

The Text Input System Using Spatial Fingertip Tracking Method

YANG-KEUN AHN, KWANG-SOON CHOI, YOUNG-CHOONG PARK

Realistic Media Platform Research Center

Korea Electronics Technology Institute

121-835, 8th Floor, #1599, Sangam-Dong, Mapo-Gu, Seoul

REPUBLIC OF KOREA

ykahn@keti.re.kr

Abstract: - This paper presents an approach to input Korean letters by using depth information originating from a single depth camera. An index finger is detected based on the hand area of a user, which is extracted from the depth information. This paper suggests a way of recognizing gestures generated by tracking information on the index fingertips. To this end, two main approaches for recognition are suggested. The first is to simplify a gesture. A gesture is simplified with the use of specific angles that are extracted by using angles between points extracted from the index finger. Based on this approach, the error rate is found to be lower than that of the case where the approach was not applied. The second approach presents a modified \$1 Recognizer. The existing \$1 Recognizer is too robust to angles and, accordingly, a limited number of pattern numbers can be input and often this causes confusion depending on the input pattern. In order to resolve this problem, this paper presents an approach of comparing angles between the starting point and the end point from the center of the pattern. An experiment showed improved performance based on the approach.

Key-Words: - Gesture, Hand Gesture, Gesture Recognition, Text Input System, \$1 Recognizer, Finger Detection, V-Touch

1 Introduction

Since PCs became available commercially, hardware elements, such as CPU processing, memory capacity, and graphic quality, have undergone dramatic development. However, the interface of these electronic devices is mainly occupied by hardware controllers including a keyboard and mouse. When using input devices, a user is forced to be positioned in a designated location and assume a posture with less degrees of freedom. Studies have been actively conducted on Human Computer Interaction (HCI), which enables electronic devices to recognize the natural voice or motion of a person and as a result provide information that a user is able to understand more intuitively, beyond inconvenient and static input devices and further simple in/output-based information technology (IT).

User Interface(UI)/User Experience(UX) involving entry of languages in text to electronic devices plays an important role in HCI fields, and many relevant studies have been conducted.

In order to replace input based on physical contact such as a keyboard, studies have been conducted on systems to enter text by touching a virtual plane in 3D space [1]. However, these systems have a disadvantage of limiting the user's position because

the virtual plane is static in 3D space and the user should touch the virtual plane while watching it displayed on a monitor screen or a Head Mounted Display (HMD) that visualizes the plane.

A method offered by the University of Oxford that inputs text by matching various hand postures to the alphabet [2] presents difficulties to the user as he or she should memorize a large number of patterns and make difficult hand postures. Furthermore, the method is weak at dealing with different hand shapes depending on people. Handwriting recognition for a touch pad investigated by a research team of Carnegie Mellon University [3] implemented text input with the use of an Artificial Neural Network. However, this recognition approach requires the collection of a massive amount of data to create a learning system and shows difficulties in teaching new patterns.

PenKII proposed in this paper is a system to input the sound unit of Korean letters or words after recognizing a uni-gesture of index fingertips. Before the recognition, patterns were simplified to eliminate anything that is incorrect at the time of input. Owing to invariable scale and rotation, the user has much higher degrees of freedom for entering the pattern in a recognizable 3D space. However, excessively robust rotation leads to difficulties in segmenting rotated patterns and

therefore a certain value is given to segment the patterns rotated to a certain extent. In addition, a uni-stroke pattern, instead of fixed 3D space, is used, thus achieving recognition only based on intuitive motion of the user, not on the display of gestures. As a template matching technique was used for the recognizer, a vast store of learning data is not needed and only an additional new template is required. New gestures can thus be easily registered and replaced according to the user's preference. Fig. 1 illustrates the overall structure of PenKII.

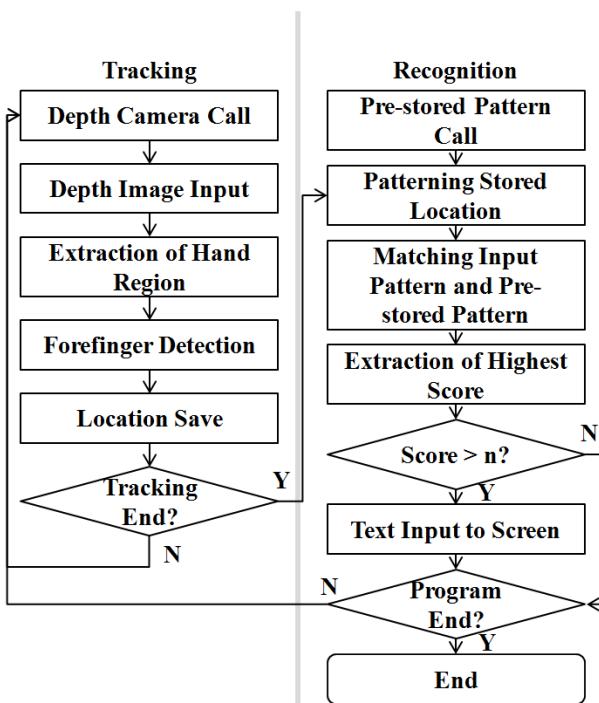


Fig. 1 Overall System Flow Map

2 Existing Studies

Recognition related to motion has been investigated by many research teams. The Hidden Markov Model[5] frequently has been used as an algorithm and Kinect[6] has been employed as equipment. The Hidden Markov Model is advantageously a probability model, capable of representing and recognizing temporal-spatial variation. However, it accompanies the issue of how to eliminate non-gesture patterns. Although Kinect SDK has an algorithm that is basically designed to identify and track the joints of the user, the algorithm is only applicable to large motion, not to detailed motion. Motion recognition using \$1 Unistroke Recognizer was also studied. This method ensures relatively high speed and accuracy but it cannot segment

patterns whose ratio value is different from that of a rotated pattern because it is too robust to angles.

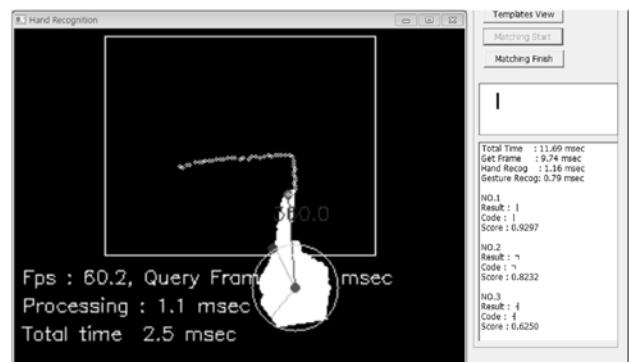


Fig. 2 Limits of \$1 Unistroke Recognizer

3 Extraction of a User's fingertips

Recognition related to motion has been investigated by many research teams. The Hidden Markov Model[5] frequently has been used as an algorithm and Kinect[6] has been employed as equipment. The Hidden Markov Model is advantageously a probability model, capable of representing and recognizing temporal-spatial variation. However, it accompanies the issue of how to eliminate non-gesture patterns. Although Kinect SDK has an algorithm that is basically designed to identify and track the joints of the user, the algorithm is only applicable to large motion, not to detailed motion. Motion recognition using \$1 Unistroke Recognizer was also studied. This method ensures relatively high speed and accuracy but it cannot segment patterns whose ratio value is different from that of a rotated pattern because it is too robust to angles.

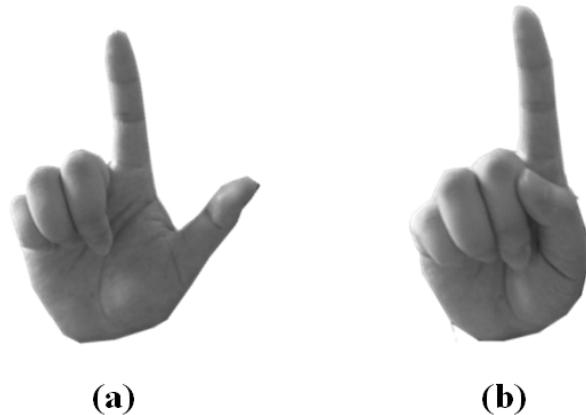


Fig. 3 Poses Used for Motions(Left : Non-input pose, Right : Input pose)

3.1 Index Finger Detection and Verification Algorithm

For index fingers detection, the area in the frontmost part should be first extracted from the depth image because the hand area is assumed to be placed in this region. In this extraction, an integral was used. Areas whose size is below a certain size are then removed via application of labeling, and among those remaining, the area in the frontmost part is left as a final candidate for the hand area. Fig. 4 illustrates a result of extracting a final candidate for the hand area.



Fig. 4 Result of Hand Area Extraction

Once the hand area is identified, the index finger should be detected and verified. For the index finger detection, the index finger is assumed to be located on the top of the area from the y-axis. As seen in Fig. 5, an X-axis area equal to 10 from the bottom centering on the reference point and an X-axis area equal to $10 + \text{thickness}$ are calculated.

The thickness refers to that of an area corresponding to 10 from the bottom centering on the reference point. These two thicknesses should be below 60 pixels based on an image of 640x480 and each length should be less than 1.5 times-longer length. If these conditions are met, it is recognized as the index; otherwise, it is recognized as noise. The process of extracting finger tips of the user is then completed and the next depth image is received.

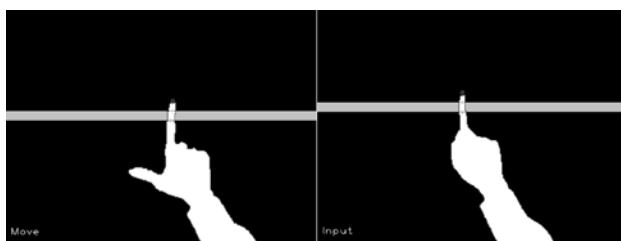


Fig. 5 Example of Successful Detection of Index Fingers



Fig. 6 Example of Unsuccessful Detection of Index Fingers

3.2 Algorithm for Thumb Detection

Thumb detection algorithm is carried out to check whether the thumb is attached or detached from the index finger, not to find a specific location. Fig. 7 displays a flow diagram.

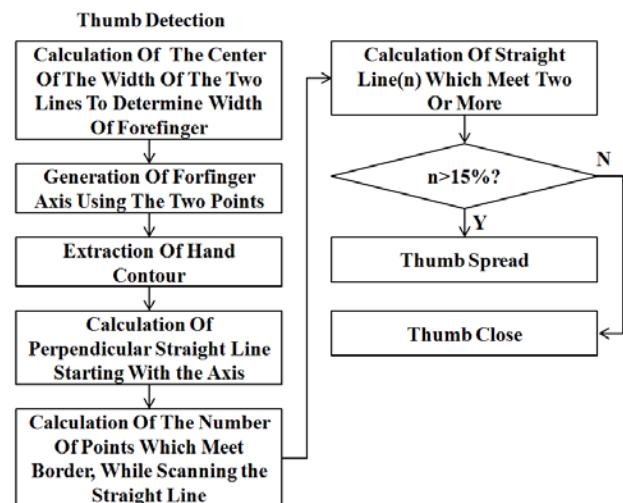


Fig. 7 Flow diagram of Thumb Detection Algorithm

Before starting the detection, the thicknesses used for the thumb detection are utilized to obtain the reference axis of the index finger. The center point of these thicknesses is calculated and then a straight line connecting to the center point becomes the index finger axis. Fig. 8 shows results of detecting the axis.



Fig. 8 Results of Index Finger Detection

Once the index finger's axis is found, the hand edge is detected. For the edge detection, Canny Edge detection is used. A straight line lying at right angles to the index finger axis is then created and the number of points where the line and edge intersect each other are compared. The thumb is recognized as being open for more than a certain rate and as being closed otherwise. As a result of a test, the appropriate rate is found as 15 %. Fig. 9 and 10 show relevant images.

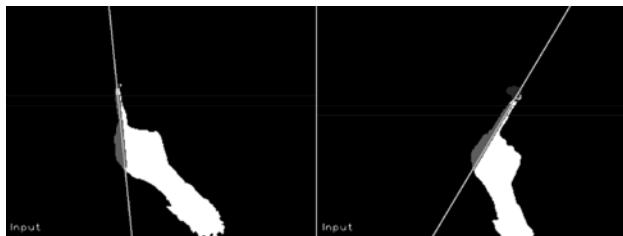


Fig. 9 Results in Case of Open Thumb(Below 1 intersection point)

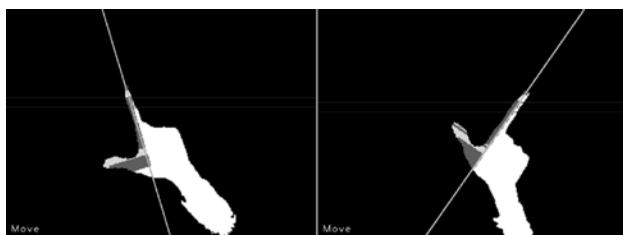


Fig. 10 Result in Case of Close Thumb(2 and More intersection points)

4 Gesture Recognition via Matching

Once gesture input is completed, matching between the gesture concerned and the gesture input is implemented. However, before the matching, an effort to simplify the gesture is made because the simplification can prevent any input error that a user may make when inputting the gesture. Additionally, the simplified gesture contributes to an increased recognition rate. After the simplification, a modified \$1 Unistroke Recognizer is used to calculate the matching rate with the stored template data. If the best matching rate exceeds a certain level of probability, the concerned is then selected to end the recognition.

4.1 Simplification of Gesture

A vector among points constituting a gesture is used for simplification of gesture. After calculating the

vector between the previous point and the current point, an angle gap between the present and previous vectors is calculated. If this angle is bent to more than a certain extent, it will become a branch point, which is to be saved as a result. Other points are passed over because they do not have specific characteristics and a branch point, a start point, and an end point are all saved to complete the simplification of the gesture. Fig. 11 displays relevant results.

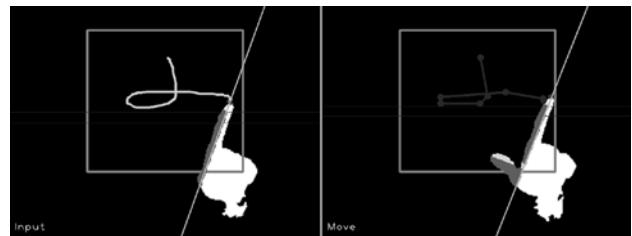


Fig. 11 Results of Gesture Simplification

4.2 Modified \$1 Unistroke Recognizer

The existing Recognizer[7] is too robust to rotation and size and, as a result, other patterns, except a rotated pattern pr ratio value, cannot be separated. For example, in the case of '入' and '冂', overall routes are identical but only rotation factors are different. In this case, the patterns could not be separated. In order to solve this problem, this paper offers the modified \$1 Unistroke Recognizer. Before starting the application of the Recognizer, the centers of each pattern are calculated; by using the centers, the vector between the center point and the start point and the vector between the center point and the end point are calculated. When the angles of these vectors are similar, recognition is started; otherwise, it is not started. Fig. 12 illustrates examples of this Recognizer and Fig. 13 shows a flow diagram.



Fig. 12 Examples of Modified \$1 Unistroke Recognizer

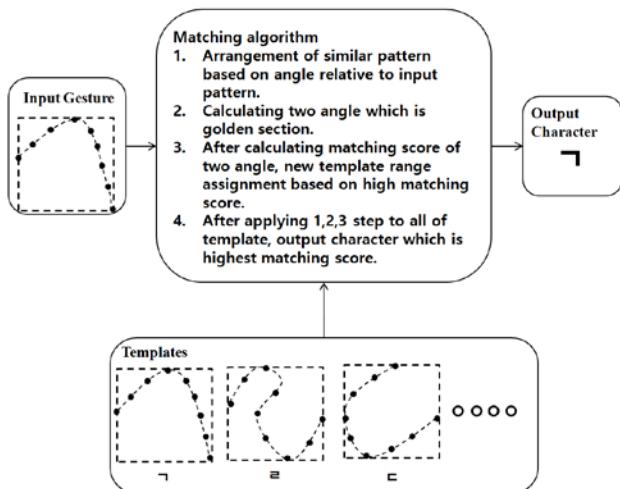


Fig. 13 Flow Diagram of Modified \$1 Unistroke Recognizer

5 Experiment Results

An experiment was conducted under the environment where a depth camera DS325[5] is connected by 60FPS to a PC equipped with a CPU of Intel(R) Core(TM) i7-2600K 3.4GHz and 8Gbyte memory (Fig. 14).

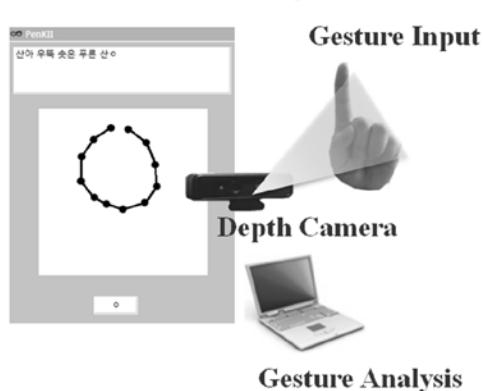


Fig. 14 Configuration of System Hardware

Software processing time is described in [Table 1]. A majority of time was consumed acquiring the depth image while only 10msec or less was consumed for recognizing the hand pose and matching templates. The template matching time was found to increase in proportion to the number of templates, number of sampling points, and range of golden section search start. Instead of using the golden section search, search of the trisection optimum matching rate was applied, consequently finding that 1.5 times greater time was spent.

Table 1 Processing Time

Type	Processing Time(msec)
Depth Image Acquisition	17 msec
Hand Pose Recognition	3 msec
Template Matching	7 msec
Total	27 msec

In order to verify the details of this paper, the Korean alphabet mapping pattern route was defined and 24 templates normalized with 64 points were registered and tested, as seen in Fig. 15.

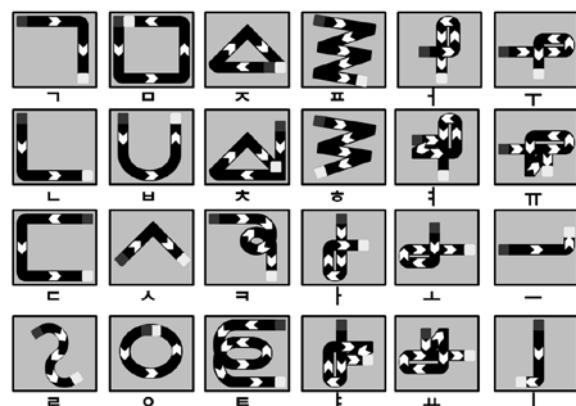


Fig. 15 Table of Korean Alphabet Mapping Pattern
Route

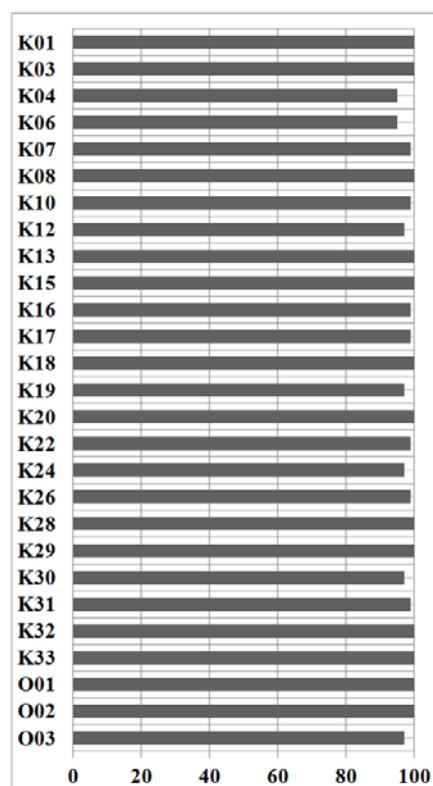


Fig. 16 Results of Recognition Measurement

A total of 10 experimenters who had practiced with the system using the pattern in Fig. 15 for 2 hours were made to repeat inputting all patterns 10 times. The results seen in Fig. 16 were thereby obtained. Significantly, almost a 100% recognition rate was achieved when the start and end positions were matched. On the other hand, errors were found when the start and end points were not matched, due to specific reasons such as late input. Consequently, the total recognition rate reached 99.27%. Fig. 17 shows the experimental environment for the test.



Fig. 17 PenKII System Experiment Environment

6 Conclusion

This paper presented a Natural User Interface (NUI) system that allows the hand of the user to input the Korean alphabet without a separate controller, by using a depth camera in 3D space. The system made it possible to input geometric pattern routes regardless of hand location in 3D coordinate space measured by a depth camera, thus improving convenience of use. In addition, as the user did not need to check the trace of the pattern route, the input speed could be increased. Moreover, unlike other learning techniques, the system enabled the user to easily register a new pattern route as a template. A more efficient system that reduces the time taken for start and end poses at the time of inputting the pattern route will be offered in the future.

References:

- [1] M. Tosas and B. Li, "Virtual Touch Screen for Mixed Reality," Proc. European Conference on Computer Vision, Lecture Notes in Computer Science, Prague, Czech Republic, Vol. 3058, May 2004, pp. 48-59.
- [2] Ray Lockton and A.W. Fitzgibbon "Hand Gesture Recognition Using Computer Vision," University of Oxford, Balliol College.
- [3] S. Jaeger and S. Manke and J. Reichert and A. Waibel "Online handwriting recognition : the NPen++ recognizer," IJDAE 2001.
- [4] Jacob O. Wobbrock and Andrew D. Wilson and Yang Li "Gesture without Libraries, Toolkits or Training : A 1\$ Recognizer for User Interface Prototypes," ACM UIST'07, , Newport, Phode Island, USA, October 7-10, 2007.
- [5] FS Chen, CM Fu, CL Huang "Hand gesture recognition using a real-time tracking method and hidden markove model.", Image and Vision Computing, 2003.
- [6] Byeongjun Jo, Hayoung Chang, and ByeongTak Chang "Motion Recognition and Categorization using Kinect Sensor Data," Autumn Workshop Journal of Korea Information Science Society , 2012.
- [7] Wobbrock, J.O., Wilson, A.D. and Li, Y. (2007). Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07). Newport, Rhode Island (October 7-10, 2007). New York: ACM Press, pp. 159-168.
- [8] <http://www.softkinetic.com> , 2013/08/29.