Depth Camera-based Screen Control System Employing Fingertip

YANG-KEUN AHN, KWANG-SOON CHOI, YOUNG-CHOONG PARK
Korea Electronics Technology Institute
121-835, 8th Floor, #1599, Sangam-Dong, Mapo-Gu, Seoul
REPUBLIC OF KOREA
ykahn@keti.re.kr

Abstract: - This paper proposes a system that controls an image shown on a screen through hand information extracted from a depth camera. The proposed method extracts the hand region from the distance information obtained from the depth camera, and employs outline extraction and convex hull to locate the fingertips. An image output on a screen was controlled by the velocity of the fingertip information using current and previous frames, and an application program was written to verify the level of control.

Key-Words: - Finger Tracking, Fingertip Detection, Air Touch, Screen Control System, Finger Touch

1 Introduction

Diverse types of display screens are now installed with various electronic devices due to their continuing development, and diverse control methods have also been introduced to control them. A touch screen installed on a smart-phone is a representative control device. Such touch-screens are widely employed. However, a touch-screen device has a disadvantage in that the screen has to be directly touched to exercise control.

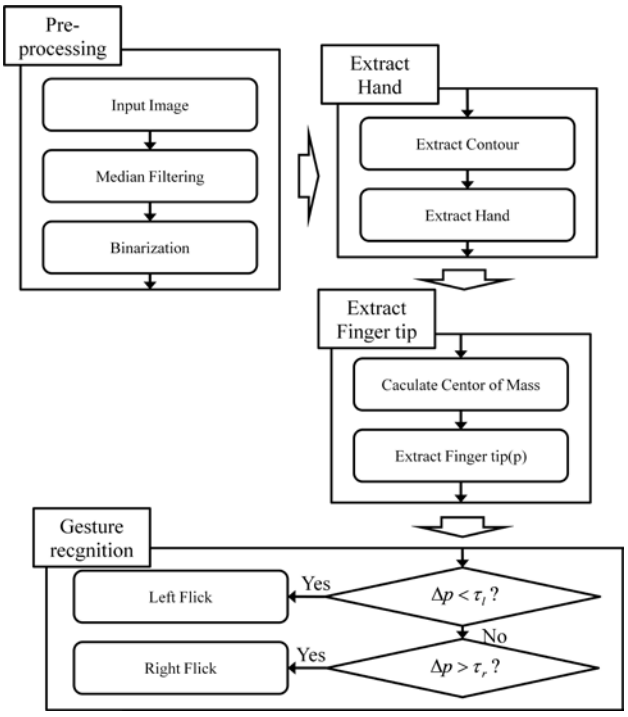

Fig. 1 Flow chart of the proposed system

To address this disadvantage various methods employing fingertips for screen control have been proposed, including a method that utilized a high-speed camera attached to a mobile device [1], which required a high-performance camera and a series of processes in the initialization stage. A method employing z-touch [2] had limited processing speed and the method failed to accurately estimate continuous depth values. A method employing a multiple number of layers [3] required the use of two panels in the form of layers and it drastically increased the equipment cost.

In order to resolve such issues, this paper proposes a screen control method employing a depth camera. The overall flow is illustrated in Fig. 1. A depth camera was installed under a screen for the control of the screen, and a convex hull and the center of gravity of the fingertips were employed in extracting and selecting the appropriate control fingertip from among several fingertips.

The velocities of the previous and present frames of the extracted fingertip points were utilized to determine whether the image on the screen was being flicked to the left or to the right, and in certain cases, the image was not changed according to the change in the velocities.

2 Preprocessing

The distance which a depth camera recognizes comprises a space for the depth camera to use. However, since the camera could malfunction in undesired ways, or the system is poorly configured, the desired control may not be achieved. Therefore, as illustrated in Fig. 2, in the present work only a region 20cm–80cm away from the camera was set

as the usable region, and the camera was installed to face 30°above the ground to recognize the upper body of a person with their hands.

The data inputted from the camera is the distance data from the objects and background to the camera. The inputted distance data becomes the basis of every image used in the system. Therefore, noise reduction is conducted on the image to obtain images with better qualities from the progressing process. Smoothing computation was employed in the noise reduction. Of the diverse smoothing computations, the proposed system employs a median filter. The processed data out of the usable region was reassigned as 0 and was not used in the system.

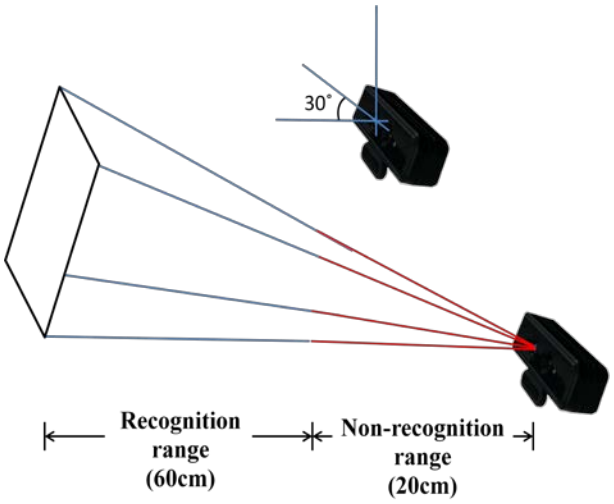

Fig. 2 Camera installation angle and hand recognition range

A black and white image was primarily implemented with a proportional expression for the distance data, having values between 0 - 255. The implemented black and white image is a depth image. The pixels in the depth image expressed in light colors represent distances close to the camera, and the pixels expressed in dark colors represent distances farther away from the camera.

3 Hand region extraction

If one stretches one's hand to make a gesture, then the hand would be located closer to the camera. In the depth image, the location closest to the camera is the part with the brightest color. The distance value of the brightest pixel inputted by the camera is extracted and only the pixels within a certain distance range of it are then employed. Pixels out of the distance range are set at 0 to remove the information. Converting the data remaining after the process into a binary image result in a binary candidate image including the hand region. Ultimately, data are obtained from the portion closest to the camera, and a binary image is obtained from the data.

Noise and parts other than hands are also typically included in the candidate image for the hand extraction. Therefore, these other parts should be appropriately removed. Outline extraction [4] is required to remove such parts. Information on each extraction region is obtained through the outline extraction, and the region with the largest size is selected as the hand. In such case, regions other than hand region are not selected, and, thus, removed.

Fig. 3 Hand region extraction

4 Fingertip coordinate extraction

An outline of the hand is obtained from the extraction of the controlling fingertip coordinate, and a convex hull [5] is obtained from the outline information.

Connecting the points constituting the convex hull identifies the parts of the fingertip connected to the wrist, and it ultimately extracts the outermost points.

The outermost points obtained from the convex hull include not only the fingertips, but also the other parts.

The center of gravity is computed to select the reference coordinate among the fingertip points, and to measure the distance between the reference point and the outermost points, obtained from the center of gravity and convex hull, by the Euclidean method.

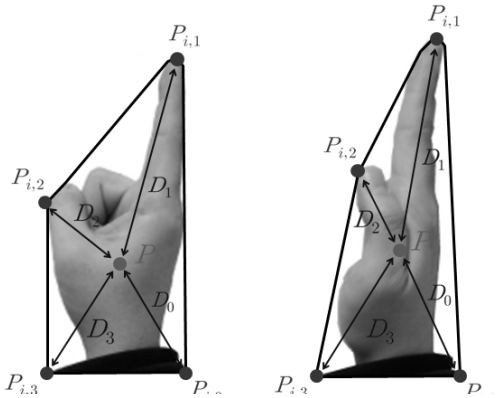

Fig. 4 Distance between the center of gravity and outermost points of the hand

$$D_k = \sqrt{(P_{i,k,x} - P_{c,x})^2 + (P_{i,k,y} - P_{c,y})^2} \quad (1)$$

The fingertip point, P_r, is the largest of the measured distances.

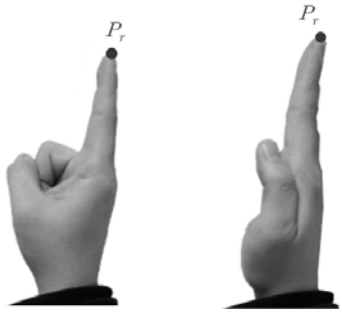

Fig. 5 Fingertip point extraction

$$P_r = \max_{P_{i,k}} D_k \quad (2)$$

5 Gesture recognition

A gesture is controlled by either a finger or all of the fingers. The difference between the references coordinates of the current and previous image frames are employed for gesture classification, and this signifies the velocity of the fingertip.

Gestures are largely classified into 2 types. One type of gesture swiftly moves the hand to the left, to flick to the left, and another type of gesture swiftly moves the hand to the right to flick to the right.

The gesture that flicks to the left is the state in which the fingertip quickly moves to the left. Either a finger or every finger is used in the gesture and the velocity with respect to the x axis should be a negative value when the gesture intends to flick to the left. The gesture will be in effect when the velocity is less than a certain velocity τ_l.

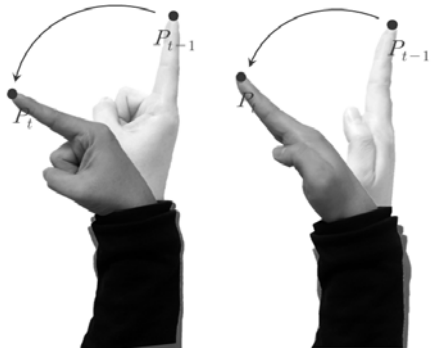

Fig. 6 The shape of the gesture used to flick to the left

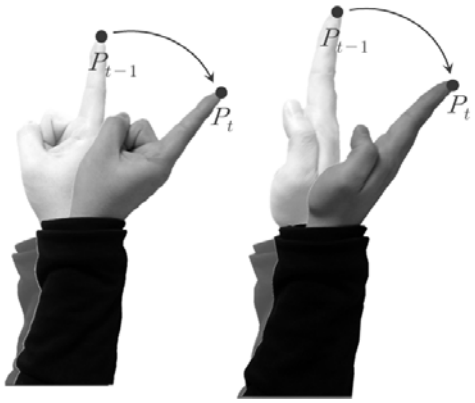

Fig. 7 The shape of the gesture used to flick to the right

The gesture that flicks to the right is the state in which the hand quickly moves toward the right side, and as was the case in the gesture to the left, either a finger or every finger is used. Unlike the gesture of the left, the velocity in the gesture to flick to the right is a positive value. Therefore, the gesture will be in effect only when the velocity is larger than a certain velocity τ_r, and it is also measured only with respect to the x axis.

However, such computation creates a problem. In gesturing to the left, the hand moves to the left and comes back to the original position. The velocity during the return of then hand is a positive value, and it is recognized as the gesture to flick to the right side. If the velocity is less than τ_l during the gesturing to the left and the velocity is also less than τ_r during the return to the original position, no problem would arise because only the gesture of the left is recognized. However, this means that the return gesture has to be intentionally slower than the gesture of the left, or, the returning action should not be made. In either of these cases the gesturing becomes uncomfortable and the gestures will not be

meaningful. Clearly, in the case of the gesture to flick to the right, returning to the original position could also be recognized as a gesture to flick to the left.

Another case is a shaking motion in which the hand moves left and right in a velocity that does exceed τ_l and τ_r. When the velocity does not exceed a certain velocity, shaking the hand left and right will not be recognized as a gesture, and, thus, it is not considered to be shaking. A method is required to prevent this. Therefore, the proposed method additionally considers certain velocities for left and right gestures. The certain velocities τ_l' and τ_r' are more sensitive than the other certain velocities, τ_l and τ_r, used for gesture recognition, so that it can recognize the shaking motion.

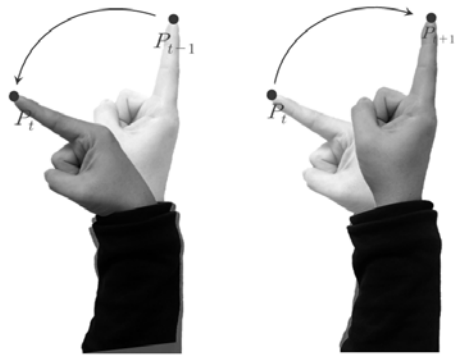

Fig. 8 Returning to the original position after gesturing to the left

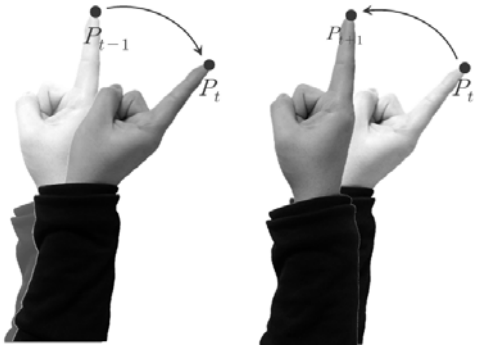

Fig. 9 Returning to the original position after gesturing to the right

A gesture is set to be not recognized when a signal is generated twice in a row in different directions within a certain period time without reaching τ_l and τ_r with respect to τ_l' and τ_r'.

The gestures of flicking to the left and right are only executed when the TRUE condition is satisfied, and the TRUE state is satisfied when P_t and P_{t-1} are usable and the gestures were made within the usable region.

$$Left\ Flick = \begin{cases} TRUE & (\text{if } (P_{t,x} - P_{t-1,x}) < \tau_l \text{ and } usable\ P_t, P_{t-1}) \\ FALSE & (else) \end{cases} \quad (3)$$

$$Right\ Flick = \begin{cases} TRUE & (\text{if } P_{t,x} - P_{t-1,x} > \tau_r \text{ and } usable\ P_t, P_{t-1}) \\ FALSE & (else) \end{cases} \quad (4)$$

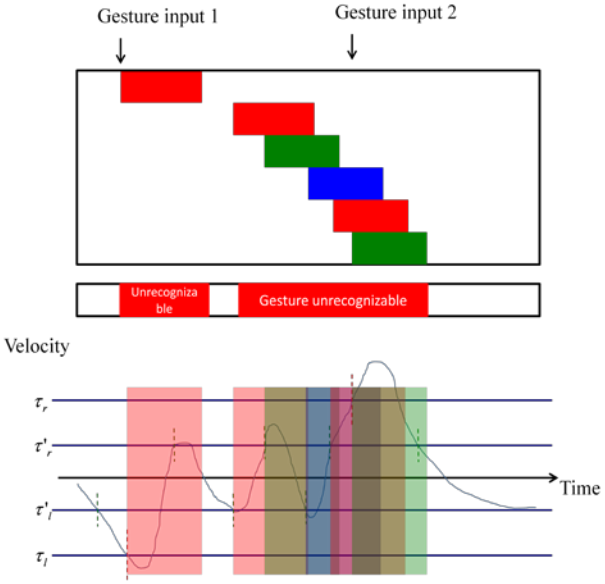

Fig. 10 Gesture input limit application

6 Range of application of the hand on the screen

When a hand goes out of the field of view through the top, the arm is captured or the gesture is not recognized. In order to prevent such occurrence, the range of application of the hand was slightly limited.

A rectangle that surrounds the hand is established to limit the use of the hand to only when the conditions of the range of application are satisfied. When the conditions are not satisfied, the information of the hand is deleted from the image to prevent undesired computations.

Fig. 11 A rectangle based on the outline of a hand

Computations of conditions within the range of application were conducted with two points of the outer rectangle because only the outermost points are required in the computations, and the other two points of the rectangle could also be employed in the computations for the conditions.

Conditions within the range of application:

$$\begin{aligned} \alpha_x &< x_1 \\ x_2 &< width - \beta_x \\ \alpha_y &< y_1 \\ y_2 &< heights - \beta_y \end{aligned} \quad (5)$$

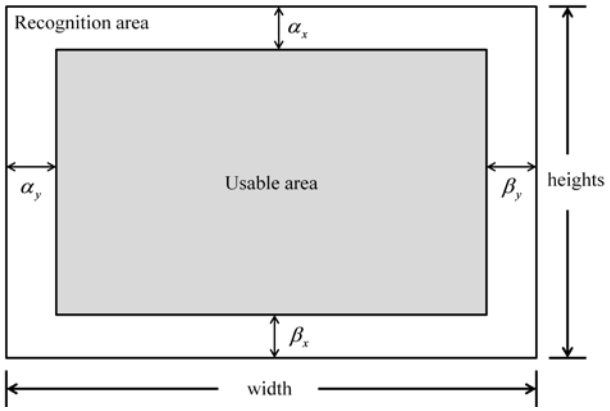

Fig. 12 Range of application of the hand

7 System composition

A representation program that represents the locations of a 3D object and a hand was implemented to verify the operation of the program proposed by this paper. A gesture based on a user's hand is inputted into a PC through a depth camera. Then the PC analyzes the input and shows the resulting motion or scaling of the object represented on the screen.

7.1 Hardware composition

Fig. 18 illustrates the hardware composition. The hardware is composed of a depth camera, a transparent monitor, and a PC. The depth camera records at hand and transmits the image to the PC through a USB interface, and the PC analyzes and processes the image recorded by the depth camera to analyze the gesture of a hand. The content processed through the gesture analysis is outputted through the screen connected by HDMI interface.

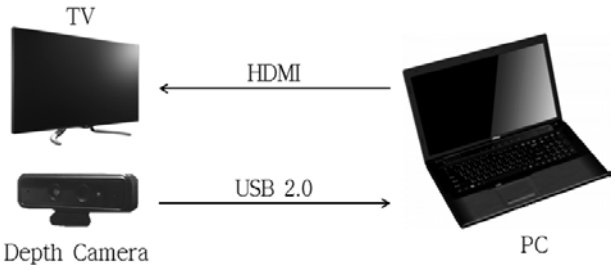

Fig. 13 Hardware connection status

7.2 Software composition

The software proposed by this paper is largely classified into two parts. The process part processes a gesture, and the application part shows the change of the image according to the given gesture.

Fig 14 illustrates the analysis from the process part. (a) shows the process regarding the fingertip and hand, (b) shows the degree of the motion of the fingertip, (c) shows the depth image, and (d) shows the depth image converted to a binary image.

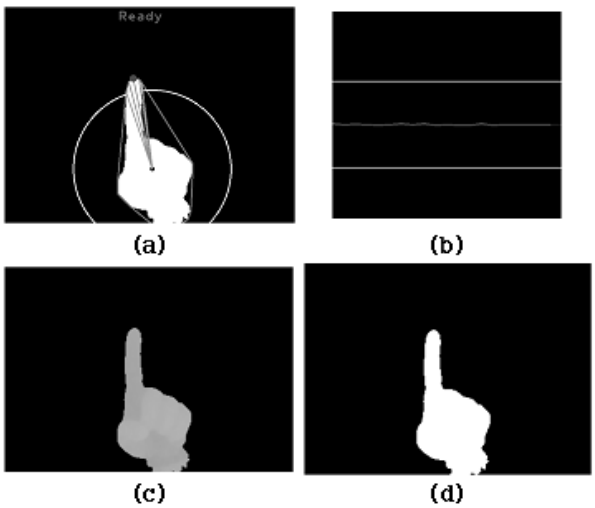

Fig. 14 Partial images of the process

8 Experiment result and conclusion

The experimental condition was set up with a depth camera DS325 connected to a PC with Intel(R) Core(TM) i7-2600K 3.4GHz CPU and 8Gbyte memory.

The gesture that quickly moves the fingertip to the left for a left flick and the gesture that quickly moves the fingertip to the right for a right flick were tested to verify the content of this paper. A total of 10 users were asked to make 50 left flicks and 50 right flicks in random order for a total of 1000 flick motions.

Fig. 15 Experiment

The experimental result is listed in Table 1 and indicates that the left flick motion yielded a higher recognition rate than the right flick motion. Such result correlates to the fact that the hand gesture moving to the left has a higher degree of freedom than the gesture moving to the right. Importantly, there was no incidence where a left flick was recognized as a right flick and vice versa, but there were cases where users' flick motions were not recognized. Such results are due to the fact that the users' motions failed to reach the threshold value, and the results did not influence an actual screen control.

Table 1 Recognition rate of flick events

User	Left(50)	Right(50)	Total(%)
1	50	50	100
2	50	48	98
3	50	50	100
4	49	49	98
5	50	48	98
6	50	50	100
7	49	50	99
8	50	50	100
9	50	49	99
10	50	49	99
Total	498	493	99.1

References:

[1] [1] Y. Hirobe, T.Niikura, Y. Watanabe, T. Komuro, M. Ishikawa, “Vision-based Input Interface for Mobile Devices with High-speed Fingertip Tracking,“Adj. Proc. ACM UIST 2009, pp. 7-8.

[2] [2] Y. Takeoka et al.: Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces, Proceedings of ITS’10, 2010.

[3] [3] Y. Tsukada et al.: Layerd touch panel: the input device with touch layers, Proceedings of CHI’02, 2002, pp. 584-585.

[4] [4] Suzuki. S. and Abe. K., "Topological Structural Analysis of Digitized Binary Images by Border Following", CVGIP, 1985, pp32-46

[5] [5] Sklansky. J., "Finding the convex hull of a simple polygon", PRL 1 $number, 1982, pp. 79-83