Journal of Fiber Bioengineering and Informatics 8:2 (2015) 337–345
doi:10.3993/jfbim00106

Hardware Based High Efficient Recognition of 3D Hand Gestures ⋆

Yi Liang [a], Liang Zhuo [b], Ning Chen [b], Cheng Cheng [b]
Ruizhi Li [b], Xinyan Gao [b,*]

[a] *School of Information Science and Technology, Guangxi University for Nationalities Nanning 530006, China*

[b] *G&S Labs, School of Software, Dalian University of Technology, Dalian 116620, China*

Abstract

This paper addresses the technical issues related to hand gestures generation and their real-time recognition. The arbitrary 3D gestures are generated by the Leap-motion Controller which detects and tracks the hands and fingers to acquire position and motion information. We combine Leap-motion sensor and hardware based pattern engine to make gesture recognition easier and propose an efficient recognition solution involving a neuron chip (named CM1K). In our experiment, we used one-finger gesture cases to demonstrate the efficiency of our solution. Experimental results showed that our solution owns a high accuracy in gesture data acquiring and only costs a few milliseconds in recognizing speed. We also considered the situation of similar gestures recognition and analyzed the causes of low matching rate from specific data.

Keywords: Gesture Recognition; Human Computer Interaction; Leap-motion; CM1K

1 Introduction

Gesture recognition, as one of the advanced nature communication means, is becoming a research focus in the filed of human computer interaction [4, 12] and playing key roles in a wide area of applications including sign language, gesture controlling [10], virtual reality, and other immersible games [8].

Especially, with the development and realization of virtual environment, the current human machine interaction approaches, such as mouse, joystick, keyboard and electronic pen are becoming more and more inefficient.

Gesture is the motion of hands or physical action, including temporal and spatial sequence information, can be used to convey sufficient meaningful messages by the users. In fact, gesture recognition is the process by which gesture made by the user is made known to the system [3].

⋆Project supported by the Fundamental Research Funds for the Central Universities (No. DUT14QY05).
*Corresponding author.
Email address: GSLabs@126.com (Xinyan Gao).

1940–8676 / Copyright © 2015 Binary Information Press & Textile Bioengineering and Informatics Society
June 2015

Hand gesture has the natural ability to represent ideas and actions easily, thus using these different gestures, being identified by human beings and interpreted to generate corresponding event, has the potential to provide a more natural interface to the computer system.

In past decades, there came many gesture recognition techniques and they had their advantages and disadvantages. For example, data glove [5], a sensor for hand motion and gesture recognition using motion sensors such as accelerometer, gyroscope, bend sensor, force sensor and so on. They are widely used in many applications, including virtual reality applications, biomechanics, robotics, and telecheric applications. However, the data glove a kind of wired technology [3] like electronics gloves cannot freely move in the room as they connected with the computer system via wire and limited with the length of wire. Through the use of sensors, these data gloves provide good results but they are extremely expensive to utilize in wide range of common application.

Data gloves are then replaced by optical markers. These optical markers project infra-red light and reflect this light on screen to provide the information about the location of hand or tips of fingers wherever the markers are wear on hand, the corresponding portion will display on the screen [7]. These systems also provide the good results but require very complex configuration. Later on some advanced techniques have been introduced such as image-based technique which requires processing of image features like texture, color and so on. But this technology is not very mature and the results could be different as skin tones and texture changes very rapidly from person to person and from one continent to other or someone under different illumination condition.

Although there are many ways to achieve gesture recognition, the improvement and innovation in this field is continuously made for years. All these approaches can not give overall consideration on either the accuracy, efficiency or convenience.

To resolve the mentioned problems in another way, this paper turns to the hardware based recognition and tries to make a good balance between accuracy and speed. Therefore, our solution combines Leap-motion Controller with CM1K chip together to construct a novel gesture capture and recognition platform.

2 Related Work

Leap-motion Controller is a tiny somatosensory controller for the PC and Mac which was first released by Leap Motion Company on Mar 19th, 2013. It can detect and track hands, fingers and finger-like objects reporting discrete position and motion. Its function is similar to Kinect and at present there has been some way of gesture recognition by using Kinect [6]. With its rapid popularization, the device has been widely applied in many related areas, such as dynamic gesture recognition, Arabic sign recognition [1], Robotic arm manipulation [2], and so on by researches and developers all over the world.

The CogniMem neural network chip named CM1K was developed by Guy Paillet which is actually a descendant of the ZISC (Zero Instruction Set Computer) chip manufactured by IBM until 2001. CM1K can be considered a high-speed recognition engine for artificial intelligence. Up to now, many companies and developers have integrated the CM1K chip in their designs, mostly to add recognition capabilities to embedded sensor boards. CM1K chip is a high-performance pattern recognition processor featuring a network of 1024 neurons operating in parallel. Also, the chip contains a recognition engine ready to classify a digital signal received directly from a

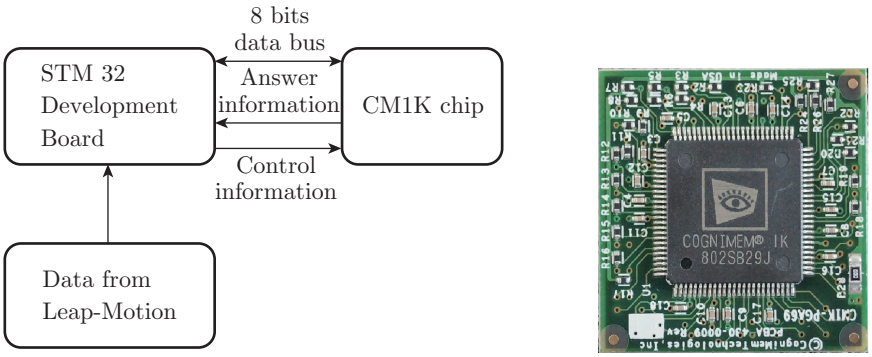
sensor. The efficiency of data processing is improved by tens of times compared to the similar chips before.

3 Recognition System

In this section, we will discuss the architecture of the target system for gesture acquisition and recognition.

3.1 System Architecture

As shown in Fig. 1, an embedded system connecting STM32 development board with CM1K chip, forms the necessary environment of experiments.



(a) Embedded system based on STM32 (b) CM1K chip

Fig. 1: The recognition system architecture

In our system, the Leap-motion Controller acts as trajectory data acquisition sensor and operates in a close proximity at a rate of 200 frames per second. It has a high position accuracy of 0.01 mm which is more than 100 times compared to the Kinect while the dimension is only $3 \times 1.2 \times 0.5$ inches with 0.1 pounds. The sensor uses two high precision infrared cameras and three LEDs to capture hand information within its active range.

However, it does not provide pictures of detected images. Its driver software processes the acquired data, extracts position and other information using complex mathematics. Leap-motion Controller can simultaneously track hundreds of thousands of targets (not used here point tracking technology), and because its accuracy is decided by the connected hardware. Leap-motion generally only occupies 2% performance of CPU. Using Leap-motion we can acquire the gesture data very easily and efficiently.

The SMT32 core reads and sends data to CM1K chip through 8-bit parallel data bus and communicates with the chip via I^2C bus, in order to control the process of learning and recognizing.

When the chip is in learning stage gesture data, the algorithm defines identifier firstly and CM1K chip matches the gesture with it. To make the results more accurate, we let chip learn more data sets or repeat steps of learning.

When recognizing the gesture, we use STM32 to send the data to the chip. Then the chip returns the identifier of the gestures which is the closest result. The algorithm inside CM1K chip

we used is based on KNN (k-Nearest Neighbor).

The flowchart of the whole process is shown in Fig. 2 and the main steps will be discussed in the following.

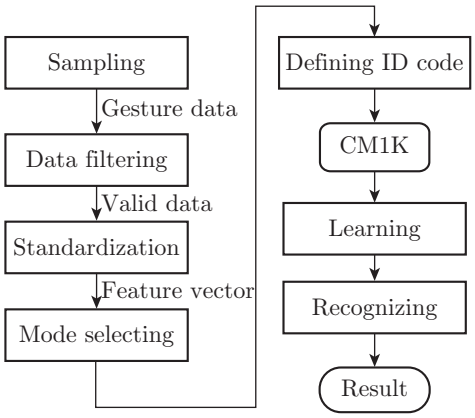


Fig. 2: Gesture recognition flowchart

3.2 Data Acquisition

Acquiring gesture data is the base for gesture recognition, the gesture data can refer to a single data or data sets. It must meet the following requirements:

- The data must have enough information which can depict a real gesture.
- The data must depict their difference clearly from the different gestures.
- All the gestures must have or can be transformed to one unified form which we can batch processing easily.
- Considering the recognition efficiency factor, the data must be as simple as possible.

As the Leap-motion controller tracks hands, fingers, and tools in its field of view, it provides the updating as a set, or frame, of data.

Each frame contains lists of the basic tracking data, such as the basic information of hands, fingers, and tools. For a continuous action, the data is contained in a set of frames returned by Leap-motion frame to frame. Each frame contains a lot of information, so choose some of them to distinguish different gestures. The Leap-motion system employs a right-handed Cartesian coordinate system (shown in Fig. 3). The origin is located at the center of the Leap-motion Controller.

For the similar gestures have the similar coordinate transformation of the action object and different gestures have obvious differences, we choose the coordinate information of a gesture object as the gesture data.

The data comes from many frames, and each frame contains a **Node** coordinate information (including tip position x, y, and z coordinate value of the fingers) of the whole action (shown in Fig. 4), so we put them together as a **NCCS** (Node Coordinate Change Set) to express the gesture.

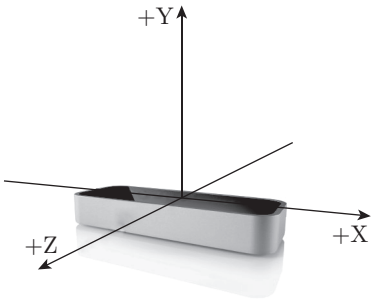


Fig. 3: Leap-motion coordinate system

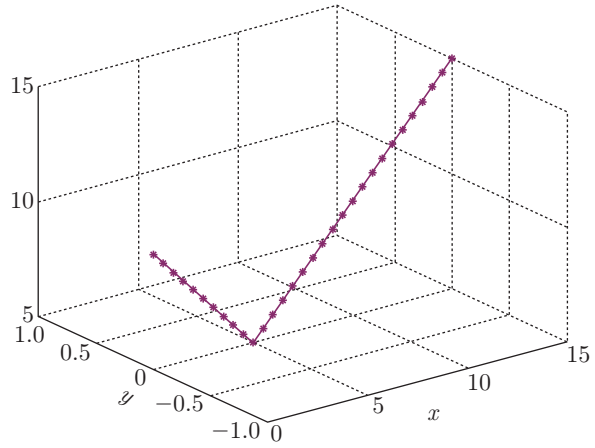


Fig. 4: Gesture trajectory with nodes

Before entering the next step, we filter raw data to valid data because some frames Leap-motion captured may be invalid. Meanwhile, before starting gesturing, Leap-motion may have captured the track of the hands moving to the gesture starting point and returned some frames. To solve this problem, the algorithm will set the conditions for recognition or just delete a small portion of the data in the front of **NCCS**.

Leap-motion returns frames continuously (about 60 fps), different **NCCS** of a gesture may have different length (the number of frame).

Here are some rules we used in this paper:

R1. Set the frame number needed to obtain quantitative data.

R2. Make changes for the frames obtained:

(a) If the frame number is less than needed, then add a frame between two frames obtained by interpolation method.

(b) If the frame number is more than needed, then delete a frame every n frames until acquiring the qualified data.

3.3 Data Standardization

In this step, we will change the form of the data set obtained from Leap-motion and make preparations for CM1K chip recognition. It includes the following steps:

(1) Reduction from 3D Space to 2D Plane

Even though gesticulated in a 3D space, mostly the same gestures could be expressed in a 2D plane. Those are called 2D gestures. Besides, consider the shaking of the hand when someone is gesticulating. This is an important step to reduce error.

This paper makes the gestures on the plane which is parallel to the XOY plane to simplify operation. So it just ignores the z values. Actually, the rules are obvious that as long as the gesture is not a 3D gesture (see samples in Fig. 5), it can be still acceptable if one of any three coordinate axis is ignored. Because its 'shape' do not change and just like the projection in 3D space (see samples in Fig. 6). This changes can be ignored to CM1K after a large quantity of data learning.

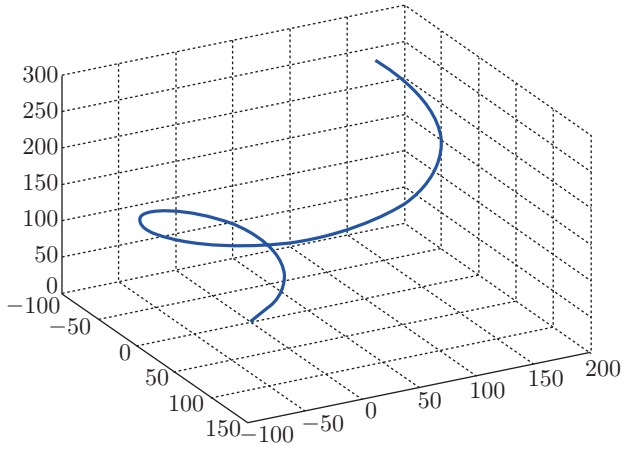


Fig. 5: 3D gesture sample

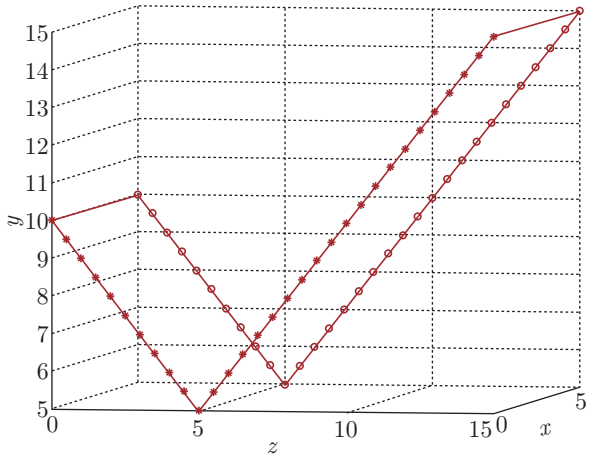


Fig. 6: Non-3D gesture's projection

(2) Standardization of 2D Gesture Data

2D gesture data standardization means for the same gestures using the same standard to measure them, it is to say for one type of gesture, the data having dealt with must become the same or similar in value.

For example, consider such a situation: as for the gesture "$\surd$", that can be a big one or a small one, that can be anywhere in its field of view. They have different coordinate values but actually these gestures are the same.

Having acquired 2D gesture coordinate data set, we change the data set by the following steps:

- Find the maximum and minimum value of X, Y axis value from the data set, set them to X_{min}, X_{max}, Y_{min}, and Y_{max} respectively.
- Use the following formulas to change the value of the data set:

$$X_{change} = \frac{X_{real} - X_{min}}{X_{max} - X_{min}} \times n \quad and \quad Y_{change} = \frac{Y_{real} - Y_{min}}{Y_{max} - Y_{min}} \times m$$

where n and m denote the range of the standardization area. This means that any gestures will be expressed in $n \times m$ area. The value range is between 0–255, which is limited by CM1K chip. The bigger the value is, the higher its accuracy will be. For the experiment, set (m, n) as $(255, 255)$ respectively.

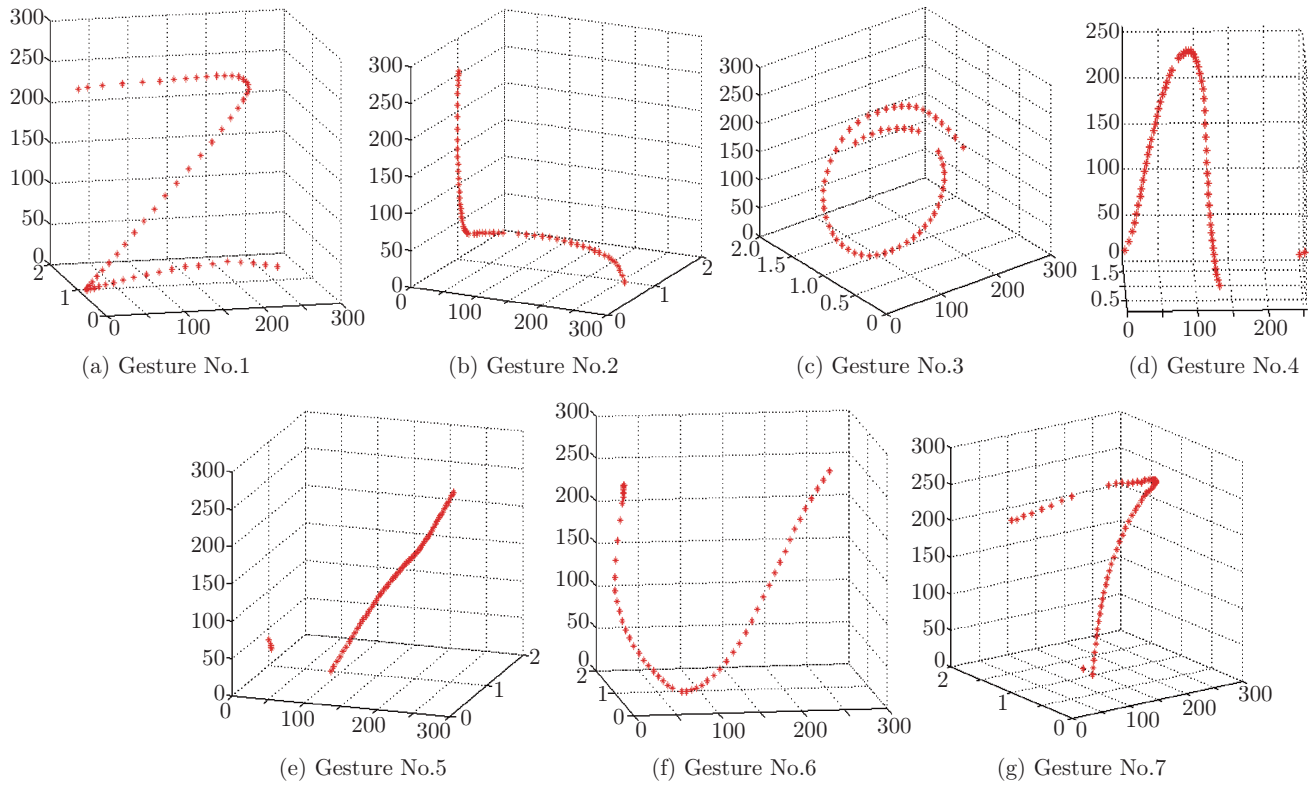
- Obtain the **CDS** (Changed Data Set).

After obtaining the **CDS**, the final step is to change the data set into a vector as the input value of CM1K chip.

4 Experiment and Results

We have implemented the previously discussed method and we have learned and tested 1400 sets of data with 7 different categories of gestures based on this method. By using these effective vectors of coordinate values, we can successfully recognize and classify all 7 classes of different hand gestures.

The experimental gestures reproduced by **Matlab** with the data obtained from Leap-motion sensor and processing, can be shown in Fig. 7. Experimental parameters are shown in Table 1. And each gesture has been assigned a unique number to denote itself. The different changed data set is always unique.



(a) Gesture No.1 (b) Gesture No.2 (c) Gesture No.3 (d) Gesture No.4

(e) Gesture No.5 (f) Gesture No.6 (g) Gesture No.7

Fig. 7: Experiment gestures

However, two hand gestures with similar hand shape could belong to the same categories of orientation and have similar data set just like 'V' vs '$\surd$' and '2' vs 'Z'.

The results of 1400 sets data tested through this method are shown in Table 2. It gives the success rate of over 96.6% approximately with average computation time of 0.986 ms for learning a gesture set and 1.171 ms for recognizing it. The method is based on simple coordinate based feature calculation which provides us with the comfort of implementation.

Table 1: Gestures and parameters

Gesture No.	Gesture Description	Number of Frame	(m, n) value
1	Arabic Numerals: “2”	60	255, 255
2	Letter: “L”	60	255, 255
3	Arabic Numerals: “0”	60	255, 255
4	Letter: “n”	60	255, 255
5	Oblique line, from left to right to right	60	255, 255
6	Letter: “v”	60	255, 255
7	Arabic Numeral: “7”	60	255, 255

Table 2: Hand gesture recognition results

Gesture No.	Learning set	Recognition set	Successful Cases	Recognition Rate	Learning Time (ms/each)	Recognition Time (ms/each)
1	100	100	98	98%	1.2 ms	1.2 ms
2	100	100	94	94%	1.1 ms	1.2 ms
3	100	100	95	95%	1.1 ms	1.2 ms
4	100	100	96	96%	0.9 ms	1.1 ms
5	100	100	95	95%	1.0 ms	1.3 ms
6	100	100	98	98%	0.8 ms	1.1 ms
7	100	100	100	100%	0.8 ms	1.1 ms
Total	700	700	676	96.90%	0.986 ms	1.171 ms

5 Conclusions and Future Work

In this paper we proposed a hardware based solution for hand gesture acquisition and recognition. Through this method we can define gestures by ourselves freely and obtain high efficiency and accuracy of recognition. The experimental results provide a promising prospect and it is especially suitable for size-sensitive and high performance embedded application scenarios. In summary, the strength includes simplicity, ease of implementation and unnecessariness of any significant amount of training or post processing. In future work, we will consider doing some useful optimization inspired from other work [11] and further implement full-hand gesture definition, acquisition and recognition without loss of performance.

References

[1] M. Mohandes, S. Aliyu, M. Deriche, Arabic sign language recognition using the leap motion controller, 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), 1-4 June 2014, 960, 965

[2] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, T. Bock, Intuitive and adaptive robotic arm

manipulation using the Leap Motion controller, Proceedings of 45th International Symposium on Robotics (ISR 2014) and the 8th German Conference on Robotics (ROBOTIK 2014), 2-3 June 2014, 1, 7

[3] Meenakshi Panwar, Hand gesture recognition based on shape parameters, 2012 International Conference on Computing, Communication and Applications (ICCCA), DOI: 10.1109/ICCCA.2012.6-179213, 22-24 Feb. 2012, 1-6

[4] Meenakshi Panwar and Pawan Singh Mehra, Hand gesture recognition for human computer interaction, Proceedings of IEEE International Conference on Image Information Processing (ICIIP 2011), Waknaghat, India, November 2011

[5] K. N. Tarchanidis, J. N. Lygouras, Data glove with a force sensor, IEEE Transactions on Instrumentation and Measurement, 52, 3, June 2003, 984, 989

[6] Youwen Wang, Cheng Yang, Xiaoyu Wu, Shengmiao Xu, Hui Li, Kinect based dynamic hand gesture recognition algorithm research, 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 26-27 Aug. 2012, 274-279

[7] Byungkyu Kang, M. Rodrigue, T. Hollerer, Hwasup Lim, Poster: Real time hand pose recognition with depth sensors for mixed reality interfaces, 2013 IEEE Symposium on 3D User Interfaces (3DUI), 16-17 March 2013, 171-172

[8] S. S. Rautaray, A. Agrawal, Interaction with virtual game through hand gesture recognition, 2011 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011

[9] Ji-Hwan Kim, Nguyen Duc Thang, Tae-Seong Kim, 3-D hand motion tracking and gesture recognition using a data glove, IEEE International Symposium on Industrial Electronics, ISIE 2009, 5-8 July 2009, 1013-1018

[10] Seong-whan Lee, Automatic gesture recognition, for intelligent human-robot interaction, 7th International Conference on Automatic Face and Gesture Recognition, FGR 2006, 2012

[11] Gritzman A., Batev T., Pantanowitz A., Improvements in feature vector selection and parameter optimization for continuous gesture recognition, 2013 IEEE International Conference on Cybernetics (CYBCONF), 2013

[12] Fakhreddine Karray, Milad Alemzadeh, Jamil A. Saleh, Mo N. Arab, Human-computer interaction: Overview on state of the art, International Journal on Smart Sensing and Intelligent Systems, Vol. 1, No. 1, March 2008, 137-159