# The Hand Gesture Recognition System Using Depth Camera

Ahn,Yang-Keun
VR/AR Research Center
Korea Electronics Technology Institute
Seoul, Republic of Korea
e-mail: ykahn@keti.re.kr

Park,Young-Choong
VR/AR Research Center
Korea Electronics Technology Institute
Seoul, Republic of Korea
e-mail: ycpark@keti.re.kr

***Abstract*— This study suggests a method for hand gesture recognition using a depth camera in a smart device environment. The hand gesture recognition can be made through the detection of fingers or the recognition of a hand. For the detection of fingers, the hand skeleton is detected through Distance Transform, and the finger detection is made by applying the Convex Hull algorithm. The hand recognition is done by comparing a newly recognized hand gesture with already learned data using the Support Vector Machine (SVM). For this, the hand's center, finger length, hand axis, axis of fingers, arm center, etc.. are reviewed. After recognition of a hand gesture, the corresponding letter is displayed. For the evaluation of the proposed method, an actual smart device system was implemented for experiments.**

***Keywords-Hand Gesture; Gesture Recognition; Text Input System; Sign Language; Sign Language Recognition.***

## I. INTRODUCTION

Nowadays, with the growth of the mobile and smart TV industries and the development of smart devices, smart equipment and devices can commonly be found in diverse places. The growth potential of these markets has motivated some leading companies to compete for the acquisition of competitive smart device technologies, further expanding their use and availability. For example, Google has acquired Flutter, and Intel has purchased Omake Interactive, while Microsoft has developed Kinect jointly with Primesense.

Recently, a new product called Leap Motion has been developed, which addresses the growing demand for an efficient input method for smart devices. With the increasing use of smart devices, the amount of information displayed on a screen has steadily grown. Generally, current technologies use a remote controller or mobile devices for input. However, these methods are not convenient, in that users have to carry such devices all the time. To resolve such inconvenience, new input methods based on the use of hand gestures, like the one developed by Leap Motion, have begun drawing significant attention.

The hand recognition methods that have been proposed to input text on a screen include: recognition based on the learning of hand gestures using a neural network [1]; recognition by extracting a finger candidate group after removing the palm area [2]; teaching by extracting the characteristics of a hand using Support Vector Machines (SVM) [3]; recognition of the fingers by opening a hand [4]; depth-based hand gesture recognition [5][6][7][8]. In the case of the neural network, recognition is possible on the condition that a hand moves from a fixed position, a limitation which causes many constraints in consumer use. Furthermore, this method only allows a limited number of inputs, and therefore is not suitable for text input. The method of removing the palm area allows free movement, but it is difficult to distinguish separate fingers when they are put together. The method of recognizing the fingers after opening a hand is a good algorithm for hand recognition, but the number of hand gesture patterns is limited. Finally, the method of teaching the characteristics of a hand using SVM is considered an efficient approach, but it also has a shortcoming in terms of the number of hand gesture patterns that can be recognized.

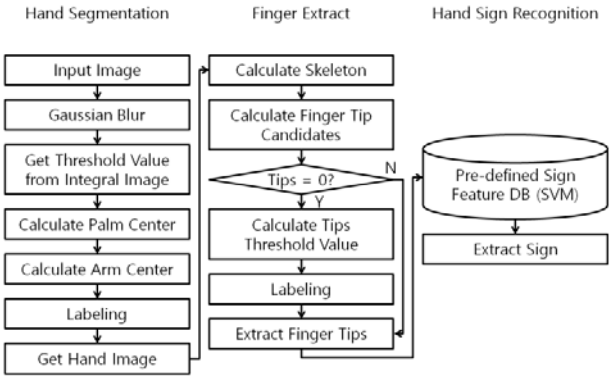


Figure 1. Flowchart of system for hand gestures recognition

The present study proposes a process to address these limitations. This method first detects a hand using a depth value. To accurately separate and recognize fingertips, a recognition based on a system similar to sign language is suggested, operated by reviewing the length and angle of the fingers and the angle of a hand. For recognition, the area of a hand is detected relatively accurately using a single infrared camera, the area and characteristics of a finger area are detected through thinning, and input data are matched against already learned data using SVM. The number of hand gesture patterns used in this study is about 30.

This study suggests a system which consists of the following three parts: of Hand Segmentation, Finger

 ISBN: 978-1-61208-538-8 

Extract and Sign Recognition. Figure 1 shows the flowchart of the process adopted for the system.

## II. HAND DETECTION

When a depth image is input, a smoothing operation is performed to remove noise. The Gaussian kernel is known as an effective smoothing method for noise elimination. Then, objects are separated using the binarization technique. Infrared lighting is applied while using an infrared camera, and the distance between objects is expressed in different brightnesses, making binarization possible. Subsequently, an integral image is produced to calculate a threshold value (T). As shown in (1), using the integral image, the average depth value is calculated, which amounts to a window of size $w$. Here, the value of $w$ is 20.

$$S(x,y) = \frac{\begin{array}{c} G(x+w,y+w) - G(x+w,y-w) - \\ G(x-w,y+w) + G(x-w,y-w) \end{array}}{(2w+1)^2} . \quad (1)$$

In (1), $G(x,y)$ means an integral image, and $S(x,y)$ signifies an average of depth values located within the $w$ area on the basis of an $(x,y)$ coordinate. T is a value obtained by adding 50 to the minimum value of $S(x,y)$.

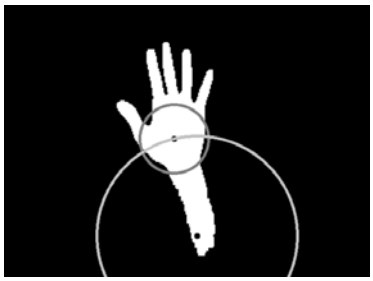

Figure 2. Hand center, arm center, and distances

Figure 3. Detection of hand area

After extracting a candidate group for the hand, the arm part is eliminated to display the hand part more accurately. For the removal of the arm part, the palm center ( $P(x,y)$ ) and palm area (L) are calculated using Distance Transform ( $D(x,y)$ ). The arm center ( $A(x,y)$ ) is calculated to be an average of the areas with a depth image value between T and T+50. Based on the above, the Euclidean distance ( $Eu(P(x,y),A(x,y))$ ) of $A(x,y)$ and $P(x,y)$ is calculated. On the basis of $A(x,y)$, the depth values within $Eu$ are calculated as the arm part. The palm part is what is within L on the basis of $P(x,y)$. Figure 2 shows the arm center and removed area as well as the palm center and palm area. Figure 3 shows the resulting hand detection image.

## III. FINGER DETECTION AND HAND GESTURE RECOGNITION

The system proposed by this study uses two different methods for finger detection: thinning and application of a minimum depth value.

Figure 4. Two cases of finger detection(Left: ㄷ, Right: ㅓ)

These two methods are used together because, as shown in Figure 4, it is not possible to detect all of the fingers of hand gestures of the sign language using only one method. For example, referring to [ㄷ] in Figure 4, finger detection can be done through thinning only. In contrast, in the case of [ㅓ], finger detection can be made by using a minimum depth value, but not through thinning.

### *A. Finger Detection Using Thinning*

To detect fingers, the hand skeleton needs to be identified first. Compared with the hand contour method applied previously, the method of detecting the hand skeleton offers some advantages. For example, the fingertips can be identified more accurately, and the fingers can also be detected more easily. To calculate the hand skeleton, as shown in Figure 5, an image of the hand part is gained using Distance Transform.

Figure 5. Result of Distance Transform using Histogram Equalization

After applying Distance Transform, the hand skeleton ( $Sk(x,y)$ ) is detected. (2) shows the algorithm for skeleton detection.

$$Sk_1(x,y) = \begin{matrix} if, D(x,y) \geq L/10 & 1 \\ else, & 0 \end{matrix}$$
$$c = 0, if\{D(x,y) < D(x+dx, y+dy)\}, c = c+1$$
$$Sk_2(x,y) = \begin{matrix} if, c \leq 2 & 1 \\ else, & 0 \end{matrix} \quad . \quad (2)$$
$$Sk(x,y) = \begin{matrix} if, Sk_1(x,y) = 1 \,\&\, Sk_2(x,y) = 1 & 1 \\ else, & 0 \end{matrix}$$

In (2), dx and dy signify a 3 x 3 mask and have values between –1 and 1. $c$ is a variable for counting cases where an adjacent pixel value ( $D(x+dx, y+dy)$ ) is larger than a current pixel value ( $D(x,y)$ ). If the value of $c$ is 3 or greater, that case is ignored since it does not form a line. $Sk(x,y)$ is recognized as a pixel (i.e., skeleton) when both conditions of $Sk_1$ and $Sk_2$ are satisfied. Figure 6 shows the result of hand skeleton detection and the palm part.

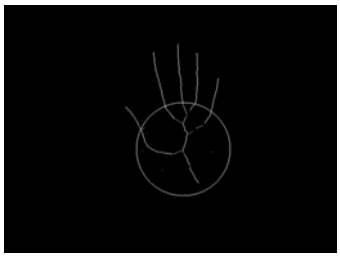

Figure 6. Display of hand skeleton and palm area

When the hand skeleton is detected, the fingertips are identified for the detection of fingers. For this, the Convex Hull(C) algorithm is applied, which is shown in (3).

$$C \equiv \{\sum_{j=1}^{N} \lambda_j p_j : \lambda_j \geq 0 \text{ for } all \ j \,\&\, \sum_{j=1}^{N} \lambda_j = 1\} \quad . \quad (3)$$

In (3), $p_1, ..., p_N$ means the locations of $Sk(x,y)$, and N is the number of the pixels of $Sk(x,y)$. Figure 7 shows a candidate group of fingertips when the Convex Hull(C) algorithm is applied.

When the Convex Hull(C) algorithm is applied, some areas which are not the fingertips are recognized as if they are fingertips. To resolve this, such areas are removed if they are found to belong to the palm part identified before. Figure 8 shows the fingertip parts after eliminating the irrelevant areas.

When the fingertips have been detected, to detect the fingers reverse tracking is started from the fingertips to the palm. The reverse tracking is done using a recursive function from the detected fingertips to the palm until no skeleton is found.

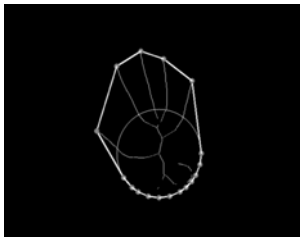

Figure 7. Application of Convex Hull

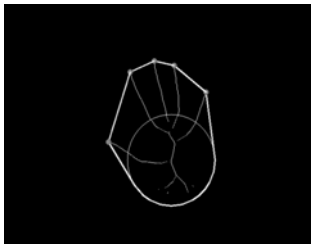

Figure 8. Detection of fingertips

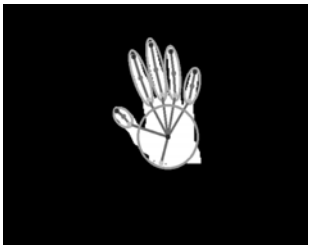

Figure 9. Detection of hand characteristics

The point which is closest to the skeleton around the middle part of a finger is recognized as the middle phalanx of a finger. Figure 9 shows the result of finger detection and the characteristics of a hand.

### *B. Finger Detection using Minimum Value*

It is assumed that the thinning-based finger detection fails if there is no recognized hand shape using the thinning technique. If the finger detection through thinning fails, an attempt is made to detect fingers based on the minimum value (i.e., the closest distance to the camera) of a depth image.

Figure 10. Finger detection using minimum value( ∃ )

 ISBN: 978-1-61208-538-8 

For this, first, a minimum depth value (Dm) should be gained. The threshold value (Tm) for finger detection is obtained by adding 55 to the Dm value. The reason for adding 55 is based on experimental experiences. Now, the binarization is complete, and a candidate area of fingertips is detected through labeling. If this area has a size of 1/2 or more of the palm part, that area is ignored. Figure 10 shows the result of this process.

### C. Hand Gesture Recognition

Basically, the recognition of hand gestures is done using the SVM. Based on data already learned, a newly input hand gesture can be recognized. For recognition, the necessary input data are entered according to the detection method. For finger detection through thinning, the hand center, palm size, axes of arm and palm, finger length, and axis of fingers should be offered. In the case of finger detection using a minimum value, the number of fingertips, area size and ratio of width to height of the area needs to be given. For fast learning, the linear SVM was adopted. However, as some errors were found, some factor values were changed, creating a more efficient SVM detector.

## IV. SYSTEM CONFIGURATION

To evaluate the text input performance of the system proposed by this study, which is based on the recognition of hand gestures, an actual text input system SignKII was implemented.

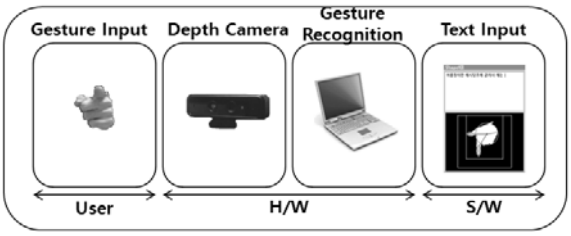


Figure 11. Diagram of gesture recognition system

The system configuration includes the input of a hand gesture by a user, capturing the input image using an infrared camera, analysis of the hand gesture by a gesture recognition module, and display of the result on a keypad. Figure 11 shows the configuration and process flow of the system.

### A. Hardware Configuration

Figure 12 shows the hardware configuration of SignKII, which includes: an LED TV, used as a display device, positioned at eye level; an infrared camera for image input, located under the LED TV; and a desktop PC, used as a gesture analysis module, connected to the camera through a USB interface as well as to the LED TV through the output module and HDMI.

### B. Software Configuration

Figure 13 shows the software configuration of the SignKII system, which includes: a main screen for the display of the input image (upper middle); binarization screen for the display of binarization results and a detection screen for the display of finger detection and characteristics (right upper); keyboard input results (left); and input examples (lower middle, right lower). The system performance has been improved by presuming empirical parameters using SignKII software. A threshold value of 10% of the value obtained when the user took the gesture of the designated character is designated as the parameter threshold value.


Figure 12. Gesture recognition system GUI

### C. Input Configuration

Figure 14 shows the hand gestures for the input of consonants.

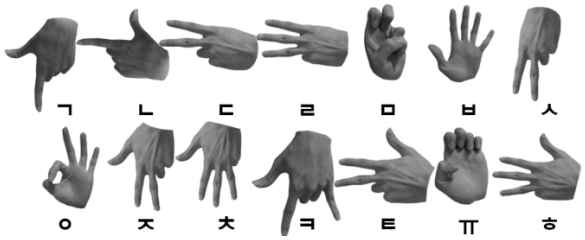

Figure 13. Examples of SignKII consonant input

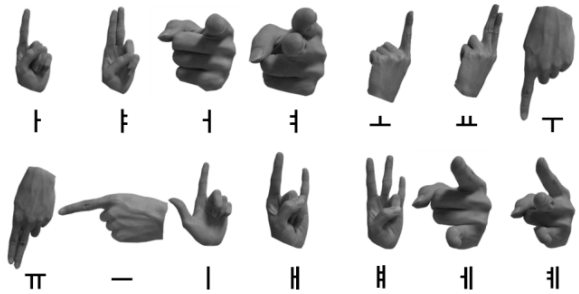

Figure 14. Examples of SignKII vowel input

Figure 15 shows the hand gestures for the input of vowels as applied in the SignKII system.

## V. EXPERIMENT RESULTS

In this paper, we experimented hand gesture recognition rates independently, because there were no korean input system that we could compare our system before. The development environment is as follows: Window7 OS, Visual Studio 2010, and MFC. The hardware configuration includes: DS325 infrared lighting

 ISBN: 978-1-61208-538-8 

camera of SoftKinetic, HDMI interface display, desktop PC Intel i7-2600k CPU and 3.48GB. As for the S/W performance, the distance range from a camera is from 11cm to 30cm, and the optimal distance for gesture recognition is 20cm±5cm. Figure 16 shows the signKII system.

Figure 15. SignKII system

TABLE I. CHANGE OF RECOGNITION RATES ACCORDING TO HAND GESTURE AND ANGLE

| Gesture | -10 | -5 | 0 | 5° | 10 |
|---|---|---|---|---|---|
| ㄱ | 100% | 100% | 100% | 0% | 0% |
| ㄴ | 100% | 100% | 100% | 0% | 0% |
| ㄷ | 0% | 100% | 100% | 100% | 0% |
| ㄹ | 0% | 100% | 100% | 0% | 0% |
| ㅁ | 100% | 100% | 100% | 0% | 0% |
| ㅂ | 100% | 100% | 100% | 100% | 100% |
| ㅅ | 0% | 100% | 100% | 100% | 0% |
| ㅇ | 50% | 100% | 100% | 50% | 0% |
| ㅈ | 100% | 100% | 100% | 100% | 100% |
| ㅊ | 100% | 100% | 100% | 100% | 0% |
| ㅋ | 100% | 100% | 100% | 100% | 100% |
| ㅌ | 0% | 100% | 100% | 100% | 100% |
| ㅍ | 100% | 100% | 100% | 100% | 100% |
| ㅎ | 0% | 100% | 100% | 100% | 100% |
| ㅏ | 100% | 100% | 100% | 0% | 0% |
| ㅑ | 100% | 100% | 100% | 0% | 0% |
| ㅓ | 100% | 100% | 100% | 100% | 100% |
| ㅕ | 100% | 100% | 100% | 100% | 100% |
| ㅗ | 0% | 0% | 100% | 100% | 100% |
| ㅛ | 0% | 0% | 100% | 100% | 100% |
| ㅜ | 0% | 0% | 100% | 100% | 100% |
| ㅠ | 0% | 0% | 100% | 100% | 100% |
| ㅡ | 0% | 100% | 100% | 100% | 100% |
| ㅣ | 100% | 100% | 100% | 100% | 100% |
| ㅐ | 100% | 100% | 100% | 100% | 100% |
| ㅒ | 0% | 100% | 100% | 100% | 100% |
| ㅔ | 100% | 100% | 100% | 100% | 100% |
| ㅖ | 100% | 100% | 100% | 100% | 100% |

Experiments were conducted in such a way that one user performed each gesture 100 times. Considering that hand gesture recognition is sensitive to the rotation of a hand (with the rotation of a hand, a totally different recognition result can be shown), experiments were performed mainly in connection with rotation. For example, the difference between ‘ㄱ′and ‘ㄴ′can be recognized due to hand rotation despite the same hand gesture. Table 1 shows the recognition results.

## VI. CONCLUSION

This study proposed an algorithm for improved hand gesture recognition based on previous studies. Based on experiments, the SignKII system was implemented. The results of the experiments demonstrated that recognition rates were very high even though the performance was affected at some hand angles. Future research will focus on more efficient and easier recognition based on hand motions as well as hand gestures.

## REFERENCES

[1] C. Nolker and H. Ritter, “Visual Recognition of Continuous Hand Postures,“ IEEE Transactions on Neural Networks Neural Networks, vol. 13, pp. 983-994, Jul. 2002.

[2] Y. Fang, K. Wang, J. Chen, and H. Lu, "A Real-time Hand Gesture Recognition Method," Multimedia and Expo, 2007 IEEE International Conference on, Jul. 2007.

[3] P. Suryanarayan, A. Subramanian, and D. Mandalapu, "Dynamic Hand Pose Recognition using Depth Data," Pattern Recognition (ICPR), 2010 20th International Conference on, Aug. 2010.

[4] Z. Ren, J. Yuan, and Z. Zhang, "Robust Hand Gesture Recognition Based on Finger-Earth Mover’s Distance with a Commodity Depth Camera," Proceedings of the 19th ACM international conference on Multimedia, pp. 1093-1096, Nov. 2011.

[5] C. Wang, Z. Liu, S. C. Chan, "Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera," IEEE Transactions on Multimedia, vol. 17, pp.23-39, 2015.

[6] S. Jadooki, D. Mohamad, T. Saba, et al., "Fused features mining for depth-based hand gesture recognition to classify blind human communication," Neural Computing and Applications, 2016.

[7] W. L. Chen, C. H. Wu, C. H. Lin, "Depth-based hand gesture recognition using hand movements and defects," International Symposium on Next-Generation Electronics (ISNE), 2015.

[8] C. H. Wu, W. L. Chen, C. H. Li, "Depth-based hand gesture recognition," Multimedia Tools and Applications, 2016.

 ISBN: 978-1-61208-538-8 