Implementation of Spatial Interaction System Using Transparent Display

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 a 3D entity shown on a transparent display by distinguishing between the hand and object using a single depth camera and by extracting relevant information for each. The hardware configuration for controlling a 3D entity has been presented. To enable the control of a 3D entity, a target area where the distinction between the hand and the object is made is extracted in the preprocessing stage. The extracted target area images are normalized to an identical size, and projected onto Zernike moment basis functions. The database is built based on moment values converted from projected images to distinguish the hand from object when input images are presented on a real time basis. This study also presents a method for interacting with a 3D entity using hand and object. To validate the performance of the system, an evaluation of recognition rate and time was performed.

Key-Words: - Hand Detection, Object Recognition, Air Touch, Transparent Display

1 Introduction

With the recent focus on smart functions, smart devices can be commonly found in diverse places. Along with this trend, the ways of utilizing information displayed on a screen have been diversified as well. A widely adopted method for mobile devices is the touch screen method. However, this method requires a users' direct touch, and cannot use information on a 3D basis. Further, its weakest point is that the areas being touched cannot be seen.

In an attempt to overcome such shortcomings, diverse methods have been proposed where information (entity) on a screen is used on a 3D basis using a depth camera. In the previous study[1], a depth camera was installed on the ceiling, and the information on a screen was controlled through a transparent display by using a hand gesture behind a screen and its matching coordinates on the screen.

This study proposes a system where a depth camera is installed on the floor to reduce spatial constraints (and therefore, the system installation is possible in places without a ceiling). Also, unlike previous systems where control was done only by hands, the system proposed by this study enables a 3D-based interaction using recognized objects. Figure 1 shows the hardware configuration of the system which includes a depth camera for calculating the location of a head, a depth camera for recognizing objects and gestures, a transparent monitor, and a PC.

The system has three modules for calculating the coordinates of the head location, object recognition and interaction processing, and result confirmation, and transmits/receives messages using the UDP communication. Figure 2 shows the overall flow of system operation.

2 Preprocessing

In the preprocessing stage, the target area is detected from an input image. While there was the distance limit (from ceiling to floor) in the previous study[1] since a camera was installed on the ceiling, there is no such limit in this study as a camera installed on the floor is directed upward. Therefore, to improve controllability, as shown in Figure 3, the usage range (i.e., target area) was set to be 20–60cm from the camera.

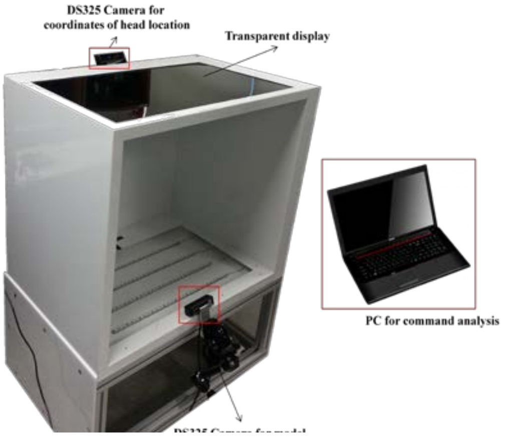

Fig. 1 Hardware Configuration

As shown in Figure 4(a), depth images are obtained from a depth camera installed on the bottom part of the system for recognizing objects and gestures. Referring to Figure 4(b), if the binarization of a received image is done based on a given threshold, there is a risk of system misoperation due to the existence of unnecessary information. Therefore, to eliminate unnecessary areas, a process for removing a background is performed. Referring to Figure 4(c), to eliminate the background of a designated place, a background learning process is conducted, where the input images of certain frames are stored as temporary image storing variables. When the learning process is done, as shown in Figure 4(d), the background is removed. Then, images which are input afterwards are shown without the image of the background.

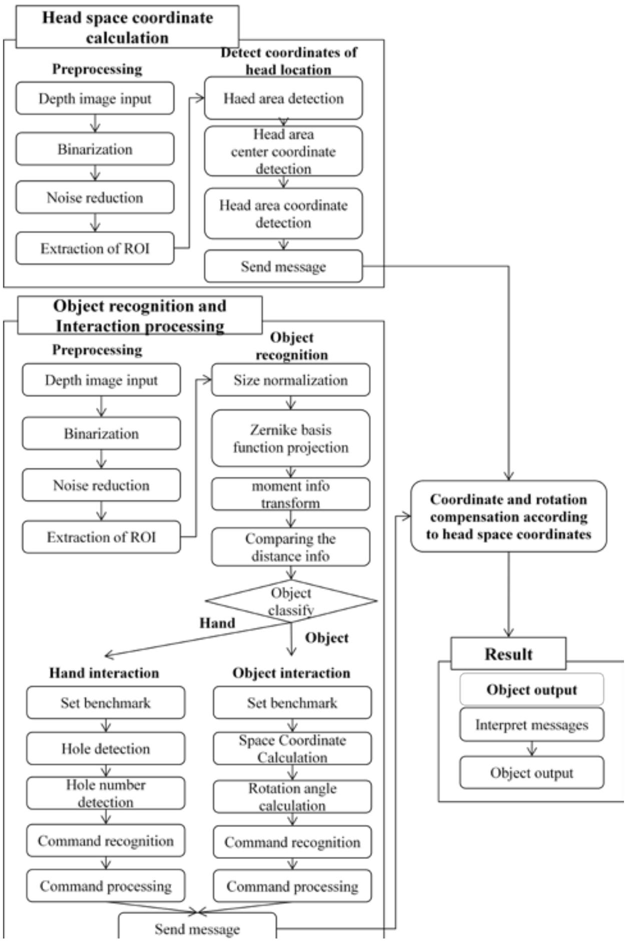

Fig. 2 Overall System Flowchart

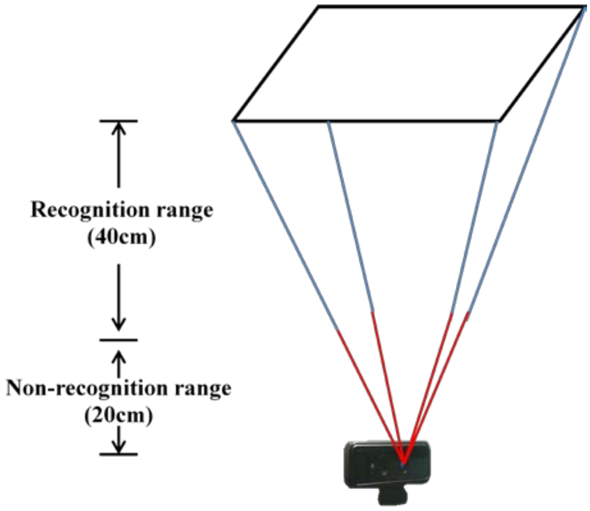

Fig. 3 Recognition Range of Camera

Then, for system stabilization, a noise elimination process is performed. In this study, the process was performed twice, since the noises which are caused when a hand or object is input are variable. First, the noises are eliminated using a median filter (refer to Figure 5(a)), and then the area of a hand or object is detected from the input image (refer to Figure 5(b)). For area detection, a labeling algorithm was applied.

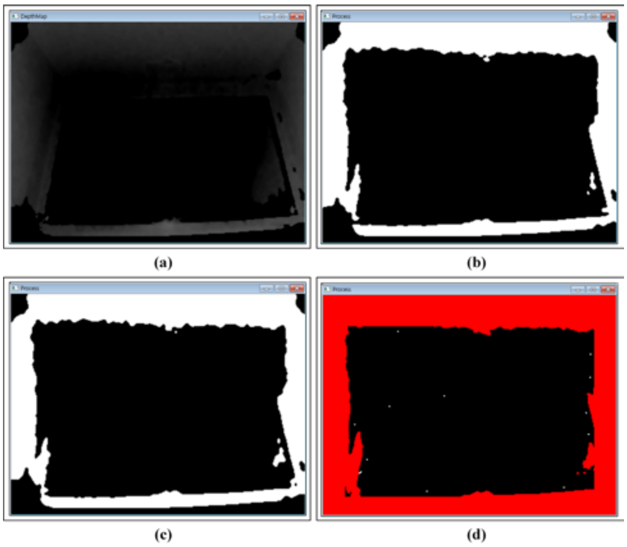

Fig. 4 Background Learning Process (a) input image (b) binarization (c) background learning (d) background elimination

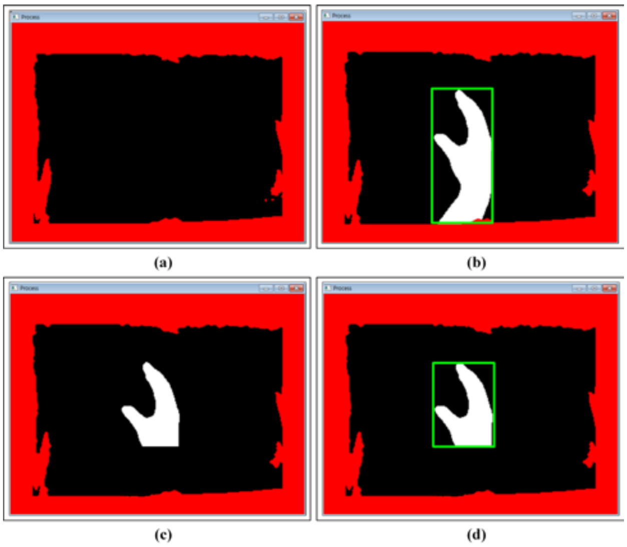

Fig. 5 Preprocessing (a) first noise elimination (b) blob detection (c) wrist elimination (d) detection of target area

When an area is detected for the first time, a user's wrist is also detected, and therefore should be removed as an unnecessary part. For the elimination of wrist part, the moment is used to find out the center of gravity and the part lower than the center of gravity is temporarily removed (refer to Figure 5(c)). The reason that the temporary elimination is done is because the elbow can be mistakenly detected as the center of gravity when detecting a target area based on the distance transform technique. However, if the distance transform is conducted after removing the wrist part temporarily, the center of gravity of a hand or object is detected correctly. Based on the resulting information, the target area is detected as shown in Figure 5(d). Then, the pixels excluding the target area are considered to be noise and removed (the secondary noise elimination).

3 Entity Recognition

When the preprocessing is completed, the Zernike moment[2], known as an efficient analyzer of shape information, is used to recognize an image as a hand or object.

3.1 Zernike Moment

The Zernike moment is defined as complex orthogonal moments where the absolute value of the moment is rotation-invariant[3] and projects an input image onto a basis function. The basis function of the Zernike moment with the order of n and repetition of m can be defined as Formula (1) below.

$$V_{nm}(x,y)=V_{nm}(\rho,\theta)=R_{nm}(\rho)\exp(jm\theta) \quad (1)$$

Wherein, V_{nm} means a set of orthogonal polynomials within a unit circle in the space of the polar coordinates. Therefore, there is no information overlaps. n is 0 or a positive integer, and m is a non-negative integer which meets the following two conditions of (1) $n - |m|$ = even number and (2) $|m| \leq n$. ρ is the distance from (0,0) to (x, y), being effective within the range of $0 \leq \rho \leq 1$. θ signifies the angle that the point (x, y) makes with the x axis, being effective within the range of $0 \leq \theta \leq 2\pi$.

$R_{nm}(\rho)$ is a Zernike real-number radial polynomial, and can be expressed as Formula (2).

$$R_{nm}(\rho)=\sum_{s=0}^{(n-|m|)/2}(-1)^{s}\frac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!}\rho^{n-2s} \quad (2)$$

Zernike moment Z_{nm} with the order of n and the repetition of m can be defined as Formula (3) below.

$$Z_{nm} = \frac{n+1}{\pi}\iint_{x^2+y^2\le 1} f(x,y)V^*_{nm}(x,y)dxdy \qquad (3)$$

Wherein, V^* means complex conjugate. To get t he Zernike moment of a discrete image, Formula (3) can be expressed as Formula (4) by approximating t he Zernike moment equation.

$$Z_{nm} = \frac{n+1}{\pi}\sum_{x}\sum_{y} f(x,y)V^*(x,y), x^2+y^2 \le 1 \qquad (4)$$

3.2 Distinction between Hand and Object

As shown in Figures 6(a) and 6(b), two objects were used for the object-based control of a 3D entity. In the case of a hand, the distinction can be done regardless of hand gestures (refer to Figure 6(c)).

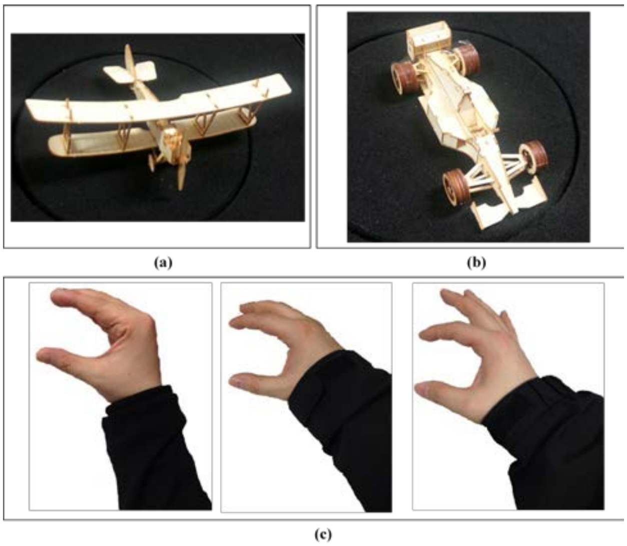

Fig. 6 Object and Hand used for Control of 3-dimensional Entity (a) airplane model (b) car model (c) hand gestures

Referring to Figure 7(a), for the distinction between object and hand, first of all, the images for which the preprocessing had been done were normalized to an identical size. In this study, the size was set to be 65x65. After the normalization of size, as shown in Figure 7(b), the images were projected to Zernike basis functions. Then, the images were converted to moment figures. The number of Zernike moments was determined based on the order. In this study, a 15thorder Zernike moment was used, and the total number of moments were 72.

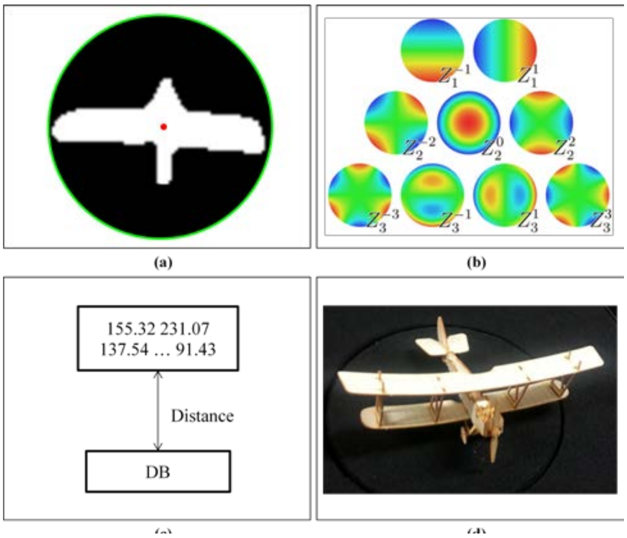

Fig. 7 Process for Recognition of Hand and Object (a) size normalization (b) Zernike base function projection (c) comparison of distance information (d) entity classification

In this study, the binary images of an airplane model and car model were converted to Zernike moment information and stored in the database. For each object, the database of 30 classes was stored. The images, which are input on a real time basis, are converted to moment information to be compared with the database. Then, the distance information is compared as shown in Figure 7(c). The classification is done to an object with the smallest distance value, and if the distance is over a given threshold, the classification is done according to the system design (refer to Figure 7(d)). Figure 7 shows the process from the input of an airplane image to the final classification.

4 3-dimensional Entity Control

The method for controlling a 3D entity using hand and object will be explained below.

4.1 Control of 3D Entity using Hand

There are two gestures used to control a 3D entity using a hand in the system: the gesture of picking up and the gesture of standby. Figures 8(a) and 8(b) show the gestures of the left hand and right hand on standby status respectively. Figures 8(c) and 8(d) show the pick-up gestures of the left hand and right hand respectively in which the thumb and index finger are put together.

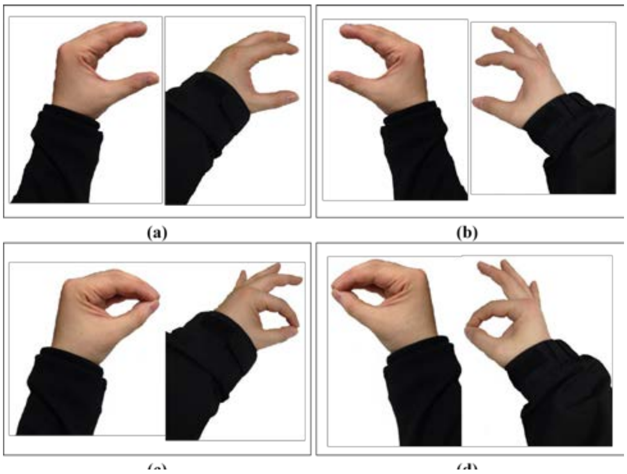

Fig. 8 Hand Gestures Used in System (a) left hand on standby status (b) right hand on standby status (c) pick-up gesture of left hand (d) pick-up gesture of right hand

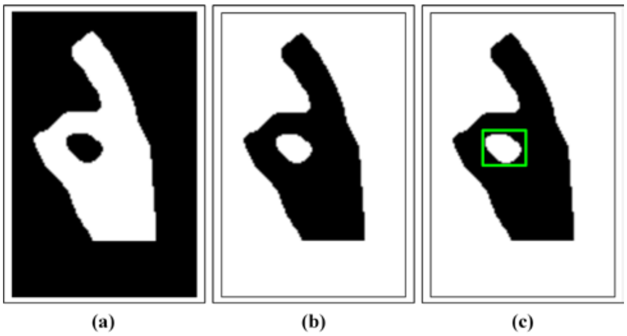

Fig. 9 Pick-up Gesture Classification (a) hand area image (b) reversal of image (c) detection of hole

To distinguish the pick-up gesture from the standby gesture, the detection of a hole is done. If an input image is recognized as a hand, the detection of up to two blobs is done. If a hole is detected in each blob, the image is recognized as the pick-up gesture. In order to detect a hole in a hand area, as shown in Figure 9(b), the input image (Figure 9(a) is reversed. If a blob is detected in a hand area of a reversed image, the pick-up gesture is recognized (Figure 9(c)). One caution is that a detected blob about window size should be treated as an exception.

The control of a 3D entity is possible using the recognized pick-up gesture, and Figure 10 shows the control process. As shown in Figure 10(a), if you place one hand within an object, make the pick-up gesture, and move the hand, the object is moved. If you place one hand within an object, make the pick-up gesture, and move your hand up and down by more than a certain distance, the size-adjustment flag appears. Referring to Figure 10(b), if you move your hand upward, the size of an object is increased, and if you move your hand downward, the object size is decreased. Referring to Figure 10(d), if you place both hands within an object, make the pick-up gesture, and move your hands close to each other, the size of an object is decreased. If you move your hands far from each other, the object size is increased. Referring to Figure 10(d), if you place one hand within an object and make the pick-up gesture while putting the other hand outside the object with the pick-up gesture, and if you move your hand located within the object back and forth, the rotation of the object becomes possible.

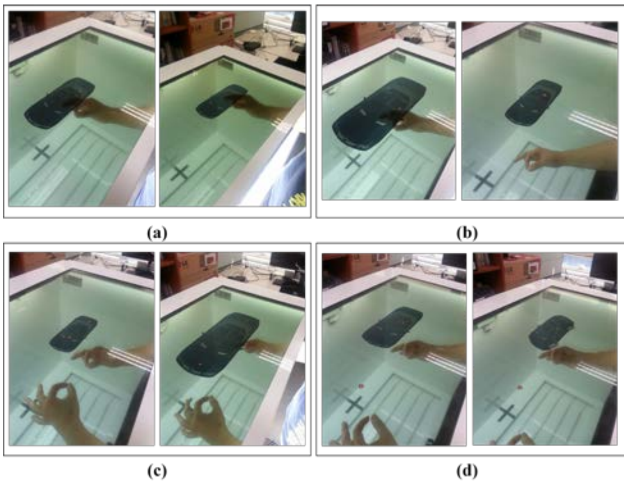

Fig. 10 Control of 3D Entity using Hand (a) object movement (b) adjustment of object size using one hand (c) adjustment of object size using both hands (d) rotation of object

4.2 Control of 3-dimensional Entity using Object

In the case of an object, the control of a 3D entity is done based on the location information about the object, distance to a camera, and rotation information. Figure 11 shows the process of controlling a 3D entity using an object. When a recognized image is an object, not a hand, the information on the recognized object is displayed on a screen. As shown in Figure 11(a), if the location of the object is moved, the 3D entity is moved to where the object is located. Referring to Figure 11(b), if the object is moved down, the size of the 3D entity gets smaller, and if it is moved up, the size of the 3D entity becomes larger. Referring to Figure 11(c), if the object is rotated to the left, the 3D entity is also rotated to the left, and if the object is rotated to the right, the 3D entity is rotated to the right as well.

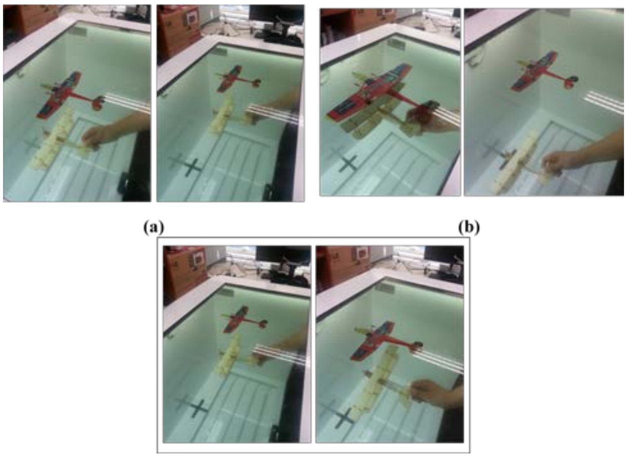

Fig. 11 Control of 3D Entity Using Object (a) movement of entity (b) adjustment of size of entity (c) rotation of entity

4.3 Correction of Location and Rotation based on Location of Head

The control of a 3D entity is done using hand and object, and the correction of the location and rotation of the entity is realized based on head location. This is to improve the feeling of user control. If a user moves his/her head to the left from its initial position, the entity rotates to the right. Referring to Figure 12, the rotation angle of the entity is the angle formed between (1) the straight line from the camera to the head and (2) the straight line from the camera to the location to which the head moved.

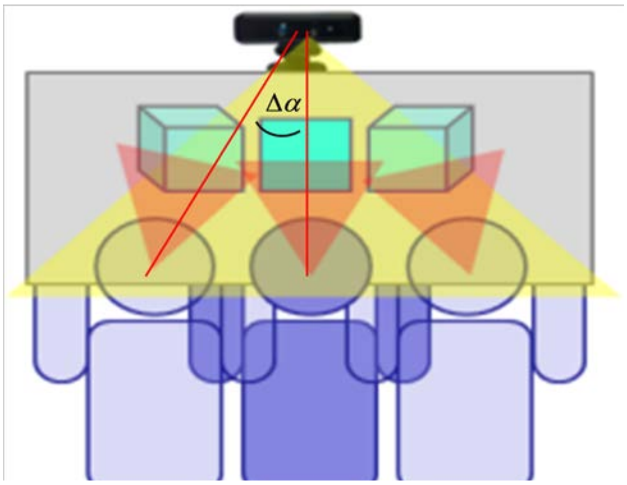

Fig. 12 Rotation of Entity according to Head Location

5 Experiment Results

The experiment was conducted using a PC with a CPU of Intel(R) Core(TM) i7-2600K 3.4GHz and 8 Gbyte memory where Depth Camera DS325[4] was connected.

To validate the performance of the system, the recognition rate and time of hand and object were measured.

To evaluate recognition rates, the hand or object was offered as an input at a designated place, and the rate of correct recognition was calculated. In the case of the hand, the distinction rate between the pick-up gesture and standby gesture was additionally calculated. The average recognition rate was also obtained by having each of the five users present an input 100 times for a total of five classes.

Figure 13 shows the graph of recognition rates of five users for each class, and Figure 14 shows the average recognition rate for each of the five classes. The overall average recognition rate was about 94% which is a satisfactory level.

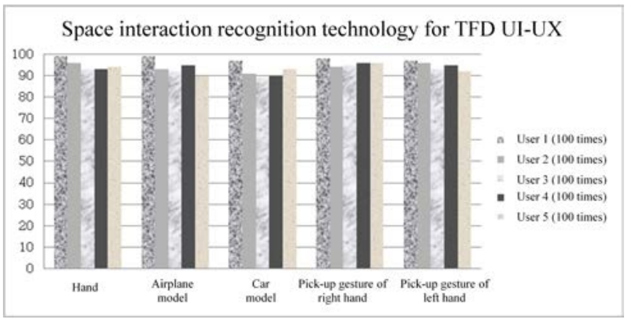

Fig. 13 Recognition Time by User

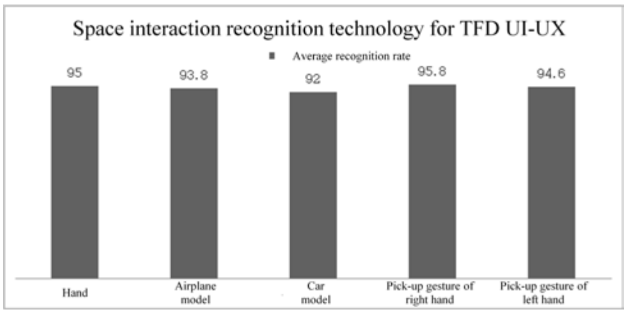

Fig. 14 Average Recognition Rates

For the evaluation of recognition time, the hand or object was offered as an input at a designated place, and the time of correct recognition was measured. The average recognition time was also calculated by having each of the five users present an input 100 times for a total of three classes.

Figure 15 shows the graph of recognition time of the five users for each class, and Figure 16 shows the average recognition time for each of the three classes. The average recognition time was about 280ms which means that the recognition was done immediately upon presenting an entity.

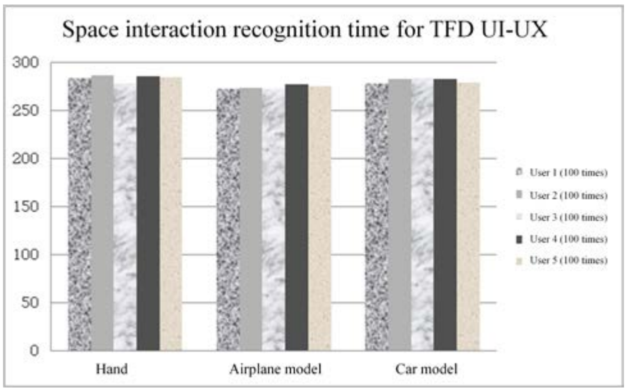

Fig. 15 Recognition Time by User

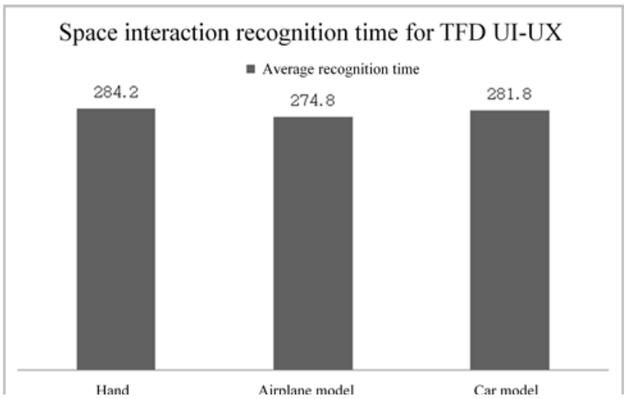

Fig. 16 Average Recognition Time

6 Conclusion

This study proposes a method for controlling a 3D entity shown on a transparent display by distinguishing between the hand and object using a single depth camera and by extracting relevant information for each. The hardware configuration for applying the proposed method and the overall system flowchart has been presented. Without using input devices such as keyboard, mouse or touch screen, the system proposed by this study enables users to control a 3D entity through a transparent display while looking at his/her own hands or objects presented as an input.

References:

[1] Jinha Lee, Alex Olwal, Hiroshi Ishii, and Cati boulanger, “SpaceTop : Intergration 2D and Spatial 3D Interaction in See-through Desktop Environment”. In Proceeding of SIGCHI, 2013, pp. 189-192.

[2] C. H. The, R T. Chin, "On Image Analysis by the Moment Invariants," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol, 10, no.4, 1988.

[3] A. Khotanzad, Y. H. Hong, "Invariant image recognition by Zernike moments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol, 12, no.5, pp. 489-497, 1990.

[4] http://www.softkinetic.com