




Real-time virtual mouse system using RGB-D images and fingertip detection

Dinh-Son Tran¹ · Ngoc-Huynh Ho¹ · Hyung-Jeong Yang¹  · Soo-Hyung Kim¹ · Guee Sang Lee¹

Received: 2 March 2019 / Revised: 3 August 2020 / Accepted: 10 November 2020 /

Published online: 23 November 2020

© The Author(s) 2020

Abstract

A real-time fingertip-gesture-based interface is still challenging for human–computer interactions, due to sensor noise, changing light levels, and the complexity of tracking a fingertip across a variety of subjects. Using fingertip tracking as a virtual mouse is a popular method of interacting with computers without a mouse device. In this work, we propose a novel virtual-mouse method using RGB-D images and fingertip detection. The hand region of interest and the center of the palm are first extracted using in-depth skeleton-joint information images from a Microsoft Kinect Sensor version 2, and then converted into a binary image. Then, the contours of the hands are extracted and described by a border-tracing algorithm. The K-cosine algorithm is used to detect the fingertip location, based on the hand-contour coordinates. Finally, the fingertip location is mapped to RGB images to control the mouse cursor based on a virtual screen. The system tracks fingertips in real-time at 30 FPS on a desktop computer using a single CPU and Kinect V2. The experimental results showed a high accuracy level; the system can work well in real-world environments with a single CPU. This fingertip-gesture-based interface allows humans to easily interact with computers by hand.

Keywords Virtual mouse · RGB-D images · Fingertip detection and tracking · Fingertip-gesture-based interface · Kinect · Human–computer interaction (HCI)

1 Introduction

With the development of augmented-reality technology, researchers are working to reduce people’s workload while increasing their productivity by studying human–computer interactions (HCI). The Natural User Interface (NUI) of hand-gesture recognition is an important topic in HCI. Hand-gesture-based interfaces allow humans to interact with a computer in the most natural way, typically by using fingertip movements.

✉ Hyung-Jeong Yang
hjyang@jnu.ac.kr

Fingertip detection is broadly applied in practical applications, e.g., virtual mice, remote controls, sign-language recognition, or immersive gaming technology. Therefore, virtual mouse control by fingertip detection from images has been one of the main goals of vision-based technology in the last decades, especially with traditional red-green-blue (RGB) cameras [1, 17, 19, 25, 31].

However, even with RGB cameras, most existing algorithms [1, 17, 19, 25, 31] tend to fail when faced with changing light levels, complex backgrounds, multiple people, or background or foreground movements during the hand tracking. Microsoft's Kinect RGB with depth (RGB-D) camera [8] has extended depth-sensing technology and interfaces for human-motion analysis applications [4, 14, 15]. Some systems use depth images from Kinect and achieve high speeds, while avoiding the disadvantages of traditional RGB cameras by tracking depth maps from frame to frame [18, 22, 28]. These methods use a complex mesh model and achieve real-time performance. However, the system only works for hand tracking, not fingertip tracking.

Fingertip detection with multiple people simultaneously poses a great difficulty that current systems have not yet overcome. In addition, choosing a target person when multi-people directly stand to face the camera is a challenging case because it is difficult to determine accurately who will be the target. Therefore, long-term fingertip tracking remains a challenging task. To overcome these disadvantages, a system that is intuitive, affordably priced, easy to use, and allows a user to accurately control a mouse cursor using their fingertips, should be introduced.

In this paper, we propose a gesture-based interface where users interact with a computer using fingertip detection in RGB-D inputs. The hand region of interest and the center of the palm are first extracted from depth images provided by the Kinect V2 skeletal tracker and converted to binary images. Then, the hand contours are extracted and described by a border-tracing algorithm. The K-cosine algorithm is used to detect the fingertip location, based on the hand-contour coordinates. Finally, to control the mouse cursor based on a virtual screen, the fingertip location is mapped to RGB images. Three computer-mouse functions are considered in our research: mouse movement, left-clicking, and right-clicking.

To explore natural gestures with real-time tracking, we investigated complicated cases, e.g., changing the light conditions, background, and distance from the camera during tracking. The proposed system can also detect up to six persons' fingertips simultaneously. Unlike existing methods, this study uses only a single CPU, does not require any special devices or markers, and users are free to move their hands in front of the camera.

The main contributions of the study are as follows:

- The system works with a single low-cost CPU without the help of a graphics processing unit (GPU), has fast detection in real-time (30 frames per second (fps)), and allows execution on computer screens with many types of resolution.
- The system works well with complex backgrounds, low light levels, and long-distance tracking, based on Microsoft Kinect Version 2.
- It provides simultaneous fingertip tracking for up to six people and selects the main person to control the mouse cursor, focusing on the right hand.

The remainder of this paper is organized as follows. Section 2 reviews the related work and Section 3 discusses the proposed method in detail. In Section 4, the performance of our approach is evaluated in comparison with other methods, and finally, the conclusion and future work are presented in Section 5.

2 Related work

Many previous studies on hand-gesture recognition have been conducted using colored gloves [32] or markers [35]. Despite remarkable successes, recognition remains challenging, due to the complexity of using gloves, markers, or variable glove sizes for users. Consequently, many recent efforts have focused on camera-based interfaces.

In recent years, traditional camera-based approaches that detect the area of the hand and recognize hand gestures have been developed [1, 2, 6, 13, 17, 19, 21, 24, 25, 31, 34]. These approaches had obvious detection difficulty when the light levels were changed or a complex background was used and required a fixed distance from the camera to the users. To overcome these limitations, some studies used RGB-D cameras, e.g., PrimeSense, Asus's Xtion Pro, and Microsoft's Kinect [8]. These cameras have advanced significantly over the past few years, with increased performance and lower prices. Compared to traditional RGB cameras, RGB-D cameras offer many advantages: 30 frames per second with depth resolution, working in low light levels, and tracking at a longer distance.

There are many types of RGB-D sensors that can support body tracking such as Kinect V2, VicoVR [20], or Orbbec [7] etc. Among these, Kinect V2 becomes more common nowadays with low-cost and can be applied without CPU. More recently, RGB-D image-based systems using convolutional neural networks (CNN) have shown outstanding performance in HCI [9, 10, 16, 27, 30, 33]. However, these systems require high-performance GPUs for model rendering and a larger dataset for evaluation.

Real-time fingertip detection and tracking can be applied in computer vision using virtual mice [1, 3, 15, 17, 19, 25, 31]. Despite significant improvements in recent years, virtual mouse systems are limited in certain aspects. The approaches in [1, 17, 19, 25, 31] use complex models and achieve real-time performance; however, they are limited by complex backgrounds, low light levels, and the distance from the camera to the hand. In [3], users must wear color pointers for finger tracking, and the mouse control systems are based on the color detection. In addition, selecting a person to perform mouse cursor control is also a big issue that needs to be resolved to eliminate the influence of the others during tracking, but the existing systems have not been mentioned.

The hand-mouse interface in [15] obtains high accuracy using a Kinect sensor; however, the gesture implementation is inconvenient because the user must control the mouse with both hands. Moreover, the work in [15] is limited by the resolution of the virtual monitor. This means that the width and height of the virtual screen depend on the skeleton joints provided by Kinect, e.g., the shoulder width and spine position. The hand-motion area is quite narrow for natural gestures. Additionally, the users must stand to perform the hand gestures.

3 Proposed method

In this section, we shall describe our proposed system. The proposed system consists of six main components, as shown in Fig. 1: (1) hand detection and segmentation; (2) hand-contour extraction; (3) fingertip detection and tracking; (4) target-person locking; (5) virtual screen; and (6) virtual mouse. In this work, we focus on the human's right-hand movement for

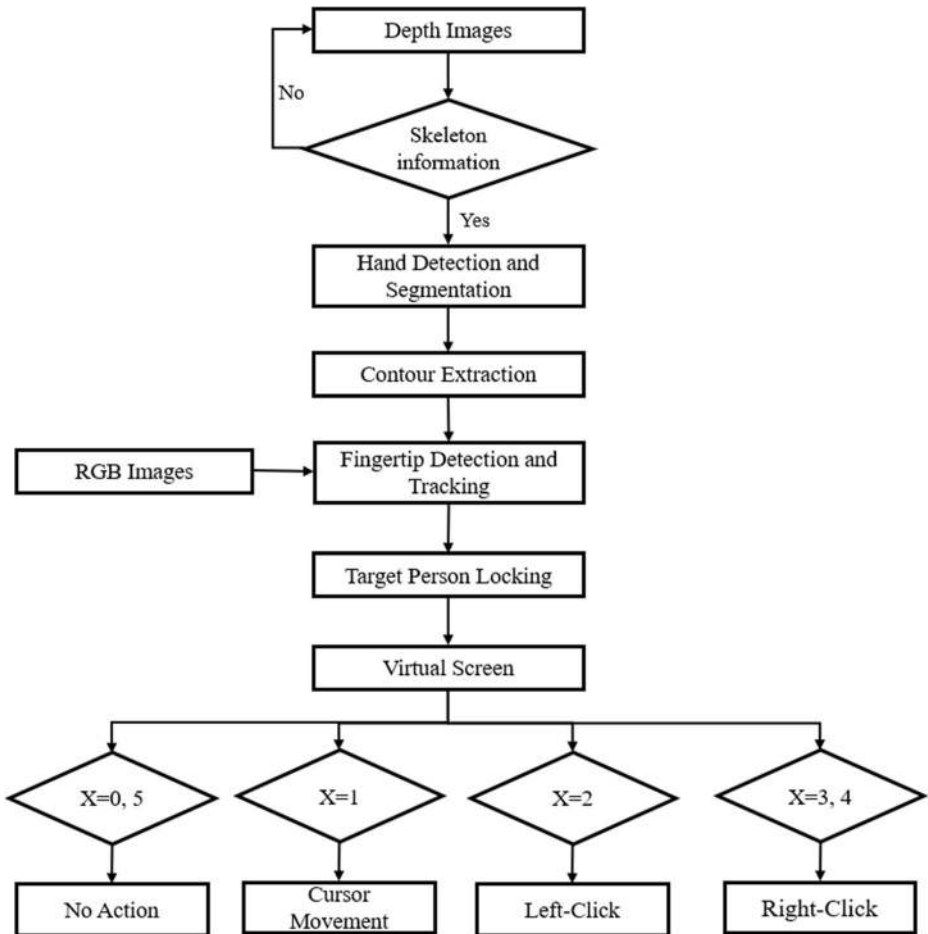


Fig. 1 Flowchart of the proposed method

simplicity and performance accuracy. In Fig. 1, we assume that X is the number of fingertips shown on the right hand.

3.1 Hand detection and segmentation

The depth images used to detect the hand are shown in Fig. 2(a). These images were captured by a Microsoft Kinect V2 sensor, which estimates each of the user's body parts through input depth images, and maps the learned body parts to the depth images through various user actions. In this manner, the camera can obtain the skeleton-joint information for 25 joints, e.g., hip, spine, head, shoulder, hand, foot, and thumb. Using the depth image of a Kinect skeletal tracker, the hand region of interest (HRI) and the center of the palm are easily and effectively extracted.

A median filter and morphological processing [11] were applied to remove noise from the hand region. Afterward, a blob-detection [26] method was used to select the

hand region and export the binary image, based on Kinect's depth signals with fixed thresholds. The results of this process are sets of pixels belonging to the hands, as shown in Fig. 2(b).

3.2 Hand-contour extraction

Hand contours are the curve of the outmost points extracted from the hand-segmentation image. In the fingertip-detection process, contour extraction is a very important step to define the fingertip locations. In this step, the hand contours are detected using the Moore-Neighbor algorithm [23]. This method is one of the most common algorithms used to extract the contours of objects (regions) from an image. After the binary images of the hand regions are detected, the algorithm can find the regional borders by scanning all of the pixels in the images.

At the end of this process, we obtained the contour pixels of the hand as an ordered array. These values are used in the fingertip extraction. The detailed implementation of fingertip detection is presented in the next session. Figure 3 shows an extracted hand contour.

3.3 Fingertip detection and tracking

After extracting the hand contour, the K-cosine Corner Detection [29] algorithm computes the fingertip points using the coordinates of the detected hand contour. This is a well-known algorithm used for detecting the shapes of certain objects and also in fingertip detection. It tries to find the angle between the vectors of a finger, as shown in Fig. 4.

$$|\cos a_i| = \left| \frac{a_i(K)b_i(K)}{|a_i(K)||b_i(K)|} \right| \quad (1)$$

Equation (1) is used to determine the fingertip locations, where the vectors are defined as $a_i(K) = \overrightarrow{P_{(i+k)}P_i}$, and $b_i(K) = \overrightarrow{P_{(i-k)}P_i}$, P_i is a contour point, $P_{(i+k)}$ and $P_{(i-k)}$ are neighbor

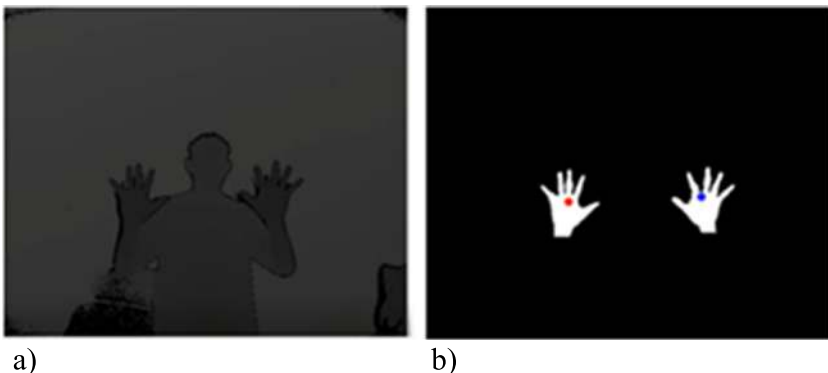


Fig. 2 Hand segmentation using a depth image and skeleton information

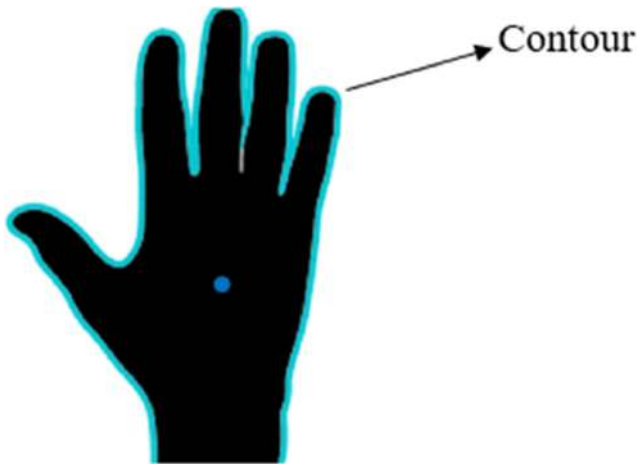


Fig. 3 Contour extraction (blue)

contour points of P_i , a_i denotes the angle between $a_i(K)$ and $b_i(K)$ for a given pixel P_i . The angle a is defined as a threshold value to distinguish the fingertips and the finger valleys. For this paper, the final k is set to 20, and angle a is set to 45 degrees, which are suitable for most situations.

From the cosine values of a_i , obtained by the K-cosine algorithm, if the point value of a_i is smaller or equal to the threshold value, it is defined as the fingertip. The number of fingertips detected is the number of fingers. To real-time fingertip tracking, the detection of frame-by-frame is used for tracking.

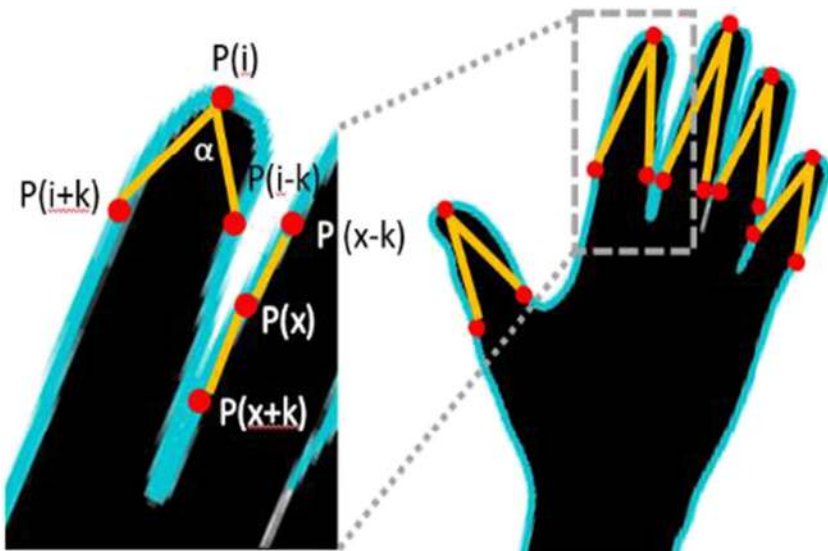


Fig. 4 Fingertip detection using the K-cosine algorithm

3.4 Target-person locking

When multiple people are present, the targeted person is the one chosen to control the mouse during tracking. In this work, the Kinect V2 sensor provides extracted 25 skeletal joints information for up to six people at once such as head, neck, hand-left, hand-right, spine base etc., as shown in Fig. 5. Therefore, the system can locate the fingertips of up to six people, using the algorithm above. However, to control the mouse cursor, we need to identify the target person to eliminate the influence of the others. To do this, we used a user-locking algorithm to solve the problem during hand tracking. The implementation is presented in Algorithm 1.

In this algorithm, given detected head joint coordinates (1) and hand right joints coordinates (13) of multi-people in-depth image using the Kinect skeletal tracker as shown in Fig. 5, we define the target person based on the distance head-hand. If the user raises his hand over the

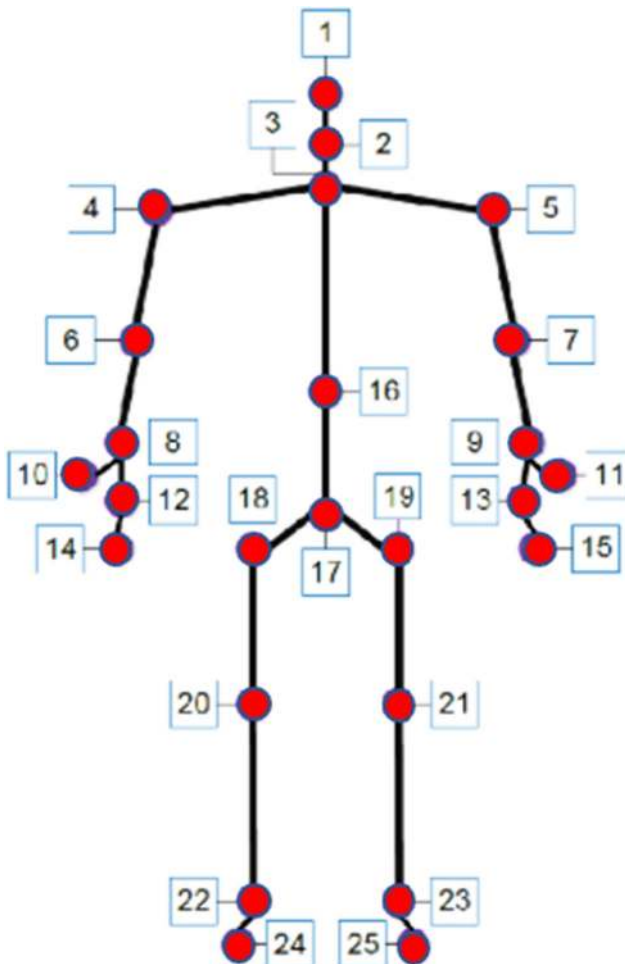


Fig. 5 The skeletal joints information from Kinect v2

head in 10 frames, that is the target user. The selected person is labeled with a yellow box, those who are not selected will be the green box as shown in Fig. 11.

Algorithm 1 Target-Person Locking

```

begin
Require User ID, head joint coordinates (x1,y1), right hand joint
coordinates (x13, y13).
1: Calculation the distance between y1 and y13.
2: If  $y_{13} > y_1$  during 10 frames
    then process target user
    else redo 1
end begin

```

3.5 Virtual screen matching

The virtual monitor concept was first introduced in [2, 15]. It is defined as a virtual space between a Kinect device and a user where a mouse cursor can be controlled by the hands. The advantage of this idea is that it can be implemented on different screen sizes and resolutions. The users only need to watch the virtual screen to control the gestures.

In this step, the resolution of the virtual screen is obtained as 512×424 (X_v , Y_v) pixels, based on the depth resolution of the Kinect V2 sensor. The transformation algorithm is used to transform the fingertip coordinate from the virtual screen to the full screen for controlling the mouse. Figure 6 shows the virtual screen and the real screen. X_r and Y_r are the width and height of the real screen resolution, respectively. X_v and Y_v represent the width and height of the

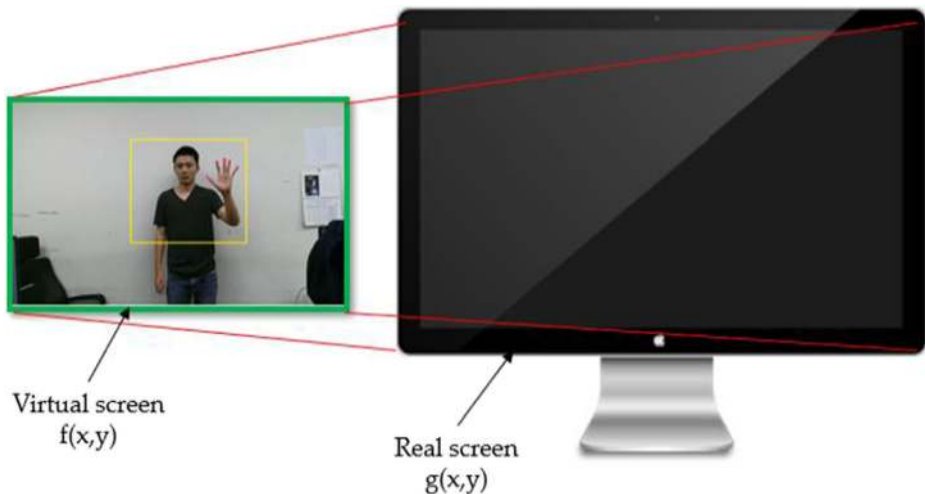


Fig. 6 Virtual screen and real computer screen

virtual monitor, respectively. x and y are the coordinates of the fingertip locations. The transformation algorithm is represented by the following formulae.

$$X_{rate} = X_r/X_v \quad (2)$$

$$Y_{rate} = Y_r/Y_v \quad (3)$$

$$g(x, y) = f(x * X_{rate}, y * Y_{rate}) \quad (4)$$

In equations (2) and (3), X_{rate} and Y_{rate} are the difference in the width and height ratio between virtual monitor and real monitor. After X_{rate} and Y_{rate} are detected, the fingertip coordinates on virtual monitor are multiplied by X_{rate} and Y_{rate} to transform to the real monitor as equation (4).

3.6 Virtual mouse

A computer mouse is a hand-held pointing device that is most often used to manipulate objects on a computer screen. This paper presents a method that allows the user to control the mouse using their fingertip without a mouse device.

In this section, the number of shown fingertips (X) is selected to replace the function of a computer mouse. The goal of this implemented system is to control the mouse cursor using fingertip from a single depth camera. We proposed to use the same four gestures to achieve the mouse control as in [12]. The considered gestures corresponding to the mouse events are mentioned in Fig. 7. There are four types of mouse gesture:

- (1) Cursor movements if $X = 1$,
- (2) Left-click if $X = 2$,
- (3) Right-click if $X = 3$ or 4, and
- (4) No action if $X = 0$ or 5.

The virtual mouse is operated as shown in Fig. 7. We suggested the right-click gesture with three or four fingertips for smooth movements. This is because it is hard to differentiate between three and four fingertips if the gestures are too fast.

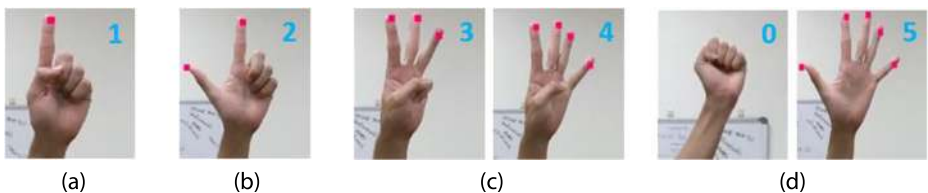


Fig. 7 Virtual-mouse functions based on fingertip counting: **a** mouse movement, **b** left-click, **c** right-click, and **d** no action

4 Experimental results

Virtual-mouse system evaluations in the literature are still somewhat primitive. Since only limited literature and public datasets are available, a cross-method comparison is difficult.

In this section, we shall first discuss the fingertip-detection performance of the virtual mouse. Then, we will present its performance with different lighting, background, and distance-tracking conditions. Next, the experimental results are presented for fingertip tracking with multiple people, and selecting the main person to control the mouse cursor. Finally, we compare our system with previous virtual-mouse studies.

We developed the proposed virtual-mouse system on a desktop PC with an Intel Core i7 4550U 2.10 GHz CPU with 8 GB of RAM. The system was implemented within the C# framework. The tracking process was at the speed of 30 frames per second.

4.1 Virtual-mouse performance analysis

In this experiment, ten subjects made various rapid gestures to evaluate the detection accuracy. The dataset was recorded with various size of monitor resolution to prove that our model is more compatible with real application, instead of using a fit resolution as presented in [1, 17, 19, 25, 31]. There are four computer resolutions as follows: 1280×1024 with 200 cases, followed by 1600×1200 100 cases, 1680×1050 200 cases and finally 1900×1200 100 cases. We assume that X is the number of fingertips shown on the right hand. Each single-person performs gestures with normal light condition. Each gesture from 1 to 5—mouse movement ($X = 1$), left-click ($X = 2$), right-click ($X = 3 \parallel X = 4$), and no action ($X = 5 \parallel X = 0$)—was performed ten times by the ten participants, resulting in 600 gestures, with manually labeled ground truth. All participants were right handed, since we focused on right-hand movement for simplicity and accurate detection. Figure 7, above, shows examples of each gesture for our proposed system.

Table 1 shows the experimental test results of our virtual mouse system. The average accuracy is 96.13%. This is exceptionally high performance for a fingertip gesture-based interface. As expected, the highest accuracy occurred in the easier gesture ‘mouse movement’ and the lowest in the harder gesture ‘right-click’. The accuracy was reduced in the ‘right-click’ gesture because, with fast fingertip tracking, the gesture was sometimes confused with others. The experiment also showed that the results did not change significantly through several resolutions.

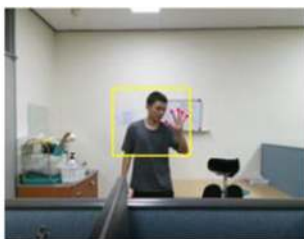
Table 1 Experimental results

<i>Gesture</i>	<i>Fingertip Counting</i>	<i>Success</i>	<i>Failure</i>	<i>Accuracy (%)</i>
Mouse Movement	1	100	0	100
Left-Click	2	97	3	97
Right-Click	3	90	10	89
	4	88	12	
No Action	5	99	1	98.5
	0	98	2	
		572	28	96.13%

4.2 Fingertip tracking in different conditions

The Kinect V2 has been used in different research scenarios as measurement range 0.5–4.5 m, various light conditions, or complex backgrounds. Based on these scenarios, we also confirmed the proposed system's performance in different illumination conditions with a normal light and a faint light condition, complex backgrounds, and long-distance tracking. We conducted a small test to summarize the results as shown in Fig. 8. The 400 other gestures collected also include many different cases, e.g., 50 cases of normal lighting and 50 cases of faint lighting, 100 cases for different backgrounds, and 200 cases for changing the user-camera distance from 0.5 m to 4 m. The experimental results are depicted in Fig. 8.

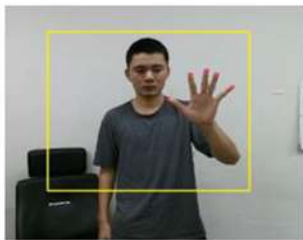
The result shows that there is no significant difference between the normal light and faint-light conditions during the tracking. This means that the system can work well with different light levels. The proposed method also performs well with changing backgrounds and tracking at longer distances. The maximum distance from the camera to the users was 4 m.



a) Normal light



b) Faint light



c) Close distance



d) Long distance (4 meters)



e) Complex backgrounds

Fig. 8 Fingertip tracking under different conditions

4.3 Performance of multiple people tracking

We also conducted fingertip-tracking experiments with varying numbers of people. We investigated five groups with two to six people, selected from the above-mentioned ten people. Each group recorded 100 frames in front of the camera with both hands.

To evaluate the fingertip detection with multiple people, we used a common metric called precision, which is widely used in image segmentation evaluations. Using the notation of true positive (TP), false positive (FP), and false negative (FN), this metric is expressed as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

Generally, this metric compares each of the predicted fingertip detection with each of manually labeled ground truth for a given depth-image input, as shown in Fig. 9. The average accuracy of each group was calculated and is shown in Fig. 10. For the group of two, the fingertip-detection accuracy was highest with 93.25%. The worst case was the group of six with an accuracy of 53.35%. The accuracies of the three, four, and five groups were 89.78%, 78.03%, and 65.38%, respectively. The results show that the accuracy decreases as the number of people in the group increases.

For locking the target person, Fig. 11 depicts the tracking results with real-time three people from RGB-D images. The yellow box is the target person, while the green boxes are tracked objects. The results show that this system can track the fingertips of multi-people in real-time and select the target user to control the virtual mouse while eliminating the influence of the others.

4.4 Comparison with other approaches

We investigated the virtual-mouse literature and summarized the comparison in Table 2. Our experimental results compared to previous approaches using gestures for virtual mice, including the different conditions such as camera type, image type, complex background, tracking distance, stable on different resolutions, target person detection. The details of the gesture comparisons are listed in Table 2.

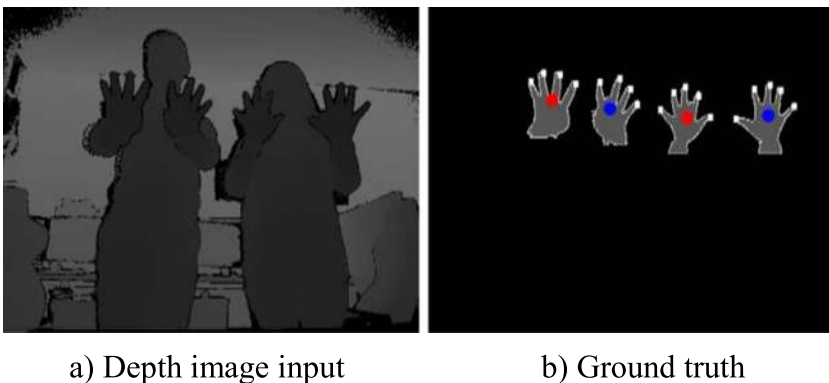


Fig. 9 Fingertips detection with two people

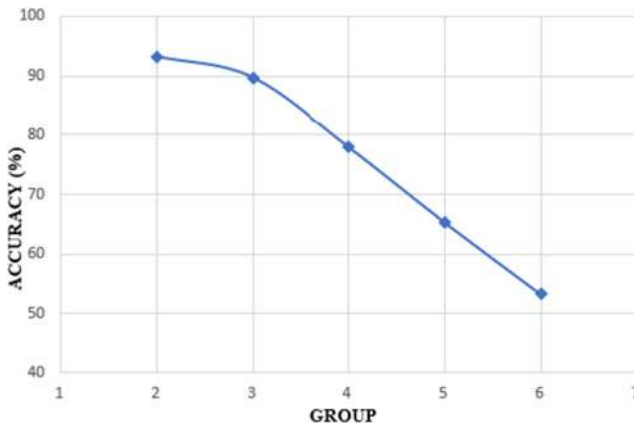


Fig. 10 Fingertip-detection accuracy with a varying number of people

Based on Table 2, it can be seen that the main drawback of [1, 17, 19, 25] is that they use a traditional camera with RGB image. Therefore, their systems only work with an unchanging background and a fixed distance, while our proposed system and [15] can overcome these disadvantages by using the RGB-D Kinect sensor. Besides that, working on a variety of resolutions is also included in these two systems, while the remaining systems only work on a fixed resolution. In particular, the two strengths of our system compared to other systems are being able to track up to 6 people and selecting one person to perform operations with the mouse cursor. This is an important premise for a real-time system in the future.

5 Conclusions

This paper presented a new virtual-mouse method using RGB-D images and fingertip detection. The user's fingertip movement interacted with the computer in front of a camera with no mouse device, gloves, or markers. The approach demonstrated not only highly accurate gesture estimates, but also practical applications.

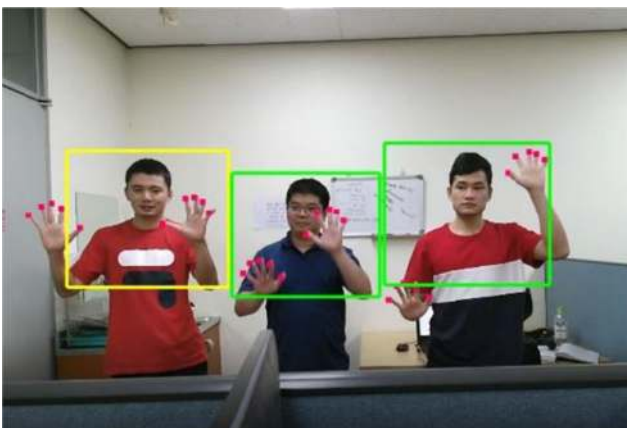


Fig. 11 Fingertip detection with three people

Table 2 Comparison of tracking conditions

	<i>Proposed System</i>	[17]	[19]	[25]	[1]	[15]
Camera type	Kinect V2	Webcam	Webcam	Webcam	Webcam	Kinect
Image type	RGB-D	RGB	RGB	RGB	RGB	RGB
Interaction type	Fingertip	Fingertip	Finger	Fingertip	Fingertip	Hand
Complex background	Yes	No	No	No	No	Yes
Tracking distance	0.5–4 m	Fixed	Fixed	Fixed	Fixed	0.5–4 m
Stable on different resolutions	Yes	No	No	No	No	Yes
# Users for fingertip tracking	6	1	1	1	1	1
Detecting a target person	Yes	No	No	No	No	No

The proposed method overcomes the limitations of most current virtual-mouse systems. It has many advantages, e.g., working well in changing light levels or with complex backgrounds, accurate fingertip tracking at a longer distance, and fingertip tracking of multiple people. The experimental results indicated that this approach is a promising technique for fingertip-gesture-based interfaces in real time.

This study still suffers from several limitations that are mainly inherited from Microsoft Kinect. Therefore, our next work aims to overcome those limitations and improve the fingertip tracking algorithm. We also intend to expand our system to handle more gestures and interact with other smart environments. Finally, it is possible to enrich skeletal tracking by using machine learning algorithms such as OpenPose [5]-based multi-person 2D pose detection, including body, hand, and facial keypoints.

Acknowledgments This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2016-0-00314) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT). (NRF-2020R1A4A1019191).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abhilash S S, Lisho Thomas, NWCC (2018) Virtual Mouse Using Hand Gesture. International Research Journal of Engineering and Technology (IRJET)
2. Bakar MZA, Samad R, Pebrianti D, et al (2015) Finger application using K-curvature method and Kinect sensor in real-time. In: technology management and emerging technologies (ISTMET), 2015 international symposium on. Pp 218–222
3. Banerjee A, Ghosh A, Bharadwaj K, Saikia H (2014) Mouse control using a web camera based on colour detection. arXiv Prepr arXiv14034722
4. Cai Z, Han J, Liu L, Shao L (2017) RGB-D datasets using microsoft kinect or similar sensors: a survey. *Multimed Tools Appl* 76:4313–4355

5. Cao Z, Hidalgo G, Simon T, et al (2018) OpenPose: realtime multi-person 2D pose estimation using part affinity fields. arXiv Prepr arXiv181208008
6. Chen Z, Kim J-T, Liang J, et al (2014) Real-time hand gesture recognition using finger segmentation. *Sci World J* 2014:
7. Coroiu ADCA, Coroiu A (2018) Interchangeability of Kinect and Orbbec sensors for gesture recognition. In: 2018 IEEE 14th international conference on intelligent computer communication and processing (ICCP). Pp 309–315
8. Fossati A, Gall J, Grabner H, et al (2012) Consumer depth cameras for computer vision: research topics and applications. Springer Science & Business Media
9. Ge L, Liang H, Yuan J, Thalmann D (2018) Robust 3D hand pose estimation from single depth images using multi-view CNNs. *IEEE Trans Image Process* 27:4422–4436
10. Ge L, Liang H, Yuan J, Thalmann D (2018) Real-time 3D hand pose estimation with 3D convolutional neural networks. *IEEE Trans Pattern Anal Mach Intell*
11. Gonzalez, RWR (2008) Digital image processing. In: digital image processing, 3rd edn, Upeer Saddle River, NJ: Prentice Hall
12. Grif H-S, Farcas CC (2016) Mouse cursor control system based on hand gesture. *Procedia Technol* 22:657–661
13. Haria A, Subramanian A, Asokkumar N, Poddar S, Nayak JS (2017) Hand gesture recognition for human computer interaction. *Procedia Comput Sci* 115:367–374
14. Ismail NHB, Basah SNB (2015) The applications of Microsoft Kinect for human motion capture and analysis: a review. In: biomedical engineering (ICoBE), 2015 2nd international conference on. Pp 1–4
15. Jeon C, Kwon O-J, Shin D, Shin D (2017) Hand-mouse Interface using virtual monitor concept for natural interaction. *IEEE Access* 5:25181–25188
16. Jiang D, Li G, Sun Y, Kong J, Tao B (2019) Gesture recognition based on skeletonization algorithm and CNN with ASL database. *Multimed Tools Appl* 78:29953–29970
17. Kadam S, Sharma N, Shetty T, Divekar R (2015) Mouse operations using finger tracking. *Int J Comput Appl* 116
18. Khamis S, Taylor J, Shotton J, et al (2015) Learning an efficient model of hand shape variation from depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2540–2548
19. Le PD, Nguyen VH (2014) Remote mouse control using fingertip tracking technique. In: AETA 2013: recent advances in electrical engineering and related sciences. Springer, pp 467–476
20. Ma M, Meyer BJ, Lin L, et al (2018) VicoVR-based wireless daily activity recognition and assessment system for stroke rehabilitation. In: 2018 IEEE international conference on bioinformatics and biomedicine (BIBM). Pp 1117–1121
21. Murugeswari M, Veluchamy S (2014) Hand gesture recognition system for real-time application. In: advanced communication control and computing technologies (ICACCCT), 2014 international conference on. Pp 1220–1225
22. Oikonomidis I, Kyriazis N, Argyros AA (2011) Efficient model-based 3d tracking of hand articulations using kinect. In: *BmVC*. p 3
23. Pradhan R, Kumar S, Agarwal R et al (2010) Contour line tracing algorithm for digital topographic maps. *Int J Image Process* 4:156–163
24. Rautaray SS, Agrawal A (2012) Real time hand gesture recognition system for dynamic applications. *Int J UbiComp* 3:21–31
25. Reza MN, Hossain MS, Ahmad M (2015) Real time mouse cursor control based on bare finger movement using webcam to improve HCI. In: electrical engineering and information communication technology (ICEEICT), 2015 international conference on. Pp 1–5
26. Robotix (2012) Technology robotix society. In: https://2018.robotix.in/https://2018.robotix.in/tutorial/imageprocessing/blob_detection/
27. Sanchez-Riera J, Srinivasan K, Hua K-L, et al (2017) Robust rgb-d hand tracking using deep learning priors. *IEEE Trans Circuits Syst Video Technol*
28. Sharp T, Keskin C, Robertson D, et al (2015) Accurate, robust, and flexible real-time hand tracking. In: proceedings of the 33rd annual ACM conference on human factors in computing systems. Pp 3633–3642
29. Sun T-H (2008) K-cosine corner detection. *JCP* 3:16–22
30. Tang D, Chang HJ, Tejani A, Kim T-K (2017) Latent regression forest: structured estimation of 3d hand poses. *IEEE Trans Pattern Anal Mach Intell* 39:1374–1387
31. Tsai T-H, Huang C-C, Zhang K-L (2015) Embedded virtual mouse system by using hand gesture recognition. In: consumer electronics-Taiwan (ICCE-TW), 2015 IEEE international conference on. Pp 352–353
32. Wang RY, Popović J (2009) Real-time hand-tracking with a color glove. *ACM Trans Graph* 28:63
33. Wang P, Li W, Ogunbona P, Wan J, Escalera S (2018) RGB-D-based human motion recognition with deep learning: a survey. *Comput Vis Image Underst* 171:118–139

34. Xu P (2017) A real-time hand gesture recognition and human-computer interaction system. arXiv Prepr arXiv:1704.07296
35. Zhao W, Chai J, Xu Y-Q (2012) Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data. In: Proceedings of the ACM SIGGRAPH/eurographics symposium on computer animation. pp. 33–42

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dinh-Son Tran He received B.S. degree in Statistics from Ton Duc Thang University, Vietnam, in 2016. He is currently pursuing his M.S. degree in Electronics and Computer Engineering from Chonnam National University. His current research interests are computer vision and machine learning.



Ngoc-Huynh Ho He received B.S. degree from Ho Chi Minh University of Technology. He is currently pursuing his Ph.D degree in Electronics and Computer Engineering from Chonnam National University. His current research interests are computer vision and machine learning.



Hyung-Jeong Yang She received her B.S., M.S. and Ph.D. degrees from Chonbuk National University, Korea. She was a Post-doc researcher at Carnegie Mellon University, USA. She is currently a professor at Dept. of Artificial Intelligence Convergence, Chonnam National University, Gwangju, Korea. Her main research interests include multimedia data mining, pattern recognition, artificial intelligence, e-Learning, and e-Design.



Soo-Hyung Kim He received his B.S. degree in Computer Engineering from Seoul National University in 1986, and his M.S. and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology in 1988 and 1993, respectively. From 1990 to 1996, he was a senior member of research staff in Multimedia Research Center of Samsung Electronics Co., Korea. Since 1997, he has been a professor in the Department of Artificial Intelligence Convergence, Chonnam National University, Korea. His research interests are pattern recognition, document image processing, medical image processing, and ubiquitous computing.



Guee Sang Lee He received his B.S. degree in Electrical Engineering and his M.S. degree in Computer Engineering from Seoul National University, Korea in 1980 and 1982, respectively. He received his Ph.D. degree in Computer Science from Pennsylvania State University in 1991. He is currently a professor of the Artificial Intelligence Convergence in Chonnam National University, Korea. His research interests are mainly in the field of image processing, computer vision and video technology.

Affiliations

Dinh-Son Tran¹ · Ngoc-Huynh Ho¹ · Hyung-Jeong Yang¹ · Soo-Hyung Kim¹ · Guee Sang Lee¹

Dinh-Son Tran
trandinhson3086@gmail.com

Ngoc-Huynh Ho
ngochuynh.03@gmail.com

Soo-Hyung Kim
shkim@jnu.ac.kr

Guee Sang Lee
gslee@jnu.ac.kr

¹ Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju 500-757, South Korea