

Implementation of Gesture Interface for Projected Surfaces

Yong-Suk Park¹, Se-Ho Park¹, Tae-Gon Kim¹, and Jong-Moon Chung²

¹Contents Convergence Research Center, Korea Electronics Technology Institute
Seoul, 121-835 - Republic of Korea
[e-mail: {yspark, sehopark, ktg2309}@keti.re.kr]

²School of Electrical and Electronic Engineering, Yonsei University
Seoul, 120-749 - Republic of Korea
[e-mail: jmc@yonsei.ac.kr]

*Corresponding author: Jong-Moon Chung

*Received October 6, 2014; revised December 4, 2014; accepted December 23, 2014;
published January 31, 2015*

Abstract

Image projectors can turn any surface into a display. Integrating a surface projection with a user interface transforms it into an interactive display with many possible applications. Hand gesture interfaces are often used with projector-camera systems. Hand detection through color image processing is affected by the surrounding environment. The lack of illumination and color details greatly influences the detection process and drops the recognition success rate. In addition, there can be interference from the projection system itself due to image projection. In order to overcome these problems, a gesture interface based on depth images is proposed for projected surfaces. In this paper, a depth camera is used for hand recognition and for effectively extracting the area of the hand from the scene. A hand detection and finger tracking method based on depth images is proposed. Based on the proposed method, a touch interface for the projected surface is implemented and evaluated.

Keywords: Interactive surface, natural user interface, gesture recognition, finger tracking, image projector, depth camera

A preliminary version of this paper was presented at APIC-IST 2014 (“Implementation of Hand Gesture Recognition Based on Depth Camera”) and was selected as an outstanding paper. This version includes details of the interactive surface projection system and modifications to the hand area tracking algorithm. This work was supported by the IT R&D program of MSIP/IITP [10047281, Remote Smart Collaboration Framework for Mixed Reality Interaction and Real-Time Data Sharing Among Remote Table Surfaces].

1. Introduction

Keyboards and mouse devices are perhaps the most familiar interface tools humans use to interact with computers. However, the ever growing popularity of smart devices have exposed and familiarized many with the comfort of natural user interfaces, such as multi-touch screens and spatial gestures. Vision and gesture-based human computer interaction is a prominent and ongoing research area, and significant research has been carried out to bridge the gap between the physical world and digital information, making human computer interaction more friendly and intuitive [1].

Displays and screens provide easy and comfortable visual cue for human computer interaction. Various kinds of information can be displayed for the human user to interact with. Image projectors can turn any surface into a display. Integrating a surface projection with a user interface transforms it into an interactive display with many possible applications. Hand gesture interfaces are often used with projector-camera systems. The projector projects the display on any surface providing visual information to the human user, and a camera is used to detect the hand gesture of the user for interaction, effectively creating an interactive display.

Many different methods are used for hand detection and gesture recognition. The method based on calculating image pixel differences using multiple cameras requires significant computation time, making it unsuitable for real-time applications. The method using color images depends on the surrounding environment which affects the recognition rate. Recently, low-cost depth cameras have become available in the market, enabling their use in hand detection and gesture recognition. A depth camera is a range imaging camera system that resolves distance based on the known speed of light. It measures the time-of-flight of a light signal between the camera and the subject to construct an image based on distance or depth.

In this paper, an effective hand detection method based on the depth information from a depth image camera is presented to be used as gesture interface for projected surfaces. The proposed method can be used to turn any smooth surface into an interactive workspace without compromising performance due to environmental conditions. The proposed method is not affected by environmental factors such as illumination or glare, color of the table, or hand cast shadows. In Section 2, previous works done in image projection and gesture-based input are reviewed. Section 3 provides details of the gesture interface which involves, hand detection and finger tracking. Implementation and evaluation of the gesture interface is provided in Section 4, by mapping the gesture interface to a mouse-control system. Concluding remarks and future work are discussed in Section 5.

2. Related Work

Numerous research has been carried out in integrating projector-camera systems with gesture interfaces. Xu et al. introduced a gesture interface based on the shadow derived by the projector [2]. Shadows can provide a simple interface between human and computer systems, but ideal lighting conditions are required for creating shadows that are well recognized. A computer vision based projected tabletop interface which can be controlled by finger gestures is proposed in [3]. A webcam is used for tracking fingers offering economic and practical way of interaction. However, this solution is also influenced by lighting conditions of the environment. In addition, a touch-less, gesture-based computer control system with dedicated

interactive whiteboard application that uses versatile components, i.e., a PC, webcam, and multimedia projector, has been presented in [4]. Goto et al. have also developed an interface system that enables the user to input and obtain information in the environment with a projector-camera system [5]. This system uses skin color detection for hand detection, which can be influenced by the environment. Shah et al. have considered portability and mobility for the projector-camera system and employed a pocket projector in their system [6]. The system proposed by Zeng et al. in [7] employs a thermal camera for robust human body segmentation to handle the complex background and varying illumination posed by the projector. While providing good performance, the use of thermal cameras may not be practical for common users. In [8], J. Hu et al. propose a bare-finger touch based interactive projection system that can be applied to mobile devices. The system is based on detecting the distortion of the projected button when a finger is placed on top of it. This system needs further work on detecting touch events on the projected screen. Tang et al. introduce Gesture Viewport in [9], a projector-camera system that enables finger gesture interactions with media content on any surface. Gesture Viewport uses detection of occlusion patterns inside a virtual sensor grid. Although computationally efficient, the virtual sensor grid is rendered on top of the view surface causing interference with the media content which maybe undesirable. Another off-the-shelf vision based projector-camera system for human-computer interface is introduced in [10]. Hand segmentation is done by combining contrast saliency and region discontinuity. Touch detection is done by exploiting the homography mapping between the projector's display panel and the camera's image plane. The system's hand segmentation processing needs improvement because computation will increase when it comes to processing multihand interface.

3. Gesture Interface

3.1 Overview

The depth camera used in the proposed projector-camera system is Kinect. However, any other depth cameras can be used. Depth cameras illuminate the scene with infrared light and measure the time-of-flight to determine the distance to the objects in the scene and build a depth image. The Kinect has a color and a 2 megapixel grayscale chip with IR (infrared) filter.

The overall flow of the gesture interface is as shown in Fig. 1. Depth image data is received from the depth camera. From the depth image objects are differentiated and the hand and fingers are detected. Fingertip detection is performed to determine the possible gestures and hand/finger movement. RANSAC (Random Sample Consensus) is used to determine the pivot or the orientation of the hand. The outline of the hand detected is smoothed. Convex-Hull is used to extract the fingertip. Further details of the gesture interface are given in subsequent sections.

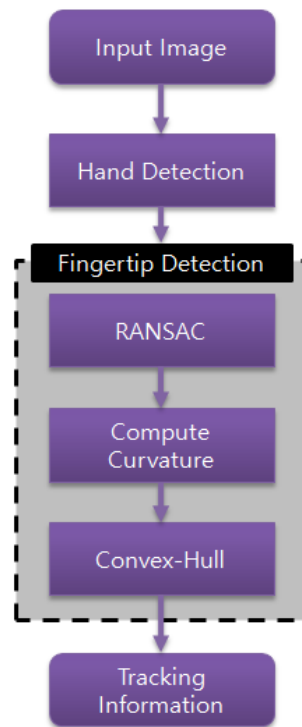


Fig. 1. Gesture block diagram

3.2 Object Detection

Background subtraction is performed using the depth images acquired to extract objects in the foreground. Using the depth information from a depth image camera, the static depth image information can be saved. By comparing the stored information with real-time depth image information, dynamic depth image information can be determined. If it is determined to be a dynamic depth image, moving objects can be recognized. If the static portion of the image is given a value of 0, an image with dynamic depth information can be extracted as shown in [Fig. 2](#).

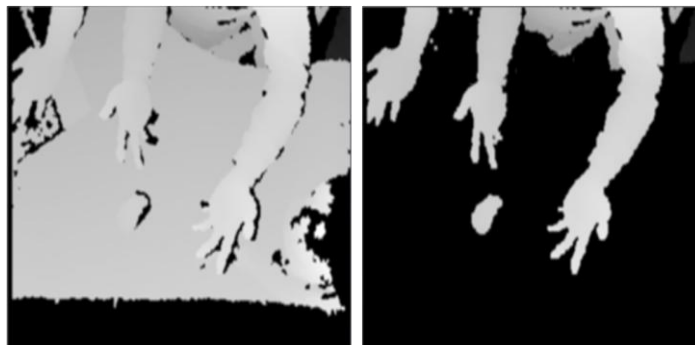


Fig. 2. Dynamic depth image information

Binary conversion is performed on the information from the depth image acquired, and Blob labeling algorithm is applied as shown in [Fig. 3](#). Blob labeling, also known as

connected-component labeling, is a well known method used in hand detection [11], [12]. After the objects are segregated by labeling the dynamic objects, pixels with values above or below the set threshold values are removed. Noise is further eliminated through the erosion and dilation process. The region of interest within the hand area can be set, and by using the number of pixels in the hand area, the center of the hand area can be inferred.

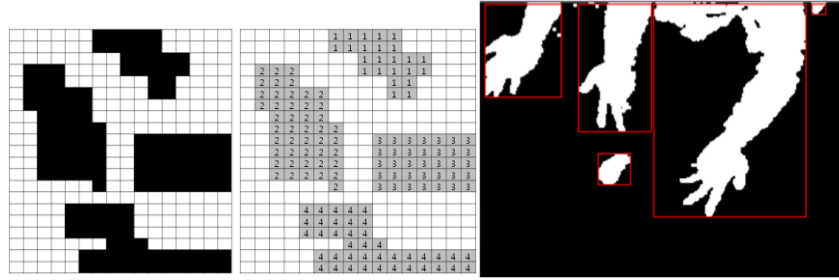


Fig. 3. Blob labeling of a binary image

3.3 Hand Area Detection

Blob labeling is a detection method that can be used not only with depth image cameras but also with color video cameras. However, hand detection using color images is not reliable when there is not enough illumination in the surrounding environment. If poor quality color is acquired then the detection rate drops as well. Therefore, depth image camera is used because it is not influenced by the surroundings. The data obtained from the depth camera undergoes blob labeling and the center of the hand area is inferred. The hand area is then extracted using area expansion method based on depth difference. The depth value is used as the criteria for area expansion. If the depth value is similar, the area is expanded. In order to avoid drastic area expansion, a global threshold value is used to limit area expansion.

$$Cond_{all} = \begin{cases} \sum_{i=1}^4 Cond_i = 4 & True \\ other & False \end{cases} \quad (1)$$

$$Cond_1 = \begin{cases} |D_i(n,m) - D_{i-1}(c,r)| < Th_{Depth} & 1 \\ other & 0 \end{cases} \quad (2)$$

$$Cond_2 = \begin{cases} D_i(n,m) - C_{Depth} < Th_{GlobalDepth} & 1 \\ other & 0 \end{cases} \quad (3)$$

$$Cond_3 = \begin{cases} Grow_{count} < Th_c & 1 \\ other & 0 \end{cases} \quad (4)$$

$$Cond_4 = \begin{cases} Dist < Th_c & 1 \\ other & 0 \end{cases} \quad (5)$$

Equations (1) to (5) are used to determine area expansion. $D_i(n,m)$ in (2) is the depth value of the pixel used to determine current expansion where n and m are x and y coordinates, respectively. $D_{i-1}(c,r)$ is depth value of the pixel from the previous expansion where c and r are

x and y coordinates, respectively. Based on the depth value of the base coordinate, if it is below threshold value (Th_{Depth}), then the area is expanded. The threshold value is used to define the connection between the object's previous pixel and pixel considered for expansion. If the difference in depth value of the pixels being compared is less than the threshold, then it is most likely hand area. Otherwise, if the difference in depth value is greater than threshold, then it is most likely background. (3) provides the conditions to limit the area of expansion. It compares the center point (C_{Depth}) with the global threshold value ($Th_{GlobalDepth}$) to restrict the range of the allowed depth for area expansion. Th_c in (4) is the threshold value count. Through experimentation, the depth value at the center of the hand is set by using a first degree polynomial as the threshold value. In order to restrict the expansion area size, the expansion count ($Grow_{count}$) is accumulated so that the range is restricted up to the value of threshold count (Th_c). (5) limits the area expansion range by using the Euclidean distance of the pixel to be expanded from the center coordinate. Conditions are checked for 8 surrounding directions relative to the center point. For all pixels considered for expansion, if the conditions of (1) are satisfied then the area is expanded. If conditions of (2) are not met, then it is indicated as a green contour. If conditions of (3) to (5) are not satisfied, then it is determined as a pixel that exceeds the threshold area and indicated as a yellow line, meaning an invalid boundary line. Fig. 4 shows the screen that displays the pixels that satisfy the area expansion condition from the center point and the 8 direction detection method for each pixel.

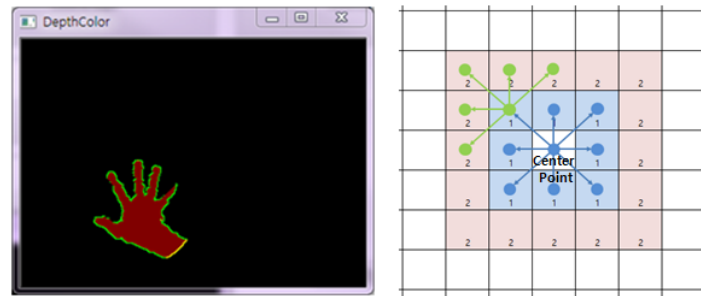


Fig. 4. Image after completion of area expansion and detection method in 8 directions

3.4 Fingertip Extraction

Using the previous area expansion, the border of the hand area can be detected but it contains a lot of noise. K -cosine is used for finger detection [13]. The noise can affect the computation of K -cosine and generate errors. Therefore, diverse methods such as first order differentiation, second order differentiation, Laplacian, etc., should be used as contour extraction methods to renew the contour and perform smoothing of the contour on its respective coordinates.

$$C_i(K) = \cos \theta_i = \frac{a_i(K) \cdot b_i(K)}{\|a_i(K)\| \cdot \|b_i(K)\|} \quad (6)$$

After detecting the contour of the hand, (6) is used to define $a_i(K) = P_{i+k} - P_i$ and $b_i(K) = P_{i-k} - P_i$. θ is defined as the angle between $a_i(K)$ and $b_i(K)$ as shown in Fig. 5.

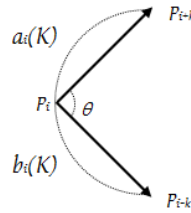


Fig. 5. Measurements using K -cosine

The P_i at the fingertips has $a_{iy}(K) > 0$ and $b_{iy}(K) > 0$ characteristics. The P_i at the valley between the fingers has $a_{iy}(K) < 0$ and $b_{iy}(K) < 0$ characteristics. The P_i at the edge of the hand other than fingers has $a_{iy}(K) \cdot b_{iy}(K) < 0$ characteristic. Using these characteristics, the value of the $\cos\theta_i$ that are not part of the fingers are all processed as 0, as seen in **Fig. 6**.

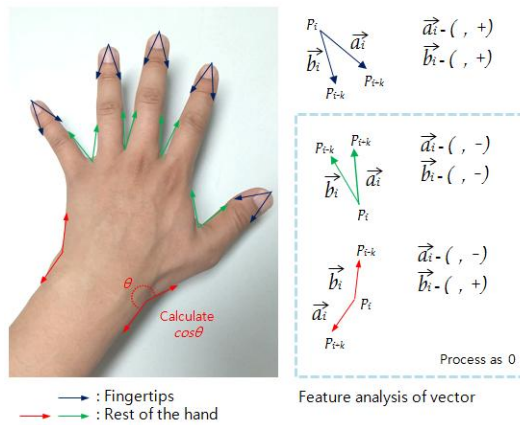


Fig. 6. Finger detection characteristics using K -cosine

Among the values of $\cos\theta_i$ obtained using K -cosine, the contours with $\cos\theta_i \neq 0$ are grouped in their corresponding arrays. The point with the greatest $\cos\theta_i$ value in each array is determined to be the hand endpoint. The number of hand endpoint detected becomes the number of fingers. **Fig. 7** shows the output image using K -cosine excluding 0 values and the output image with the largest $\cos\theta_i$ values as the hand endpoints.

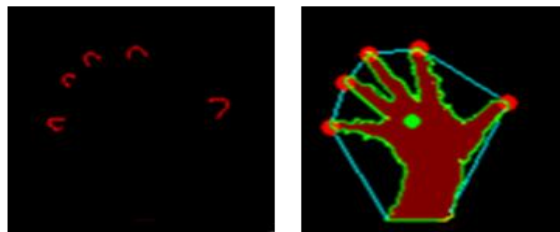


Fig. 7. Output of hand endpoints

3.5 Touch-point Setting

In previous methods, to output the point at the fingertip, information on hand movement, number of fingers used, etc. need to be supplied as input. In this paper, the intention is to implement touch gesture using the fingers over a table surface. Therefore, the depth camera is

positioned in such a way so that it is facing downwards towards the ground. Determining the touch gesture on the surface of the table relying only on fingertip information is not useful because of the big margin of error.

Using the fingertip as the base a set area is created. Using the center point of the area created, the center of the finger is determined. Using the hand endpoint obtained by K -cosine as the center, a circle is formed depending on the hand area depth data. The information of the image created and the existing hand area information is binary encoded and the AND operation is performed on the two images. A new image is then created using only the intersecting parts. Find Contour is used on the new image to determine how many divided pixels are there for each hand area object in order to find the center point of each group pixel. The center coordinate obtained is located approximately in the middle of the fingernail.

Fig. 8 shows the creation of the circle around the hand endpoint (b), results after performing AND operation with the binary image of the hand (c), and the center coordinates of the output pixels (d).

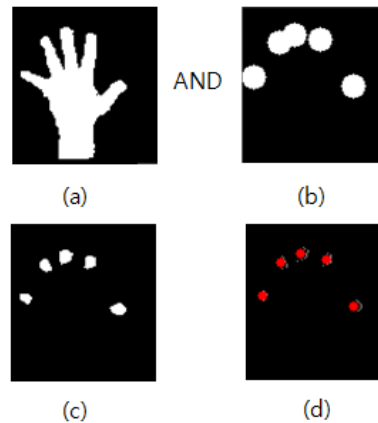


Fig. 8. Obtaining finger center points through AND operation

3.6 Hand Pivot Inspection

The Random Sample Consensus (RANSAC) algorithm is a method used to predict model parameters from noisy data [14]. It randomly samples a minimum amount of data necessary from the original data to determine the model parameters. The value is computed repeatedly until an optimal value is found. This method is opposite in concept from the traditional statistical methods. Most of the methods use as much data as possible to obtain an initial value, and based on the result irrelevant data are eliminated. RANSAC uses a small amount of initial data to expand the set of consistent data. RANSAC can be applied to the hand image in the following way. From the detected hand are two pixel points are randomly sampled. A linear equation is obtained from the two selected points. Points with distances that are below a certain threshold value are included in the linear set. A linear equation that includes most of the points in the set is computed. The number of pixels included in the linear equation is used to compute suitability. After several iterations, the linear equation with the best suitability is obtained which indicates the pivot of the hand area.

Fig. 9 shows the application of RANSAC on the hand area obtained. The pivot is determined even with diverse changes in the finger forms. Accurate rotation angle computation is possible when the hand is rotated or changed.

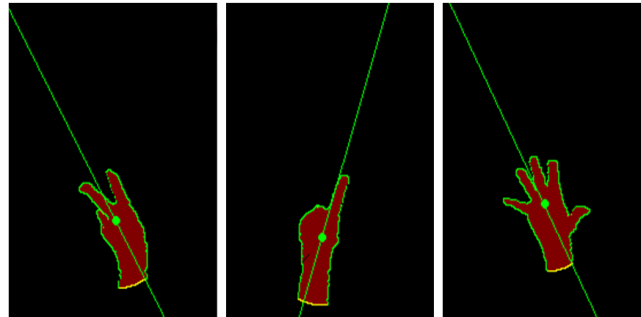


Fig. 9. RANSAC output results

4. Implementation and Analysis

Fig. 10 shows the basic projector-camera system setup. The PC used to process depth data and gesture related algorithms uses a 2.40 GHz processor, a 450 MHz GPU, and 4 GB of RAM. The application is built in Windows environment, using Kinect SDK and OpenCV. The depth information is obtained using the `imageStreamGetNextFrame()` API function, and the depth values are stored in a 16-bit 320x240 array. The stored depth values are quantized and mapped to grayscale values from 1 to 255 for visual representation.

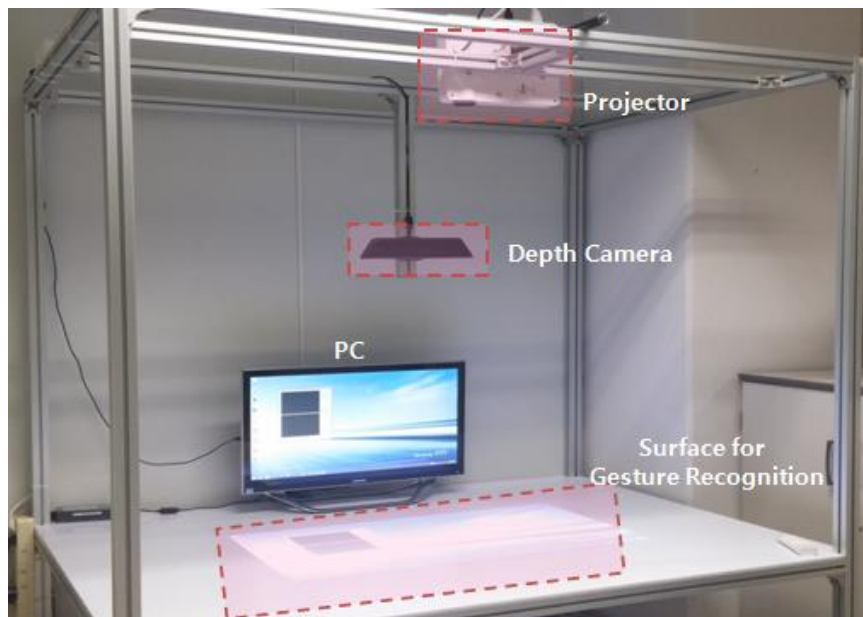


Fig. 10. Depth image and its corresponding cursor location on PC

Using the method from section 3.5, the finger center point is used to control the cursor and test the operation of the touch gesture at the location moved. Since the touch gesture system is still in its early stages of development, the movement portion of the interface is initially tested. The actual operation of the gesture will be further developed during the later stages. At this time, the movement of the finger is coordinated with the movement of the mouse pointer on the screen. `GetSystemMetrics()` Windows API function is used to get the screen resolution currently used. Dividing this by the depth image generation window size and multiplying the

result to the finger center point, we can make the cursor from the screen move proportional to the depth image (7).

$$center_{coord} * (screenSize / DepthSize) \quad (7)$$

Fig. 11 shows the how the finger center point from the depth image can be used to set the coordinates obtained from (7) to mouse coordinates. It can be seen that the location of the depth image resembles the Windows coordinates. For touch recognition, the object's height information saved in the table from section 2.1 is used. The depth value at the finger coordinates is compared with this information. If the difference is under a certain value, then it is in touch state. Otherwise, it is not. The mouse_event can be used to apply the UP, DOWN status of the mouse button. **Fig. 12** shows a snapshot of Windows manipulation using touch gestures on the projected surface.



Fig. 11. Depth image and its corresponding cursor location on PC

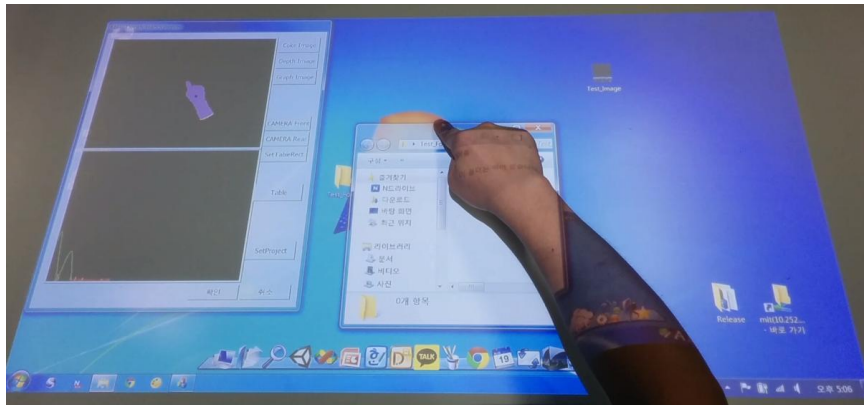


Fig. 12. Controlling the PC using touch gestures on the projected surface

Fig. 13 shows the graphical output representation to verify the touch gesture. The green line shows the depth or the height of the detected fingertip. The green peaks indicate that the fingertip is moving away from the surface or moving closer to the camera. The green valleys indicate that the fingertip is moving towards the surface or moving away from the camera. The red portion of the graph indicates that the fingertip is touching the surface. If two red areas are detected consecutively, then it is considered as a double-click. When you click and drag on the surface, the output is shown as in the right.

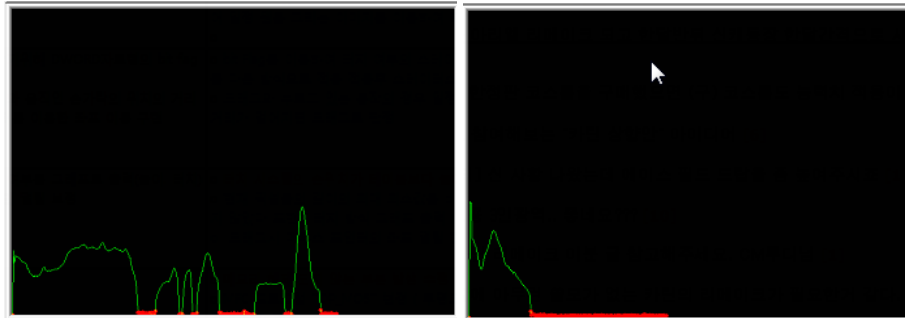


Fig. 13. Graph for touch verification

The average execution time of the key blocks of gesture recognition is shown in **Table 1**. It can be seen that the implementation is suitable for real-time use.

Table 1. Average execution time measured

Block	Time in seconds
Blob Labeling	0.00415
Hand Extraction	0.00125
Fingertip Extraction	0.00225
RANSAC	0.00343

5. Conclusion

In this paper, a gesture interface that can be used with projected surfaces is discussed. The projector-camera system proposed employs a depth camera, and the depth image data from the camera is used for hand detection. Since the depth camera measures the time-of-flight of a light signal between the camera and the subject for each point of the image, it is not susceptible to illumination or lighting conditions of the environment. The method developed does not use skin color information from color images, but relies solely on depth image information to detect the hand area. Therefore, it can detect the hand in dark places, and it is not influenced by the surroundings, enabling robust detection.

A method for determining fingertips from the hand area detected is also discussed. A preliminary discussion on its application for touch-based systems is also carried out. The fingertips are determined from the hand area. The centers of the fingertips are determined and their position converted to real coordinates to be used in a touch-based system. As future work, more work on touch-based system implementation needs to be done by creating diverse dynamic gestures.

References

- [1] S. P. Kumar and O. Pandithurai, "Sixth sense technology," in *Proc. of 2013 Int. Conf. on Information Communication and Embedded Systems (ICICES)*, pp. 947-953, February 21-22, 2013. [Article \(CrossRef Link\)](#).
- [2] H. Xu, D. Iwai, S. Hiura and K. Sato, "User interface by virtual shadow projection," in *Proc. of SICE-ICASE International Joint Conference 2006*, pp. 4814-4817, October 18-21, 2006. [Article \(CrossRef Link\)](#).
- [3] P. Song, S. Winkler, S. O. Gilani and Z. Zhou, "Vision-based projected tabletop interface for finger interactions," in *Proc. of IEEE Int. Workshop Human-Computer Interaction (HCI) 2007*, pp. 49-58, October 20, 2007. [Article \(CrossRef Link\)](#).

- [4] M. Lech and B. Kostek, "Gesture-based computer control system applied to the interactive whiteboard," in *Proc. of 2010 2nd Int. Conf. on Information Technology (ICIT)*, pp. 75-78, June 28-30, 2010. [Article \(CrossRef Link\)](#).
- [5] H. Goto, Y. Kawasaki and A. Nakamura, "Development of an information projection interface using a projector-camera system," in *Proc. of 19th IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 50-55, September 13-15, 2010. [Article \(CrossRef Link\)](#).
- [6] S. A. H. Shah, A. Ahmed, I. Mahmood and K. Khurshid, "Hand gesture based user interface for computer using a camera and projector," in *Proc. of 2011 IEEE Int. Conf. on Signal and Image Processing Applications (ICSIPA)*, pp. 168-173, November 16-18, 2011. [Article \(CrossRef Link\)](#).
- [7] B. Zeng, G. Wang and X. Lin, "A hand gesture based interactive presentation system utilizing heterogeneous cameras," *Tsinghua Science and Technology*, vol. 17, no. 3, pp. 329-336, June, 2012. [Article \(CrossRef Link\)](#).
- [8] J. Hu, G. Li, X. Xie, Z. Lv and Z. Wang, "Bare-fingers touch detection by the button's distortion in a projector-camera system," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 566-575, vol. 24, no. 4, April, 2014. [Article \(CrossRef Link\)](#).
- [9] H. Tang, P. Chiu and Q. Liu, "Gesture viewport: Interacting with media content using finger gestures on any surface," in *Proc. of 2014 IEEE Int. Conf. on Multimedia and Expo Workshops (ICMEW)*, pp. 1-2, July 14-18, 2014. [Article \(CrossRef Link\)](#).
- [10] J. Dai and C. R. Chung, "Touchscreen everywhere: on transferring a normal planar surface to a touch-sensitive display," *IEEE Trans. on Cybernetics*, pp. 1383-1396, vol. 44, no. 8, August, 2014. [Article \(CrossRef Link\)](#).
- [11] A. Rosenfeld and J. Pfaltz, "Sequential operations in digital picture processing," *Journal of the ACM*, vol. 13, no. 4, pp. 471-494, October, 1966. [Article \(CrossRef Link\)](#).
- [12] K. Wu, E. Otoo, and K. Suzuki, "Optimizing two-pass connected-component labeling algorithms," *Pattern Analysis and Applications*, vol. 12, no. 2, pp. 117-135, June, 2009. [Article \(CrossRef Link\)](#).
- [13] A. Rosenfeld and E. Johnson, "Angle detection on digital curves," *IEEE Transactions on Computers*, vol. C-22, no. 9, pp. 875-878, September, 1973. [Article \(CrossRef Link\)](#).
- [14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, no. 6, pp. 381-395, June, 1981. [Article \(CrossRef Link\)](#).



Yong-Suk Park is a managerial researcher at the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, Korea. Before joining KETI in 2003, he was with I&C Technology and Samsung S1, where he worked in projects relevant to wireless networks and system integration. He received his B.S. and M.S. degrees in electrical and computer engineering from Carnegie Mellon University in 1997 and 1998, respectively. He is currently working towards a Ph.D. degree in the School of Electrical & Electronic Engineering from Yonsei University, Seoul, Korea. His current research interests are in the areas of media sharing and contents delivery networks.



Se-Ho Park is currently a managerial researcher at the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, Korea. Before joining KETI in 2005, he was with I&C Technology and Samsung Electronics, where he worked in projects relevant to SoC (System on Chip) for digital broadcasting and wireless networks. He received his B.S. and M.S. degrees in electrical engineering from Kyungpook National University in 1998 and 2000, respectively. His current research interests are in the areas of digital broadcasting and wireless mobile networks.



Tae-Gon Kim is currently a researcher at the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, Korea. He was with Hyundai Mobis as an ECU (Engine Control Unit) chip engineer before joining KETI in 2013. He received his B.S. in Chungkang College of Cultural Industries in 2003. His current research involves augmented reality and natural user interface design.



Dr. Jong-Moon Chung received the B.S. and M.S. degrees in electronic engineering from Yonsei University in 1992 and 1994, respectively, and the Ph.D. degree in electrical engineering from the Pennsylvania State University in 1999. He is a professor in the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Republic of Korea (ROK). He has been with Yonsei University since 2005. From 1997 to 1999, he served as an assistant professor and instructor in the Department of Electrical Engineering, Pennsylvania State University. From 2000 to 2005, he was with the School of Electrical and Computer Engineering (ECE), Oklahoma State University (OSU), where he served as a tenured associate professor of ECE and director of the OCLNB and ACSEL labs. His research is in the area of MANET, VANET, WSN, satellite & mobile communications, and broadband QoS networking. In 2012 he received the ROK Defense Acquisition Program Administration (DAPA) Director's Award for military technology R&D, in 2008 he received the Outstanding Accomplishment Professor Award from Yonsei University. As an associate professor at OSU, in October 2005 he received the Regents Distinguished Research Award and in September the same year he received the Halliburton Outstanding Young Faculty Award. In 2004 and 2003, respectively, he received the Technology Innovator Award and the Distinguished Faculty Award, both from OSU, and in 2000 he received the First Place Outstanding Paper Award at the IEEE EIT 2000 conference. He is a senior member of the IEEE, member of the IET and IEICE, and life member of the HKN, IEEK, and KICS. He was the General Co-Chair of IEEE MWSCAS 2011. He is also an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and TPC Co-Chair of IEEE VNC 2012.