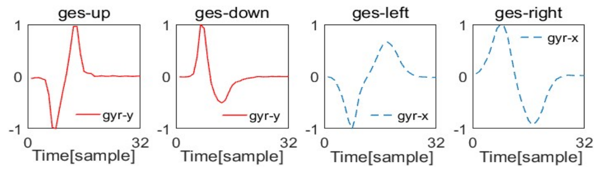

Figure 3. The variation of the angular velocity

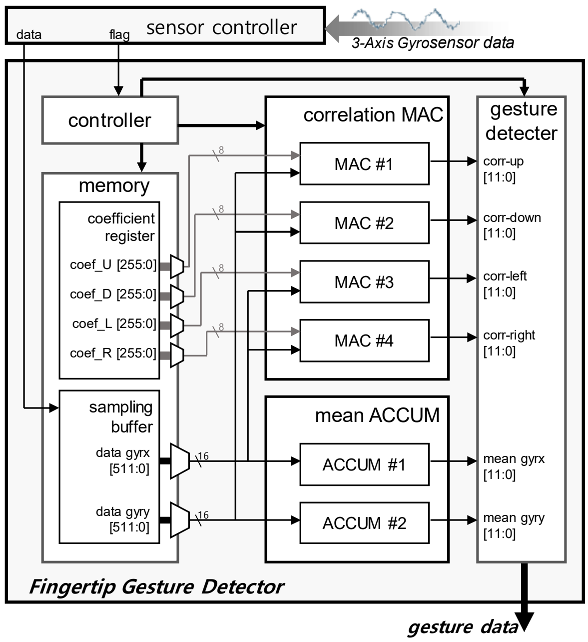

Figure 4. The architecture of the finger gesture recognition system

III. System Archtecture

Figure 4 shows the architecture of our fingertip gesture recognition system, which is connected to the *sensor controller* to communicate with the gyro sensor. The *sensor controller* reads the value from the sensor every sampling period and transmits the data to the *finger gesture* *detector* with the *flag* signal indicating valid data. The *fingertip gesture detector* consists of *controller*, *memory*, *gesture detector* and calculation module for correlation and mean value. Detailed description of each function block is as follows.

A. Controller

The *controller* controls other internal modules of the *finger gesture detector*. It updates the data in the *sampling buffer* of the *memory*, when sensor data comes in with the *flag* signal. Also, it operates each core for arithmetic calculation of correlation and mean value, controlling the multiplexers in order to deliver appropriate data to the cores on a time-sharing basis. When all cores have completed the operation, the controller activates *the gesture detector* to output the gesture data.

B. Memory

The *memory* mainly stores coefficients information and the sampled sensor data. Since both correlation and mean value are calculated for 32 sample-windows, each axis of sensor data having 16bit data width occupies 512 bits of memory. Similarly, each coefficient data for correlation having 8bit data width occupies 256 bits of memory. The output of the allocated memory is transferred to the arithmetic cores through the multiplexers.

C. Correlation MAC and Mean ACCUM

These modules calculate correlation and mean values for gesture detecting. We focused on hardware-cost and power-efficiency rather than high-performance. Thus, the calculation core for cross-correlation is designed, exploiting Multiplier-Accumulator (MAC) unit. In this *correlation MAC* units, the correlation value is computed as shared multiplier architecture. When the correlation cores receive start flag from the controller, it initializes the accumulator register. Then the sensor signal in *the sampling buffer* and gesture coefficient in the *coefficient register* are allocated as operand and processed for one sample at a clock. When processing is completed, each of 12bit width correlation data is transmitted to the gesture detector. The mean calculation cores are designed as the accumulator unit. In the same manner with the correlation core, the accumulator register is initialized before operation. Then, the mean calculation cores receive and process total 32 samples of gyro data for one sample at a clock. Finally, mean data of 12bit width, which is the result of the accumulator unit, is transmitted to gesture detector.

D. Gesture detecter

This module plays a key role in the gesture detector. Based on the correlation data for the four gestures and mean data for two sensor signal, the *gesture detector* continuously outputs the gesture information. The gesture detector recognizes the gesture by comparing correlation data with defined threshold. As described in Section II, the mean values are monitored in order to prevent misrecognition with excessive correlation data when non-cyclical pattern signal is entered. When the average value exceeds threshold, the activation of the gesture is ignored.

IV. Implementation and Verification

We implemented the proposed finger gesture detector using Verilog HDL and mounted the digital logic circuit on FPGA. Figure 5 shows sensor data, correlation data, average data, and pattern recognition result when four patterns are inputted in the implemented environment. When the corresponded correlation data exceeds threshold with the cyclical pattern, the corresponding gesture is correctly activated. After 170th sample time, unintentional movements were detected. Because of the non-cyclic movements, the mean value is out of the defined range and the gesture is not activated even though the correlation value exceeds threshold. Finally, our finger gesture recognition system successfully detects the defined gestures. Additionally, our experiments of the 50 times have shown that the recognition rate is 88%, 80%, 84% and 92% according to the gestures of up, down, left and right,

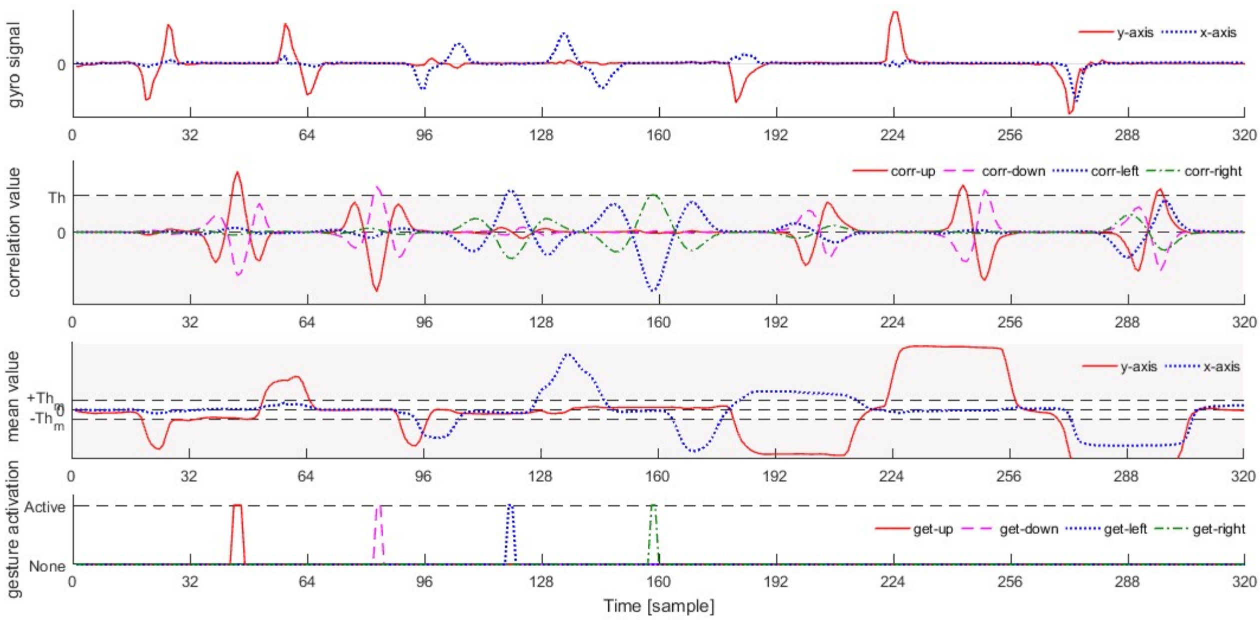

Figure 5. The contiuous signal variation, according to the correct gesture and irrelevant movement

respectively and the final average recognition rate was 86%.

V. Conclusion

We proposed real-time fingertip gesture recognition for embedded devices. In order to extract the physical movement of the fingertip, we exploited 3-axis gyro sensor and defined four gestures as up, down, left, and right. The correlation and mean feature were calculated by using hardware arithmetic units and used to spot and recognize cyclic-pattern in real-time. Our gesture recognition system was implemented on FPGA and we verified its feasibility with final average recognition rate of 86%.

Acknowledgment

This study was supported by the Research Program funded by the Seoul National University of Science and Technology (2017-0082).

References

[1] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta and K. Sharma, "A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors," in *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6425-6432, Aug.15, 2016.

[2] S. Patil, I. Bidari, B. Sunag, S. V. Gulahosur and P. Shettar, "Application of HMI technology in automotive sector," *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, Mysuru, 2016, pp. 322-324.

[3] T. Kasai and K. Takano, "Design of Sketch-Based Image Search UI for Finger Gesture," *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*, Fukuoka, 2016, pp. 516-521.

[4] J. H. Oh, J. K. Kim and S. E. Lee, "Design of read-out IC for wearable computing," International Journal of Computer Systems, Volume 03– Issue 12, December, 2016, pp. 666-669.

[5] T. Chakraborty, M. Nasim, S. M. B. Malek, M. T. H. Majumder, M. S. Saeef and A. B. M. A. A. Islam, "Low-cost finger gesture recognition system for disabled and elderly people," *2017 International Conference on Networking, Systems and Security (NSysS)*, Dhaka, 2017, pp. 180-184.

[6] P. Salunkhe and A. R. Patil, "A device controlled using eye movement," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016, pp. 732-735.

[7] L. Long, X. Fu, Honglei Zhu and T. Ge, "Finger gesture-based natural user interface for 3D highway alignment design in virtual environment," *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, Harbin, 2015, pp. 105-111.

[8] Z. Yan and R. W. Lindeman, "A multi-touch finger gesture based low-fatigue VR travel framework," *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, Arles, 2015, pp. 193-194.

[9] S. Khare, "Finger gesture and pattern recognition based device security system," *2015 International Conference on Signal Processing and Communication (ICSC)*, Noida, 2015, pp. 443-447.

[10] M. I. Quraishi, K. G. Dhal, J. P. Choudhury, P. Ghosh, P. Sai and M. De, "A novel human hand finger gesture recognition using machine learning," *2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing*, Solan, 2012, pp. 882-887.

[11] C. Quan and J. Liang, "A Simple and Effective Method for Hand Gesture Recognition," *2016 International Conference on Network and Information Systems for Computers (ICNISC)*, Wuhan, 2016, pp. 302-305.

[12] R. R. Itkarkar and A. V. Nandi, "A survey of 2D and 3D imaging used in hand gesture recognition for human-computer interaction (HCI)," *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, Pune, India, 2016, pp. 188-193.

[13] X. Dang, W. Wang, K. Wang, M. Dong and L. Yin, "A user-independent sensor gesture interface for embedded device," *2011 IEEE SENSORS Proceedings*, Limerick, 2011, pp. 1465-1468.

[14] S. M. Lee, S. D. Kim, J. H. Jang, S. M. Lee and S. E. Lee, "Design of an EMG signal recognition system for human-smartphone interface," *2015 International SoC Design Conference (ISOCC)*, Gyungju, 2015, pp. 337-338.

[15] F. T. Liu, Y. T. Wang and H. P. Ma, "Gesture recognition with wearable 9-axis sensors," *2017 IEEE International Conference on Communications (ICC)*, Paris, 2017, pp. 1-6.

[16] H. G. Doan, H. Vu and T. H. Tran, "Dynamic hand gesture recognition from cyclical hand pattern," *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, Nagoya, 2017, pp. 97-100.

[17] B. Noronha, S. Dziemian, G. A. Zito, C. Konnaris and A. A. Faisal, "“Wink to grasp” − comparing eye, voice & EMG gesture control of grasp with soft-robotic gloves," *2017 International Conference on Rehabilitation Robotics (ICORR)*, London, United Kingdom, 2017, pp. 1043-1048.