Real-Time Recognition Method of Counting Fingers for Natural User Interface

Doyeob Lee, Dongkyoo Shin, and Dongil Shin
Department of Computer Engineering, Sejong University
98 Gunja-Dong, Gwangjin-Gu, Seoul 143-747 – South Korea
[e-mail: dy03615@gmail.com. {shindk, dshin}@sejong.ac.kr]
*Corresponding author: Dongkyoo Shin

Received November 28, 2015; revised February 8, 2016; accepted March 26, 2016;
published May 31, 2016

Abstract

Communication occurs through verbal elements, which usually involve language, as well as non-verbal elements such as facial expressions, eye contact, and gestures. In particular, among these non-verbal elements, gestures are symbolic representations of physical, vocal, and emotional behaviors. This means that gestures can be signals toward a target or expressions of internal psychological processes, rather than simply movements of the body or hands. Moreover, gestures with such properties have been the focus of much research for a new interface in the NUI/NUX field. In this paper, we propose a method for recognizing the number of fingers and detecting the hand region based on the depth information and geometric features of the hand for application to an NUI/NUX. The hand region is detected by using depth information provided by the Kinect system, and the number of fingers is identified by comparing the distance between the contour and the center of the hand region. The contour is detected using the Suzuki85 algorithm, and the number of fingers is calculated by detecting the finger tips in a location at the maximum distance to compare the distances between three consecutive dots in the contour and the center point of the hand. The average recognition rate for the number of fingers is 98.6%, and the execution time is 0.065 ms for the algorithm used in the proposed method. Although this method is fast and its complexity is low, it shows a higher recognition rate and faster recognition speed than other methods. As an application example of the proposed method, this paper explains a Secret Door that recognizes a password by recognizing the number of fingers held up by a user.

Keywords: Kinect, Hand Gesture, Hand Gesture Recognition, Depth Image, NUI/NUX

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A01059253).

http://dx.doi.org/10.3837/tiis.2016.05.022 **ISSN : 1976-7277**

1. Introduction

Humans start, maintain, develop, and terminate relationships through interpersonal communication [1]. Interpersonal communication is an interactive process that involves one person sending a message to another. Interpersonal communication can be divided by the methods used to communicate. The sharing criterion is the use of verbal or non-verbal elements. Verbal communication involves the use of words or writing, and non-verbal communication involves body movements such as winking, shaking hands, and laughing.

We often use speech as a verbal element, along with gestures as a non-verbal element, during everyday conversations. This can provide much more accurate communication than dialog that does not use gestures because a human collects most of their information through sight and hearing, and non-verbal elements prompt these senses [2].

Today, the use of non-hardware interfaces has been generalized through the development and spread of various image display devices. A hardware interface uses a physical device such as a mouse or keyboard to interact with a computer. On the other hand, a non-hardware interface interacts with a computer using a person's body rather than a physical device. Thus, a non-hardware interface is easy to use and easy to learn. This is called a NUI (Natural User Interface) or NUX (Natural User eXperience). Research on an NUI/NUX has been actively pursued in the fields of HCI (Human Computer Interaction) and HRI (Human-Robot Interaction) [3]. In particular, many studies that define an interface by recognizing gestures have been conducted for interaction with a computer. Because gestures are simple and intuitive, they can easily be used to control a computer [4].

Studies on gesture recognition have been performed using a variety of equipment. A data glove and camera are prime examples. A data glove is a multimedia input device used for interaction with a computer. The location is entered on the computer by a sensor mounted in the glove when it is worn on a hand. Thus, it is possible to accurately measure the shape of the hand movement through the sensor in the date glove [5]. However, this limits the movement of the hand, requires additional correction, and is expensive. A recognition method using a video camera is the method that is considered here. However, separating the background and hands has not been easy with this method. If there is a change in the lighting, rapid movement of an object, or an object the color of skin, it is difficult to recognize the user's gesture.

It has been possible to solve these problems with the release of Kinect by Microsoft. Kinect uses an RGB sensor and IR sensor to provide RGB and depth images. The depth image is generated using a pattern matching method that reads an illumination pattern with an infrared camera. In addition, the depth information is provided by a depth image and used to calculate the distance between the object and Kinect. When the distance between Kinect and the object becomes smaller, the depth information has a smaller value. On the other hand, as the distance between the object and Kinect increases, the depth information has a larger value. The greatest distance between Kinect and the object is 1.2~3.5 m. This Kinect technique makes it possible to separate the background and people and is not affected by the illumination, unlike a conventional video camera. Even if there is a person with a different skin color near the object, Kinect accurately separates the background.

In this paper, we propose a hand recognition method using Kinect and the geometric characteristics of the hand, which separates the hand and background using the depth information and recognizes the fingers and number of fingers using the distance between the hand's center point and hand region using the contour. The contour is detected using the Suzuki85 algorithm [6].

As an example application of the proposed method, we explain a Secret Door that recognizes a password through recognizing the number of fingers held up by a user without special devices such as door locks.

2. Related Work

Gestures can be used for the simplest and most intuitive interface and can effectively control a variety of displays [7][8]. Therefore, studies have been carried out on various methods for gesture recognition. Among these studies, a gesture recognition method using an image input device such as a camera and Kinect and a data glove has been highlighted. A method that uses an image input device was presented using the depth values, skin color, or skin color and depth value at the same time.

Park proposed a method that recognized gestures using a data glove [9]. It used a data glove from 5DT. It categorized the input data from the data glove and developed a gesture recognition model. First, after inputting data, it identified the difference between the current value and the average value of the previous three data points. If the value obtained by subtracting the average value of the previous three data points from the current value was larger than 10, the operation to process a movement change was conducted. If the value obtained by subtracting the average value of the previous three data points from the current value was less than 10, the current operation was determined to be meaningless. This same method was used for all five fingers. Thus, a gesture was recognized by comparing the sequences of operations.

Camatra proposed a gesture recognition method based on a data glove and LVQ (Learning Vector Quantizer) [10]. It consisted of two modules. The first module extracted features from the data input by the data glove. The second module performed the classification role using the LVQ. The data glove (DG5 VHand 2.0) updated the flexure information of each finger every 20 ms, and updated it every 100 ms because it calculated the average of five consecutive flexure information retrievals for each finger for accuracy. The LVQ traversed all the input vectors and looked for the best weight vector closest distance. This method recognized a gesture using the vector input from the data glove.

Shitala captured a gesture with a data glove and mapped it to a keyboard to control a media player [11]. The gesture was recognized using the data glove and a decision tree. The data glove used was DG5 VHand 2.0. The data glove transmitted the data using Bluetooth, and the transmitted data were compared with pre-defined gestures using the decision tree. These pre-defined gestures included seven components, including play, forward rewind, back rewind, pause, next, and previous.

Choi and Han separated the background and hand region and recognized the number of fingers using skin color [12]. For the hand region separation, they extracted a part that was similar to the skin color from the color image input through a camera. Using the skin color made it easy to separate the background and hand by extracting a pixel with a color similar to the pixel value of the person. In this method, the detected area was the hand region. However, the detection area could include more than the hand region. It could include the arm region as well as the hand region. In this case, exact hand recognition was impossible. Therefore, the geometric characteristics of the hand were used to separate the arm region and hand region to detect the hand region. This was done by drawing a circle in each region and calculating the intersection angle between the respective region and a circle that separated the hand region and arm region. If more than two intersection angles were greater than 25°, it was recognized as the arm region. **Fig. 1-a** shows the process for recognizing the hand and arm regions using this method, and **Fig. 1-b** shows the recognition results for the hand and arm regions.

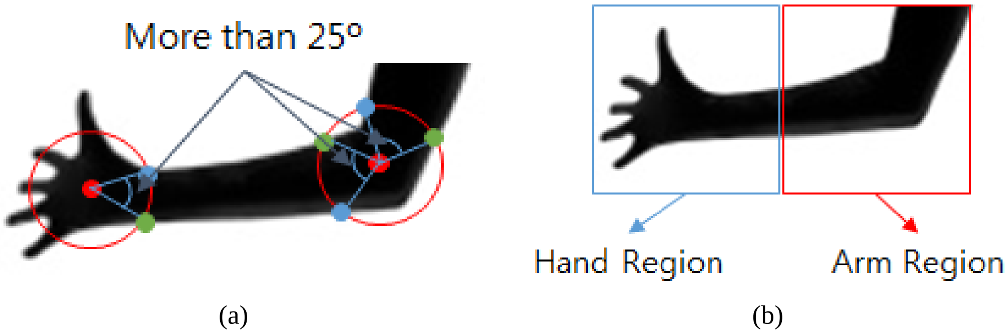


Fig. 1. Recognition results for hand and arm regions

After they found the hand region without the arm region, they found its center point. The center of the hand region had the largest pixel value using the distance converted from the hand region. After recognizing the center of the hand, a circle was drawn in this region to identify the number of fingers. The radius of this circle was 1.5 times the distance from the center of the hand to the hand region. After drawing a circle, the overlapping portion of the circle and hand region was explored in a clockwise direction. The angle of the overlapping portion of the two uninterrupted points was calculated, where a value below 10º indicated a finger and 25º indicated the wrist. After the search ended, the number of fingers was recognized by counting.

Choi and Seo recognized fingers using the depth information and skeleton tracking of Kinect [13]. Kinect provides depth information and a skeleton tracking function using the depth information. This skeleton tracking function is the ability to track the location of a human bone in the image. By utilizing this, Choi and Seo were able to track the hand skeleton and define the hand region as a part of the hand skeleton area. Moreover, they found the center point of the hand using distance conversion. Then, the curved portion was located and used to find candidates in the hand area. If the outer product value of a candidate was a positive number, it was excluded as a candidate. In addition, they recognized the number of fingers in the rest of the finger candidates.

Jagdish identified the number of fingers using the depth information and openNI module [14]. First, he separated the background and hands using the depth information provided by a depth image of Kinect. When a user gestures, the distance between Kinect and the hand is the smallest. That is, the depth information for the hand is the smallest. Therefore, setting the threshold to the depth information could eliminate the background. He then found the hand point with the NITE module using Bayesian Object Localization. A region with a hand point was recognized as the palm region. He found the finger region by removing the palm region from the hand region. In the finger area, the finger number could be recognized by exploring the depth values.

Hongyong identified the number of fingers using the depth information and skin color at the same time [15]. In the case of using only the skin color, a similar part can be detected in addition to the skin area to exclude the hand region, and in the case of using only the depth information, accurate detection is difficult because of noise. However, when the two methods were used at the same time, the hand region could be detected accurately. First, the hand region was detected. This was separated from the background to set the depth information threshold. Then, a region similar to the skin color was detected in an image using YCrCb. Next, he found common parts in the obtained regions that were found using the two methods. The common part was the hand region. The number of fingers was recognized by finding the smallest depth information in the hand region.

Park recognized the hand shape using the depth information, a color image, and the geometric characteristics of the hand. [16]. First, he found the common part of the hand region using the skin color and depth information of the hand region in an image. Then, he recognized the center of the hand based on the largest pixel value after applying the distance conversion. The center point of the hand was a circular area, and the palm region was detected by drawing a circle with a radius equal to the distance conversion value. In addition, the hand outline was detected to find the fingertips, with sudden gradient changes used to specify the fingertip candidates. At this time, in order to remove the portion other than the fingertip in a candidate, the part that had a positive value for the third element of the outer product was removed. Then, the number of fingers was recognized using the remaining candidates. In addition, Choi recognized the hand region based on the geometric characteristics of the hand [17], and Cao determined the pose or shape of a hand based on a database of feature vectors [18].

3. Algorithm for recognizing number of fingers

The hand region detection algorithm is shown in **Fig. 2** in the form of a flowchart used to recognize the number of fingers. Kinect simultaneously provides color and depth images. The depth image is input from an infrared sensor and offers depth information. The depth information shows the distance between Kinect and the object. It varies depending on the distance, that is, it has a smaller value for a smaller distance between Kinect and the object and a larger value for a larger distance between Kinect and object. The depth information is appropriate for recognizing a gesture that is input in three-dimensional form. The hand region is preferentially separated from the background by setting the depth information threshold, in order to correctly recognize a gesture. The depth information of the hand has a smaller value because the distance between Kinect and the object is smaller when a user makes a gesture. After the separation of the hand region and background, the image is converted into a gray image and binary image to recognize the number of fingers. Thus, the outline of the hand region is detected in the image.

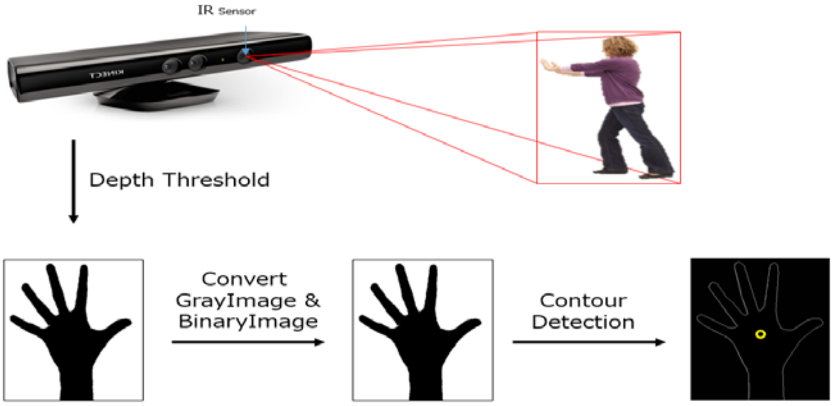


Fig. 2. Flowchart for hand region detection algorithm

3.1 Hand Region Detection

To correctly recognize the hand region, pre-processing is used to separate the background and hand region. In this paper, we utilize the depth image from Kinect to remove the background and hand. The depth information value of the depth image is smaller for a distance between Kinect and the object. That is, there is a larger difference between the depth information for the hand and background. Therefore, we separate the hand region in an image by setting a threshold for the depth information. The hand region is converted into a gray image and binary image to recognize the number of fingers. **Fig. 3** shows the results for detecting the hand region using a threshold for the image.

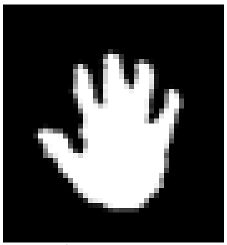


Fig. 3. Detection Of Hand Region By Threshold

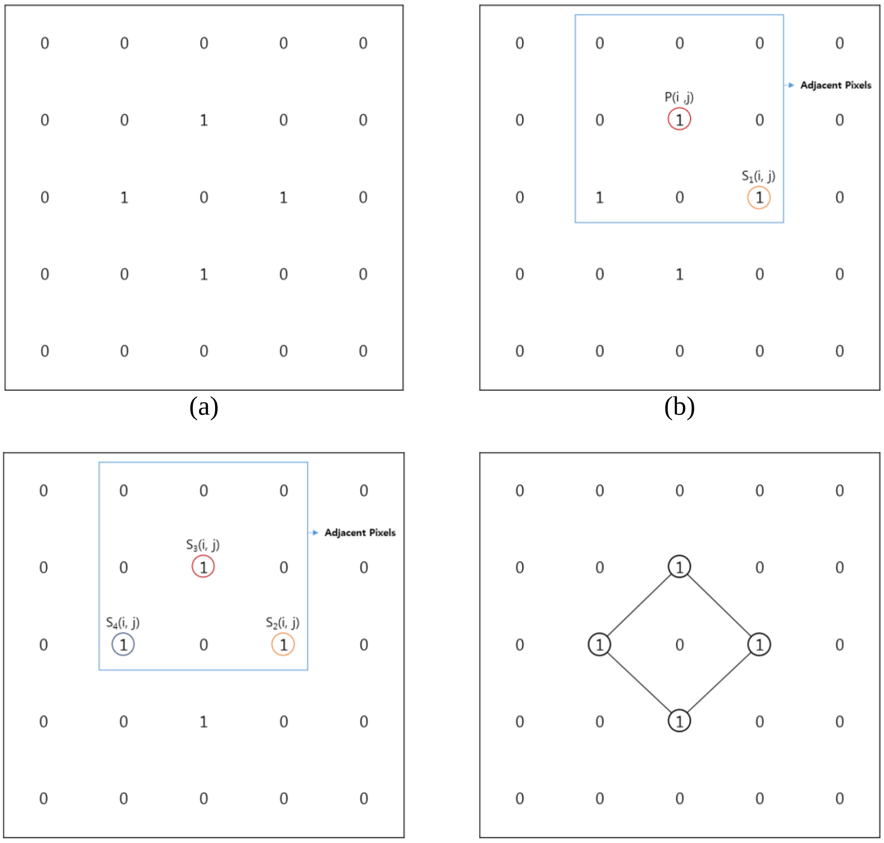
3.2 Hand Outline Detection

The center point and outline of the hand are captured based on the contour, which is detected using the Suzuki85 algorithm [6]. The contour is a set of pixels with the same value. However, a frame of the input image for Kinect is composed of pixels with different values. It takes a long time to detect a contour in this form. Therefore, the pixel values are simplified using a binary image. A binary frame is configured in row and columns. A row is expressed by "i" and column by "j." The pixel located at row "i" and column "j" is expressed by P(i,j). The row numbers increase from top to bottom, and the column numbers increase from left to right. In addition, we store four pixel locations to confirm the detected contours. The saved pixel location is displayed as Sn(i,j), ($1 \leq n \leq 4$).

Fig. 4-a shows an example of a binary image frame, where 1 is a pixel with a value greater than the threshold, and 0 is a pixel with a value smaller than the threshold. To detect the contour, one line is scanned. If P(i,j) is $P_{ij} \neq 0$, we perform the following step. **Fig. 4-d** shows the contour detection result.

1. If $P_{ij} = 1$ and $P_{i,j-1} = 0$, this is the starting point of the contour, and $S_2(i,j) \leftarrow P(i, j\text{-}1)$. Otherwise, go to 6.
2. From the center in P(i,j), the adjacent pixels are searched in the clockwise direction. The search starts at $S_2(i,j)$. If point $P_{ij}=1$ is found, it is stored as $S_1(i,j)$(**Fig. 1-b**). Otherwise, go to 6.
3. $S_2(i,j) \leftarrow S_1(i,j)$ and $S_3(i,j) \leftarrow P(i,j)$.
4. From the center in $S_3(i,j)$, the adjacent pixels are searched in the counterclockwise direction. The search starts at $S_2(i,j)$. If point $P_{ij}=1$ is found, it is stored at $S_4(i,j)$(**Fig. 4-c**).

5. If $S_4(i,j) = P(i,j)$ and $S_3(i,j) = S_1(i,j)$, go to 6. Otherwise, $S_2(i,j) \leftarrow S_3(i,j)$ and $S_3(i,j) \leftarrow S_4(i,j)$. And go to 3
6. The process is restarted from P(i,j+1). It is terminated if P(i,j) is the lower right corner of the frame.



(a) (b)

(c) (d)

Fig. 4. Contour Example

We find the center of the hand by calculating the average value of the detected contour coordinates, and the outline of the hand is found by connecting the contour coordinates. **Fig. 5** shows the detection results for the center point and outline of the hand region.

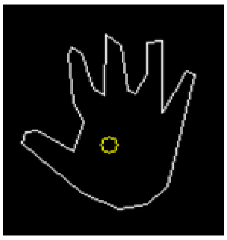
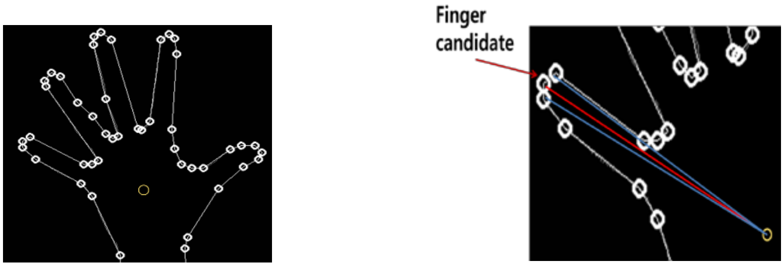


Fig. 5. Detection Of Center Point and Outline Of Hand

3.3 Finger Count Recognition

The number of fingers is recognized using the distance between the center point of the hand and its outline. We use the coordinates of each contour because the outline consists of the contours. Therefore, we can calculate the distance between the coordinates for the outline and the center point of the hand. The navigation direction progresses counterclockwise, and we compare the distances for three continuous coordinates and the center point to find the number of fingers. If the value of the second is the largest, in a series of three values, it is a finger candidate. If a contour coordinate is below the center point of the hand, it is excluded from the calculation. After scanning, the number of fingers is recognized from the finger candidates. Fig. 6-a shows the contour detection result, and Fig. 6-b shows the process of finding the finger candidates using the proposed method. The white points show the contour of the hand area and the red and blue lines are connected to the center point (which is marked in yellow) of the hand. The red line has the longest length between the contour and center point of the hand among the three consecutive contours. The contour connected to the center point marked in red is a finger candidate. **Fig. 7** shows the result of recognizing the number of fingers using the method proposed in this paper.



(a) (b)

Fig. 6. Contour Detection and Finger Navigation Process

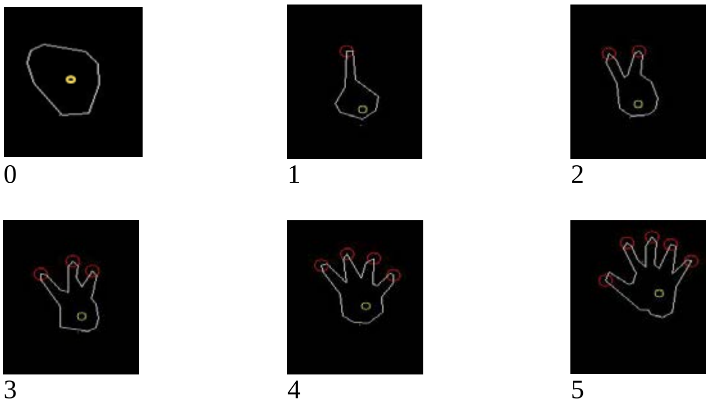


Fig. 7. Recognition of Finger Count Using Distance

4. Performance Evaluation of Algorithm for Finger Count Recognition

The recognition rate is evaluated according to the number of fingers using the proposed algorithm for finger count recognition. The experiments were conducted by five people. We compare the recognition rates for different numbers of fingers in **Fig. 7**. The experiment was conducted in a typical environment, in which a personal computer with an AMD FM(tm)-8150 Eight-Core Processor (3.60 GHz), 32 GB of memory, and Windows 7 Professional K Service Pack 1(64 bit) was used, along with Kinect for Windows. A depth image had a 640 × 480 resolution and was output at 30 frames per second. **Table 1** lists the recognition results for the number of fingers used for the proposed method.

Table 1. Recognition rate according to number of fingers

Recognition according to number of fingers						The average recognition rate
0	1	2	3	4	5	
100%	100%	98%	96%	98%	100%	98.6%

Table 2 compares the results of the methods proposed in the previous studies and this paper. The recognition rate for the number of fingers using a data glove had an average of 95%, whereas when using the depth information and skin color at the same time, the average was 98%.

Table 2. Recognition rate comparison of results of methods in previous studies and this paper

Method / Number	Data Glove [8]	Depth & Skin Color [12]	Proposed Method
1	97%	100%	100%
2	95%	100%	98%
3	92%	93%	96%
4	91%	100%	98%
5	100%	97%	100%
Average	95%	98%	98.4%

Table 3 shows a comparison of the processing times for the methods proposed in previous studies and this paper. The execution time of the method using the depth information and skin color is 25 ms, and that for the method proposed in this paper is 0.065 ms.

Table 3. Algorithm execution time comparison of conventional methods and proposed method

Method / Execution time	Data Glove [8]	Depth & Skin Color [12]	Proposed Method
Execution time	-	25 ms	0.065 ms

The accuracy of the method proposed in this paper had an average of 98.6%, and the execution times was 0.065 ms. The accuracy was high compared to the other methods, because we used the depth information, which was not affected by the surrounding environment, along with the contour and geometric characteristics of the hand. Moreover, because we used the depth information, no refining process or additional operations were needed for the input data. In addition, the operation for recognizing the finger number was simple and faster than conventional methods because it used the geometric characteristics of the hand.

5. Conclusion

With the exception of the voice, gestures are the most intuitive human means of expression. Opinions can be conveyed by gestures as non-verbal elements such as a wink or handshake. In addition, because gestures may be found in a variety of forms, they have been highlighted as a means of interaction between a human and computer. In this paper, we proposed a method for recognizing the finger count in an image through an image input device to recognize a gesture. The hand region is detected using the depth information in a depth image from Kinect, and the finger count is recognized by comparing the distances between the coordinates of the contour that constitutes the outline of the hand and its center point. The average recognition rate for counting fingers with the proposed method was 98.6%, and the execution time was 0.065 ms. In other words, the recognition speed of the proposed method was fast, and its complexity was low.

The proposed method can be used in any application that requires the recognition of the number of fingers such as a secret door. For example, if only a camera is used, without any other equipment such as a door lock, a secret door can remain secure by implementing the method proposed in this paper. The secret door system is shown in Fig. 8. First, we receive the Kinect image. Then, if a password has not been set, you set the password, which is recognized using the algorithm proposed in this paper with the input image. If the recognized password is the same as the set password, the door is opened.

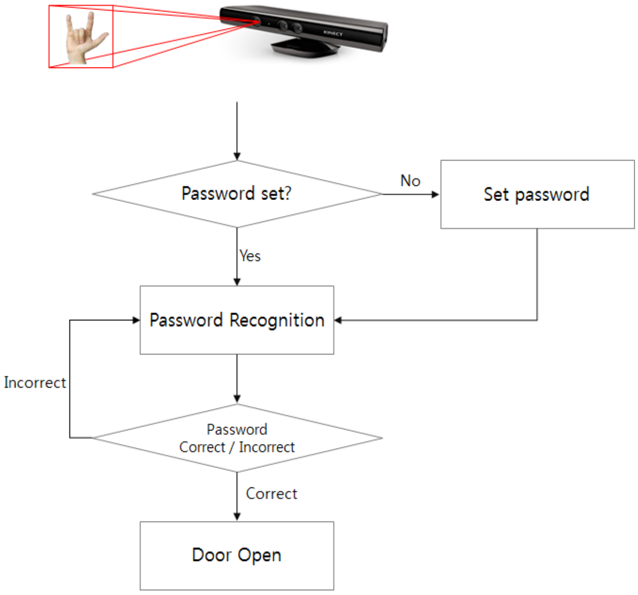


Fig. 8. Flowchart for Secret Door System

The password used for the secret door system is the number of fingers. Numerous gestures that depend on the number of fingers can be expressed. In other words, the password used by the secret door system is a gesture that can be represented by a hand. Therefore, if the system recognizes the pose according to the number of fingers, the gestures that can be used as a password will be increased in a future study. Accordingly, the security strength of the secret door will be maintained at a high level.

References

[1] J. L. Applegate and G. B. Leichty, "Managing interpersonal relationships : Social cognitive and strategic determinants of competence.," *Competence in communication : A multidisciplinary approach*, pp. 33-56, 1984.

[2] S. J. Chae, "The Importance of Nonverbal Communication Skills," *Korean journal of medical education*, vol. 22, no. 2, pp. 149-150, 2010. Article (CrossRef Link).

[3] J. P. Wachs, M. Kolsch, H. Stern and Y. Edan, "Vision-based hand gesture application," *Communications of the ACM*, vol. 54, no. 2, pp.60-71, 2011. Article (CrossRef Link).

[4] S. Y. Park and E. J. Lee, "Hand Gesture Recognition Algorithm Robust to Complex Image," *Journal of Korea Multimedia Society*, pp.1000-1015, 2010. Article (CrossRef Link).

[5] L. Connelly, Y. Jia, M. L. Toro, M. E. Stoykov, R. V. Kenyon and D. G. Kamper, "A pneumatic glove and immersive virtual reality environment for hand rehabilitative training after stroke," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 5, pp.551-559, 2010. Article (CrossRef Link).

[6] Suzuki, Satoshi, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics and Image Processing 30.1*, pp.32-46, 1985. Article (CrossRef Link).

[7] M. Y. Chen, L. Mummert, P. Pillai, A. Hauptmann and R. Sukthankar, "Controlling your TV with gesture," in *Proc. of the international conference on Multimedia information retrieval, ACM*, pp.405-408, 2010. Article (CrossRef Link).

[8] H. P. Jain, A. Subramanian, S. Das and A. Mittal, "Real-time upper-body human pose estimation using a depth camera," *Computer Vision/Computer Graphics Collaboration Techniques*, Spring Berlin Heidelberg, pp.227-238, 2011. Article (CrossRef Link).

[9] I. K. Park, J. H. Kim and K. S. Hong, "An implementation of an FPGA-based embedded gesture recognizer using a data glove," in *Proc. of the 2nd international conference on Ubiquitous information management and communication*, pp.496-500, 2008. Article (CrossRef Link).

[10] Camastra, Francesco, and Domenico De Felice, "LVQ-based hand gesture recognition using a data glove," *Neural Nets and Surroundings*, Springer Berlin Heidelberg, pp. 159-168, 2013. Article (CrossRef Link).

[11] Prasad, Santasriya, Pranaw Kumar, and Kumari Priyanka Sinha, "A wireless dynamic gesture user interface for HCI using hand data glove," *Contemporary Computing (IC3), 2014 Seventh International Conference on. IEEE*, pp. 62-67, 2014. Article (CrossRef Link).

[12] J. Choi, S. Han, H. Park and J. I. Park, "A Study on Providing Natural Two-handed Interaction Using a Hybrid Camera," *In The Third International Conference on Digital Information Processing and Communications (ICDIPC2013), The Society of Digital Information and Wireless Communication*, pp.481-484, 2013. Article (CrossRef Link).

[13] J. Choi, B. K. Seo, D. Lee, H. Park and J. I. Park, "RGB-D Camera-based Hand Shape Recognition for Human-robot Interaction," *In Robotics (ISR), 2013 44th international Symposium on, IEEE*, pp.1-2, 2013. Article (CrossRef Link).

[14] J. L. Raheja, A. Chaudhary and K. Singal, "Tracking of fingertips and centers of palm using kinect," in *Proc. of Computational intelligence, modelling and simulation (CIMSiM), 2011 third international conference on*, pp.248-252, 2011. Article (CrossRef Link).

[15] T. Hongyong and Y. Youling, "Finger tracking and gesture recognition with kinect," in *Proc. of* Computer and Information Technology (CIT), IEEE 12th International Conference on, pp.214-218, 2012. Article (CrossRef Link).

[16] H. Park, J. Choi, J. I. Park and K. S. Moon, "A Study on hand region detection for Kinect-based hand shape recognition," *Journal of Broadcast Engineering*, vol. 18, no. 3, pp.393-400, 2013. Article (CrossRef Link).

[17] J. Choi, H. Park and J. I. Park, "Hand shape recognition using distance transform and shape decomposition," in *Proc. of Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp.3666-3669, 2011. Article (CrossRef Link).

[18] C. Cao, Y. Sun, R. Li and L. Chen, “Hand posture recognition via joint feature sparse representation,” *Optical Engineering*, vol. 50, no.12, pp.127210-127210, 2011. Article (CrossRef Link).

Doyeob Lee received a B.S. in Computer Science from Academic Credit Bank System, Korea, in 2014. He is currently a Master Student in the Department of Computer Engineering at Sejong University in Korea. His research interests include Natural User Interface, Human Computer Interaction and Data Mining.

Dongil Shin received a B.S. in Computer Science from Yonsei University, Seoul, Korea, in 1988. He received an M.S. in Computer Science from Washington State University, Pullman, Washington, U.S.A., in 1993, and a Ph.D. from University of North Texas, Denton Texas, U.S.A., in 1997. He was a senior researcher at System Engineering Research Institute, Deajun, Korea, in 1997. Since 1998, he has been with the Department of Computer Engineering at Sejong University in Korea where he is currently a Professor. His research interests include Information Security, Bio-Signal Data Processing, Data Mining and Machine Learning

Dongkyoo Shin received a B.S. in Computer Science from Seoul National University, Korea, in 1986, an M.S. in Computer Science from Illinois Institute of Technology, Chicago, Illinois, in 1992, and a Ph.D. in Computer Science from Texas A&M University, College Station, Texas, in 1997. He is currently a Professor in the Department of Computer Engineering at Sejong University in Korea. From 1986 to 1991, he worked in Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he worked in the Multimedia Research Institute of Hyundai Electronics Co., Korea as a Principal Researcher. His research interests include Information Security, NUI/NUX, Ubiquitous Computing.