Spatial Gesture Interaction System

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 study proposes a method for controlling information output on a table-type display through hand movements in the air using a single depth camera without any actual touching of devices. First, the hardware configuration of the AirTouchTable system, which enables the control of a table-type display based on spatial interactions, was described. To control of the system, a user detection zone, gesture recognition zone, and object recognition zone are determined. When a user enters the system control area, the user can control the system by controlling the coordinates of a mouse using his/her hand and through gesture recognition-based control. The user can also place an object on a predetermined area to gain information about it. To verify the excellence of the system proposed by this study, the recognition rates of snap gestures and the existence of objects were measured.

1 Introduction

Currently, diverse types of display devices are available and used for the presentation of information, including the CRT, LCD, and etc. In the last few years, touch displays which allow a user to interact directly with the screen have been developed as well. The display industry has seen a steady advancement, and displays with more advanced technologies have been recently developed, including flexible displays, transparent displays which can be seen through, etc. [1] It has also become possible, by using a natural user interface (NUI) sensor [2], to control displays in a more natural way.
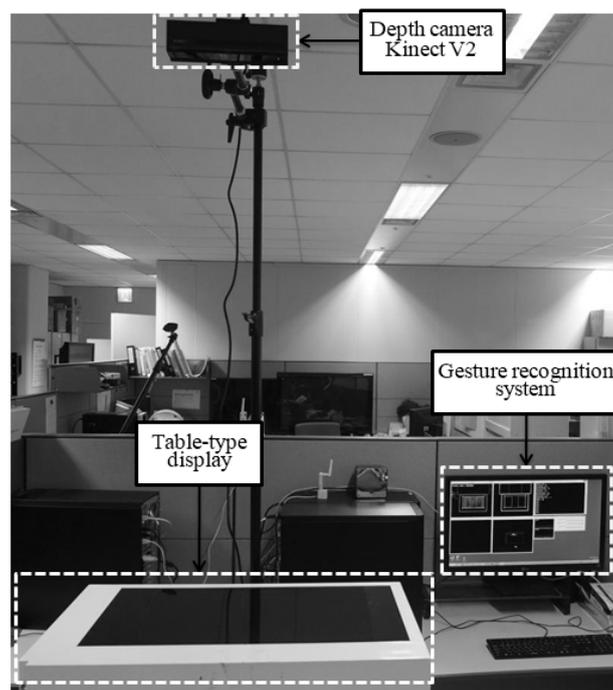
Fig. 1 Schematic view of the system

Various products with transparent displays have been launched as well. Examples of such products include a transparent refrigerator for a shop which allows people to see products inside it, and check product information on a display, a show case with a transparent display for people to check both products and product information at the same time, etc.
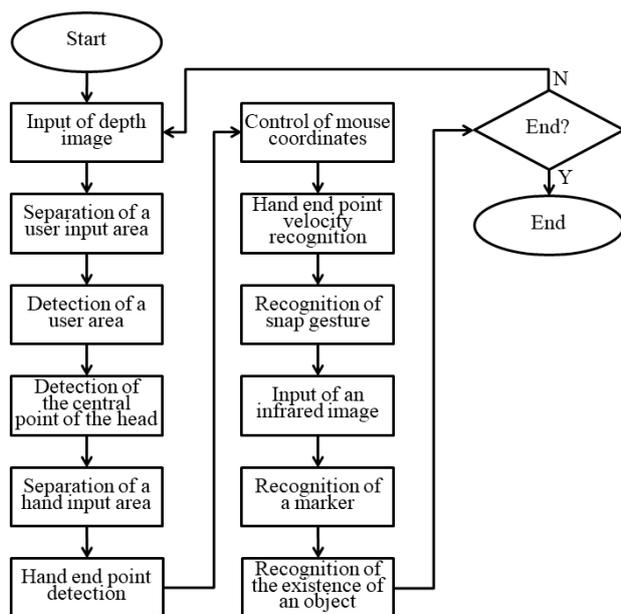
Fig. 2 System Flow Chart

This study proposes a method for user spatial interaction based on a transparent display, without the need for a user to touch the display directly. [Figure 1] shows a system equipped with the spatial interaction function proposed by this study, and [Figure 2] is a system flow chart for the spatial interaction process.

2 User Detection and Mouse Control

In order to conduct spatial interactions using the table-shaped system proposed by this study, user detection should be done first. To be able to determine whether there is a user or not, an area for user detection should be designated. [Figure 3] (a) is a photo of a user positioned in front of the system, and the rectangle of white dotted lines in [Figure 3] (b) is a user detection area. In order to determine whether or not a user has entered the detection zone, the depth information from a Kinect camera [3] is binarized by applying a threshold value. The largest blob of the binarized image is detected, and the rest is removed (refer to Figure 3 (c)). If the blob detected from the user detection zone has a value larger than a predetermined threshold, it is recognized that a user has entered the detection zone. If the relevant blob disappears from the user detection zone, it is

recognized that the user is out of the detection zone.
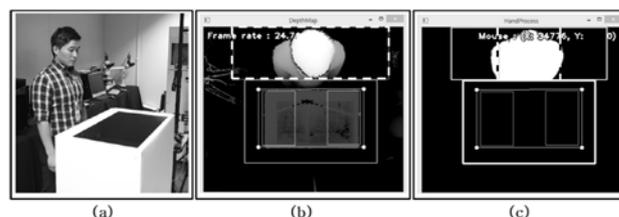
Fig. 3 User Detection

(a) entry of a user, (b) user detection zone, and (c) detection of a user
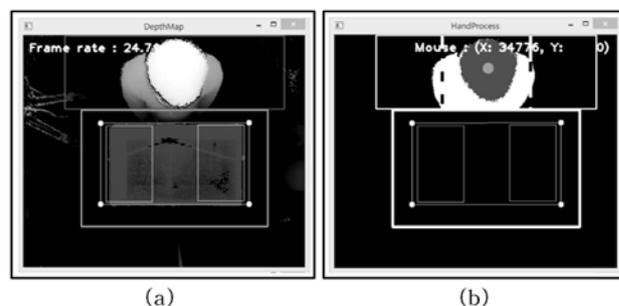
Fig. 4 Extraction of the Central Point of the Head

(a) user depth image, (b)   extraction of the area and central point of the head

For spatial interactions of this system, the control of a mouse is required, which may be done using a hand or the coordinates of the head. Here, the method for controlling the coordinates of a mouse using the coordinates of the head will be explained.

The control of the mouse coordinates is done based on a hand when a hand is entered, but if this is not the case, the coordinates of the head are used to control the mouse coordinates. A camera is installed at the top for downward observation. Referring to [Figure 4 (a)], the head of a user is expressed as a lighter-colored area than others. That is, a certain part of a user zone which is closer to the camera (lighter-colored image) is an area for the head. After determining an area for the head, the center of gravity of the head is determined, to calculate the center-of-the-head coordinates (refer to Figure 4 (b)). Referring to Figure 4 (b), the x-coordinate of the head within an area marked by white dotted lines in the user detection zone is mapped as the x-

coordinate of the mouse, and its y-coordinate is mapped as "0". In other words, if the coordinates of the head reach the white dotted lines on the left, the mouse coordinates are moved to the upper left edge of the screen.

3 Detection of Hand Area and Control of Mouse

For spatial interaction using this system, the control of mouse coordinates can be performed using the end point of a user's hand.

(a)

(b)   (c)

Fig. 5 Detection of Hand Area

(a) hand gesture, (b) hand detection zone, and (c)

detection of hand and its end point

[Figure 5] (a) shows a user moving his hand over the system. The zone of the white rectangle of [Figure 5] (b) is the hand detection zone. As shown in [Figure 5] (c), when a hand is placed over the screen, by applying a threshold to depth information from a Kinect camera, the table image can be removed, leaving only the hand zone image. When the end of a hand is located at the bottom of the hand zone, a certain area from the y value of the lowest point is recognized as an end-of-the-hand area, of which center-of-gravity is gained to finally designate the area. Referring to [Figure 5] (c), the white point within the hand detection area is the end point of a hand.

Fig. 6 Control of Coordinates of Mouse

As shown in [Figure 6], the coordinates of the end point of a hand extracted this way are mapped into a rectangular screen which is formed by the coordinates of the four edges and expressed by white dotted lines. The warping matrix [4] can be calculated from the four coordinates of a camera image and the four coordinates of the screen, which is actually mapped. The mouse coordinates corresponding to the screen can be gained by multiplying this matrix by the coordinates of the end point of the hand.

Shaking of the mouse coordinates can be caused due to the resolution of the camera and shaking of the coordinates of the end point of a hand. This can be resolved by applying a low-pass filter [5].

4 Recognition of Snap Gestures and Existence of Objects

The process for recognizing gestures and objects required to conduct spatial interactions proposed by this study is as follows

4.1 Recognition of Snap Gestures

The screen can be controlled using snap gestures without the need to directly touch the table for spatial interactions.
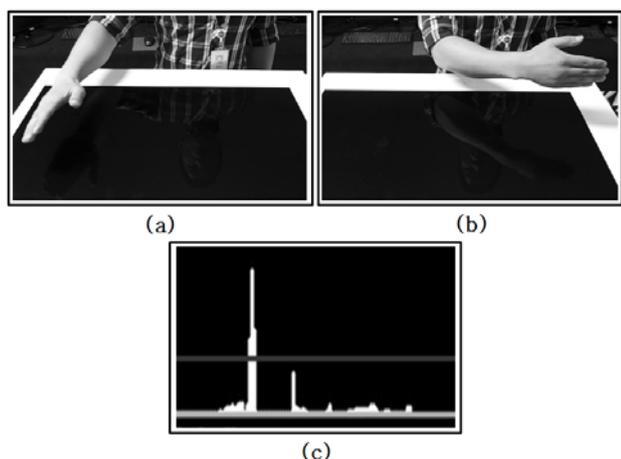
Fig. 7 Snap Gesture to the Left

(a) starting gesture, (b) ending gestures, and (c) Speed graph
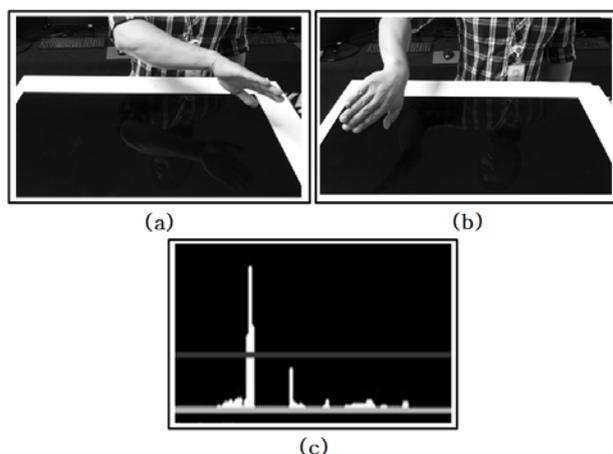
Fig. 8 Snap Gesture to the Right

(a) starting gesture, (b) ending gestures, and (c) Speed graph

The speed of the x-axis of the end-of-the-hand coordinates is measured to recognize the left or right snap gesture of the end of a hand. At this time, to reduce errors, the x-coordinates of the end point of a hand are found from three consecutive frames, and the average speed is calculated. If the average speed calculated this way is larger than the speed threshold value defined to the left or right direction, it is recognized as a snap gesture to the relevant direction. [Figure 7] shows a snap gesture to the left direction. [Figure 7] (c) is a graph showing the speed change from the starting gesture of [Figure 7] (a) to the ending gesture of [Figure 7] (b). If the speed is faster than the threshold value designated in [Figure 7] (c), a snap event to the relevant direction is generated. [Figure 8] shows a snap gesture to the right direction. [Figure 8] (c) is a graph showing the speed change from the starting gesture of [Figure 8] (a) to the ending gesture of [Figure 8] (b). If the speed is faster than the threshold value designated in [Figure 8] (c), a snap event to the relevant direction is generated.

4.2 Recognition of Existence of Objects

When a product inside a table is taken out and placed on the table, the existence of the product is recognized and information on the product is presented. A reflective IR marker is used to determine whether or not an object exists. The part marked by a white circle in [Figure 9] is a reflective IR marker.
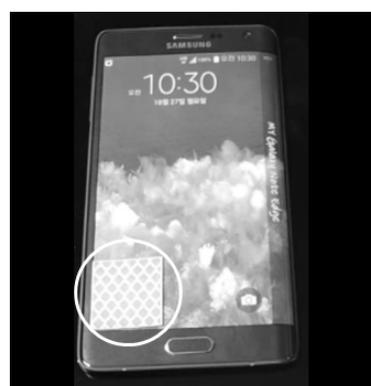
Fig. 9 Reflective IR Marker
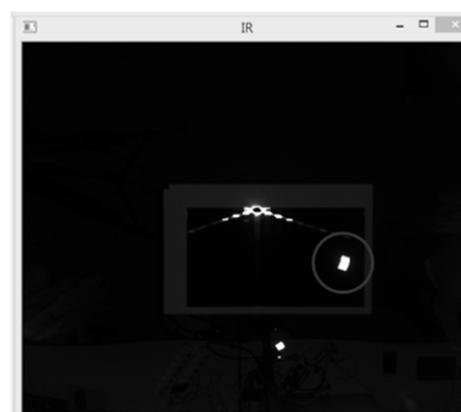
Fig. 10 Recognition of Existence of Objects

As shown in [Figure 10], when a reflective IR marker is used, the location of the marker can be checked clearly from a camera IR image (see the blob inside the circle in [Figure 10]). The problem with a screen made of glass reflecting infrared light

can be resolved by storing the background image when initiating the program and using a difference image.

If a marker within the designated area is recognized using an infrared image, the existence of an object is recognized. In [Figure 11], Rectangular A marked by dotted lines is Area A, and Rectangular B marked by dotted lines is Area B. When an object appears, Area A produces a keyboard message 'A', and Area B produces a keyboard message 'B'. For the stable change of a state when a marker is removed, the program has been set to recognize a certain state only when the state is maintained during a predetermined number of frames.
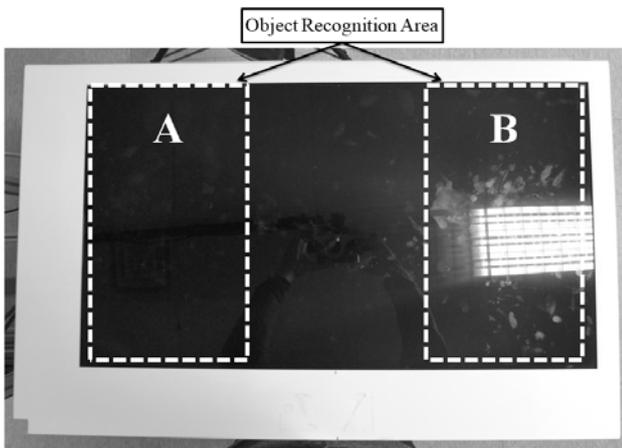
Fig. 11 Object Recognition Area

5 Results of Experiments

The experiments were conducted using a laptop PC with Intel(R) Core(TM) i7-3610QM 2.3GHz CPU and 16Gbyte memory, connected to a Kinect depth camera..
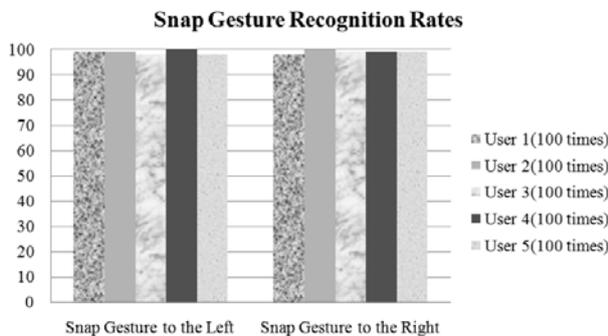
Fig. 12 Snap Gesture Recognition Rates

The In orders to verify the performance of this system, experiments were conducted to measure snap gesture recognition rates and rates of recognizing the existence of objects.

For snap gesture recognition rates, five users participating in the test moved their hand under a camera, and the hand recognition rate was gained.

For this experiment, each of the five users made snap gestures 100 times to the left and to the right respectively, and their average recognition rate was calculated [Figure 12]. [Figure 13] shows the average recognition rates of snap gestures to the left and right. As the results of the measurement, 989 out of the 1,000 snap gestures in total were successfully recognized, making the recognition rate 98.9%.
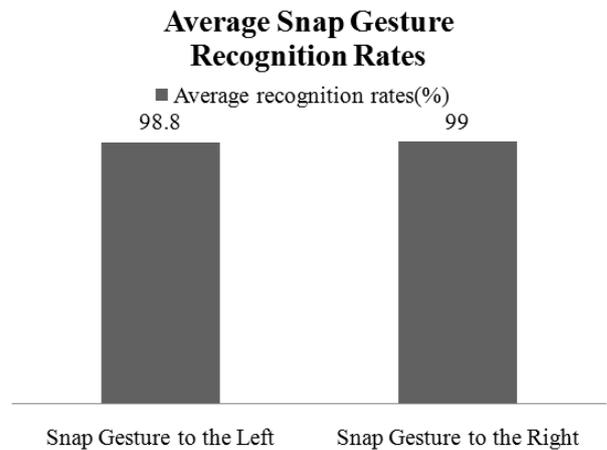
Fig. 13 Average Snap Gesture Recognition Rates
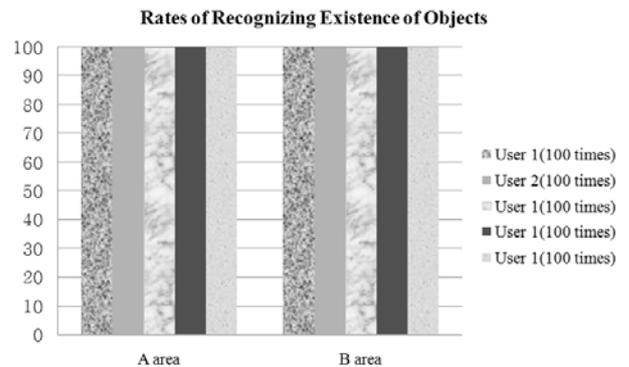
Fig. 24 Rates of Recognizing Existence of Objects

To test the recognition of objects, an object was placed on Area A or Area B under a camera, and the recognition rate was calculated. Each of the five users participating in the test were asked to put an object on Area A and Area B 100 times respectively, and the average recognition rate was

calculated (refer to [Figure 14]). [Figure 15] shows the average object recognition rates for Area A and Area B. As the results of the measurement, the recognition rate was found to be 100%, meaning the recognition of the existence of an object was made successfully each of the 1,000 times this test was repeated

Average Recognition Rate of Existence of Object

■ Average recognition rates(%)
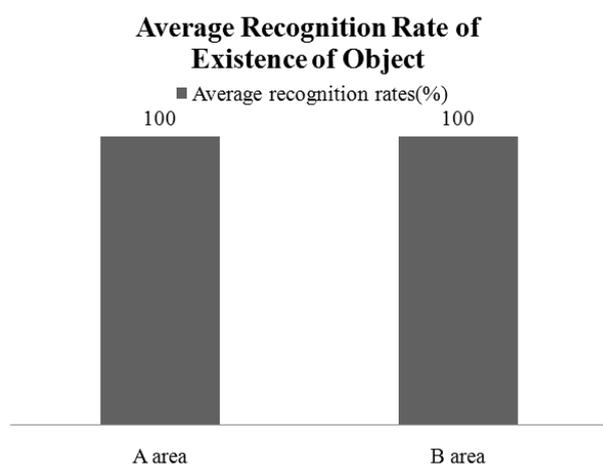
Fig. 35 Average Recognition Rate of Existence of Object

[Figure 16] shows a user conducting spatial interactions using actual contents. In [Figure 16] (a), a user is moving contents on the screen to the left through a snap gesture to the left, and in [Figure 16] (b), a user is comparing information on an object after placing the object on the table
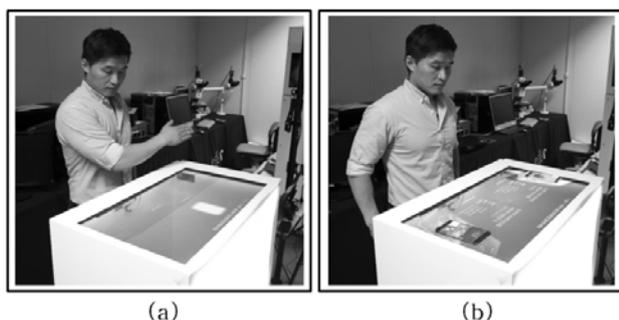
Fig. 46 User Test

(a) Snap gesture to the left, (b) recognition of existence of an object

6 Conclusion

In this study, a method for controlling information presented on a table-type display using a single depth camera, without any actual touching of the device, was proposed. The hardware configuration for applying the proposed method was explained, and the overall process was described. Using the proposed system, a user was able to conduct spatial interactions through hand movements in the air. Further, it was possible for the user to place an object on the table and check information related to it following the system's recognition of the existence of the object.

References:
[1] http://www.lgtfd.com
[2] D.A Norman, "The Way I See It: Natural User Interfaces Are Not Natural," Interactions, 2010.
[3] http://www.microsoft.com/en-us/kinectforwindows
[4] Mikkel B. Stegmann, "Image Warping", Informatics and Mathematical Modeling, 2001.
[5] http://en.wikipedia.org/wiki/Low-pass_filter