A Robust Fingertip Detection Algorithm for Mobile User Interfaces

Jong-In Park, Byung-Hun Oh and Kwang-Seok Hong

College of Information and Communication, Sungkyunkwan University,
300 CheonCheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea
{pji17, sincelife}@skku.edu, and ksh@skku.ac.kr

Abstract

A robust algorithm for finger-gesture-based mobile user interfaces (UIs) is presented. The proposed algorithm adopts edge detection for fingertip detection, which is robust to changes in lighting conditions for mobile UIs based on finger gestures. The proposed algorithm has low complexity, and can easily cope with real-time processing using rear-facing cameras in mobile phones. Experimental results demonstrate that the proposed algorithm has better correct-detection probability than the conventional algorithm in various environments.

Keywords: *fingertip detection, mobile user interface, edge detection*

1. Introduction

With the rapid development of devices such as computers and mobile phones, the importance of human-computer interaction (HCI) has been growing to improve the interaction between devices and users. With developments in the processing performance and memory of computers, much faster and much more direct response to input signals has become possible, and various types of input devices and techniques have thus been proposed for more efficient interaction.

Since mobile devices have become an essential part of daily life, mobile users tend to demand more specific and efficient mobile user interfaces (UIs) for interaction. Even though various types of input devices and techniques have been developed, such as keypads, styluses, and touch screens, they are not capable of providing perfect interaction.

Alternative solutions such as acceleration sensors and camera-based interaction are becoming popular with the development of larger screens and richer data on mobile devices. Camera-based interaction methods have especially attracted more and more attention due to its clear superiority [1]. Since mechanical devices such as sensors and gloves are not required, the method is totally contactless, which can provide users with a more natural and unencumbered interaction experience.

There are several types of camera-based UI methods that have been proposed over the years [2-4], employing the face, hands, and fingers for interaction. Most contactless UI methods have involved skin-color-based detection [5]. Recently, a novel UI method using a rear-facing camera was proposed [6]. This method uses the movements of the fingertip, which are detected based on skin color and as an input signal. However, skin-color-based detection has a limitation: the illumination changes frequently.

In this paper, a robust algorithm that uses edge detection for finger-gesture-based mobile UIs is proposed. The proposed algorithm is robust, which makes it possible to achieve better correct-detection probability than the conventional algorithm [6] in various environments, which is demonstrated experimentally.

2. Conventional Algorithm

Figure 1 shows the flow chart of the conventional algorithm, which consists of a fingertip detection part and a fingertip tracking part [6]. In this section, each part of the conventional algorithm is explained in detail.

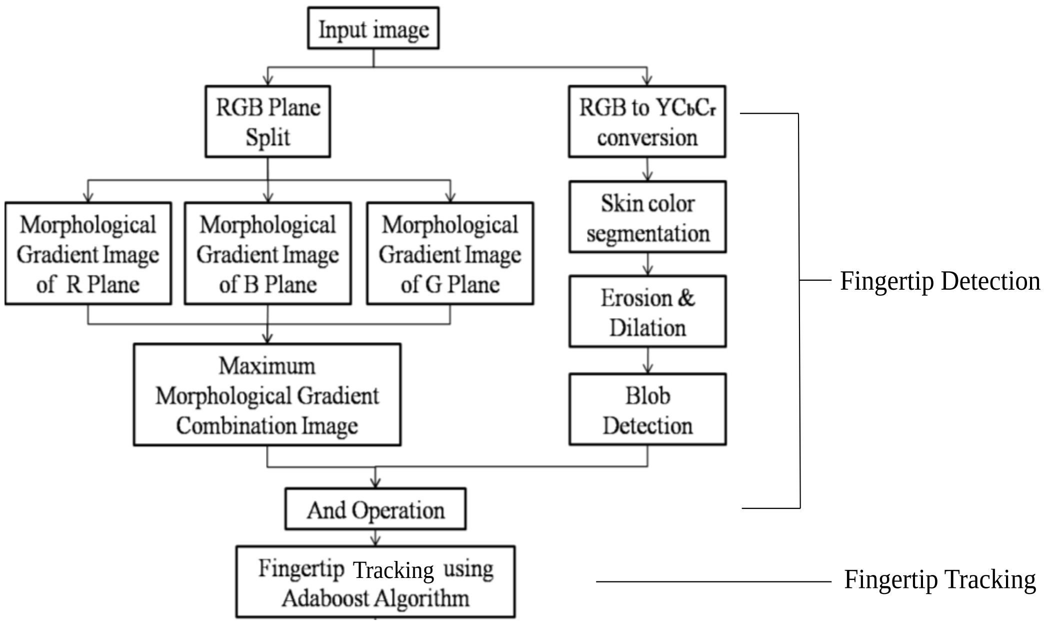


Figure 1. The flow chart of the conventional algorithm

2.1. Fingertip Detection

In the conventional algorithm [6], the YCbCr space is used for skin color segmentation. The component Y reflects the luminance, and the Cb component reflects the difference between the blue component and a reference value, while the Cr component reflects the difference between the red component and a reference value.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.229 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

In the conventional algorithm, the luminance component Y was not considered, because the chrominance components Cb and Cr are almost independent of the luminance component in the space [6]. Therefore, only the Cb and Cr components were used to classify each pixel as skin or non-skin pixels, with the threshold value defined as:

$$\mathrm{SkinColor}(x,y) = \begin{cases} 1 & \mathrm{if}(77 \le Cb \le 127) \cap (133 \le Cr \le 178) \\ & \mathrm{otherwise} \end{cases} \quad (2)$$

Next, the two basic morphological operations, erosion and dilation [7], are applied with 3×3 structuring elements to remove small background objects and separate the finger blob after skin color segmentation. Erosion is used to erode the foreground pixel region boundaries. Thus, the foreground pixel areas shrink, and the holes within these areas become larger.

Even when morphological operations are performed, non-skin-colored objects that are larger than the 3×3 structuring element still exist. To remove larger objects to isolate the finger, blob detection is performed. Each blob is assigned a unique label to separate it from other blobs, and all of the pixels within a blob of spatially connected 1's are assigned the same label. In general, the blob detection is performed with the following two steps: 1) the image is processed from left to right and top to bottom to generate labels for each pixel, while all of the equivalent labels are stored in a pair of arrays, and 2) each label is replaced by a label assigned to its equivalence class. However, the second step has a problem for large images, in that the equivalence arrays can become unacceptably large [8]. To resolve this problem, the conventional algorithm keeps the blob containing the lowest row of the image, and other blobs are eliminated [6].

Next, the conventional algorithm combined the maximum morphological gradient (MMGC) values in the split R, G, and B planes into a single image that has clearer gradient values than a gray image. The MMGC is defined as:

$$\mathrm{MMGC} = \sum_{j}^{\mathrm{height}} \sum_{i}^{\mathrm{width}} \max\left(\mathrm{MG_R}(i,j), \mathrm{MG_G}(i,j), \mathrm{MG_B}(i,j)\right) \tag{3}$$

where i and j denote the index of the horizontal pixel and the vertical pixel, respectively, and $\mathrm{MG_R}(i,j), \mathrm{MG_G}(i,j), \text{and}\, \mathrm{MG_B}(i,j)$ denote the result of the morphological gradient at (i, j) pixel in the R, G, and B planes, respectively.

Finally, the conventional algorithm can detect a finger by using an AND operation between the result image of the MMGC and the result image of skin color segmentation and blob detection.

2.2. Fingertip Tracking

In the conventional algorithm, the combination of Haar-like features and boosted classifiers using the AdaBoost machine-learning algorithm is used. For a fingertip detection system, half-circle area images of the fingertip from the results of pre-processing are collected as positive samples. The samples were collected under various illumination conditions, and the numbers of the collected positive samples and negative samples from the results of pre-processing for the training process are 2240 and 4500, respectively. The fingertip cascade classifier is a 13-stage cascade that is 20×10 pixels in size.

3. Proposed Algorithm

Figure 2 shows a flow chart of the proposed algorithm, which consists of a fingertip detection part and a fingertip tracking part. In this section, each part of the proposed algorithm is explained in detail.

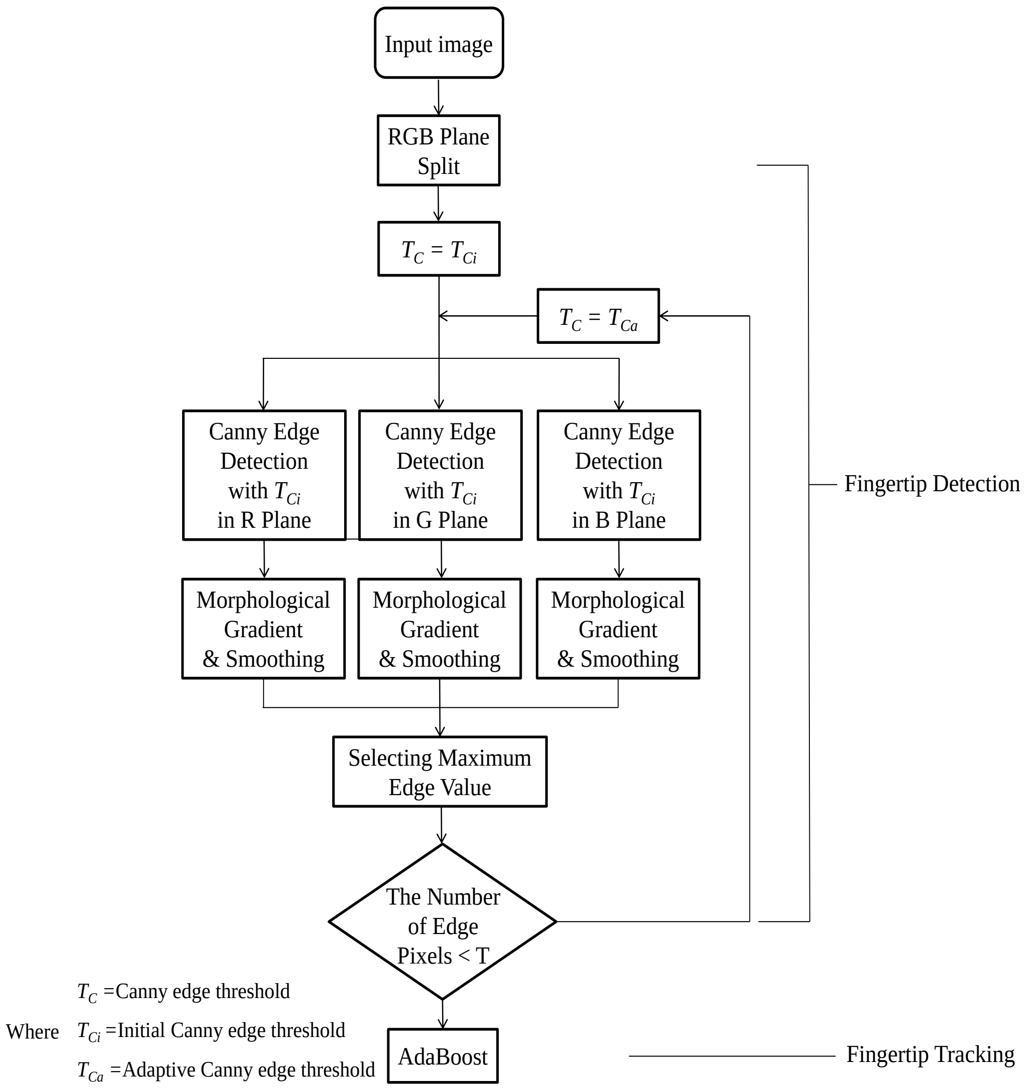


Figure 2. The flow chart of the proposed algorithm

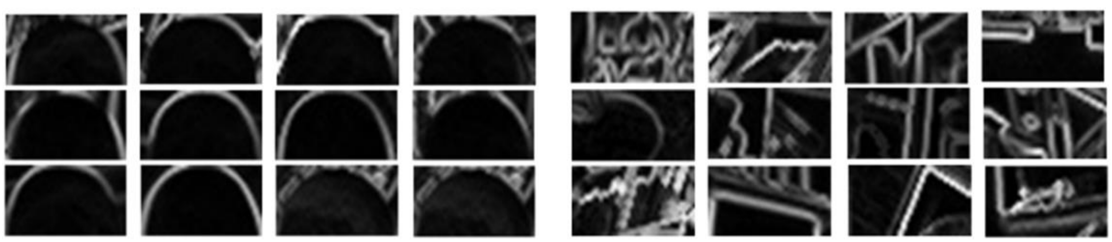
3.1 Fingertip Detection

In the proposed algorithm, edge detection is used to overcome the weakness of the conventional algorithm in various lighting conditions, and the Canny edge detection method is used as the fingertip detector.

First, an input image splits into R, G, and B planes to acquire clearer edges than those obtainable using a gray scale image. Next, edges are detected in each plane by Canny edge detection with initial Canny edge thresholds $T_C = T_{Ci} = (100,200)$, and then detected edge regions are expanded in each plane by morphological gradient and smoothing. The pixel values of edge regions in each plane are compared, and the pixels which have the maximum pixel value among the three planes are selected as an edge. Then, the number of edge pixels is compared with the pixel number threshold T (=Ten percent of the number of whole pixels in an image). If the number of the edge pixels is less than T, the fingertip detection is complete. If not, T_C is reset by the adaptive Canny edge threshold $T_{Ca}(\in\{(100,250),(150,200),(150,250),(200,200),(200,250),(250,250)\})$ and the Canny edge detection is repeated until the condition is satisfied. If the ratio of the edge pixels in an image is too large, the performance of the fingertip tracking can be degraded by the edges of the background. For this reason, in the proposed algorithm, T is set so that the ratio of the edge pixels in an image is kept less than T, and it is possible to use the property that the number of edge pixels decreases as T_{Ca} increases.

3.2 Fingertip Tracking

The combination of Haar-like features and boosted classifiers using the AdaBoost machine-learning algorithm is widely used to train parts of the body, such as the head, hands, and fingers, due to its good performance. For this reason, we also adopted the combination of Haar-like features and the AdaBoost algorithm for fingertip tracking, just like in the conventional algorithm. The samples for fingertip tracking were collected under various illumination conditions, and the numbers of the collected positive samples and negative samples from the results of pre-processing for the training process are 2240 and 4500, respectively. The fingertip cascade classifier is a 13-stage cascade that is 20×10 pixels in size. Figure 3 shows some positive and negative samples used in the proposed algorithm.



(a) Positive samples **(b) Negative samples**

Figure 3. Some positive and negative samples

4. Experimental Results

In this section, the proposed algorithm is compared with the conventional algorithm in terms of fingertip detection performance and the mean processing time in various environments.

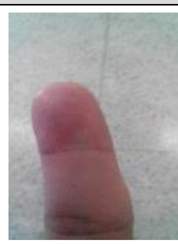
In the experiments, four conditions were considered, as shown in Table 1, and about 300 images of each condition (10 frames/sec) which has the size of 120×160 pixels were used.

Table 1. Four conditions used in the experiment

Condition 1	Bright lighting condition I
Condition 2	Bright lighting condition II
Condition 3	Dark lighting condition
Condition 4	Sunset lighting condition

Figure 4 shows the step results of the proposed algorithm in four conditions, and the results of fingertip detection performance of the proposed and conventional algorithms are shown in Tables 2 and 3. The proposed algorithm has better correct detection performance than the conventional algorithm, except for in condition 2. Even though the proposed algorithm shows worse correct-detection performance than the conventional algorithm in condition 2, the difference is very small. These results are caused by the property associated with using the rear-facing camera, in that since the ratio of the finger is high in an image, the ratio of the edge pixels is low. This makes the edge-detection-based proposed algorithm capable of achieving better correct detection probability than the skin-color-based conventional algorithm.

Since the proposed algorithm has a repetition part for edge detection, the mean processing time of the proposed algorithm is longer than that of the conventional algorithm by about 3.5 % to 216.0 %. However, in the paper [9], experiments about the intelligibility of isolated hands signs were conducted at varying frame rates. They found that there are no significant differences from 30 to 15 fps, but a slight decrease in intelligibility from 15 to 10 fps and a large decrease in intelligibility from 10 fps to 5 fps exist. Consequently, the proposed algorithm is sufficiently capable of dealing with real-time applications considering the minimum frame rate of about 17 fps as the results.

	Condition 1	Condition 2	Condition 3	Condition 4
Input image				
Morphological gradient & smoothing (R plane)				
Morphological gradient & smoothing (G plane)				

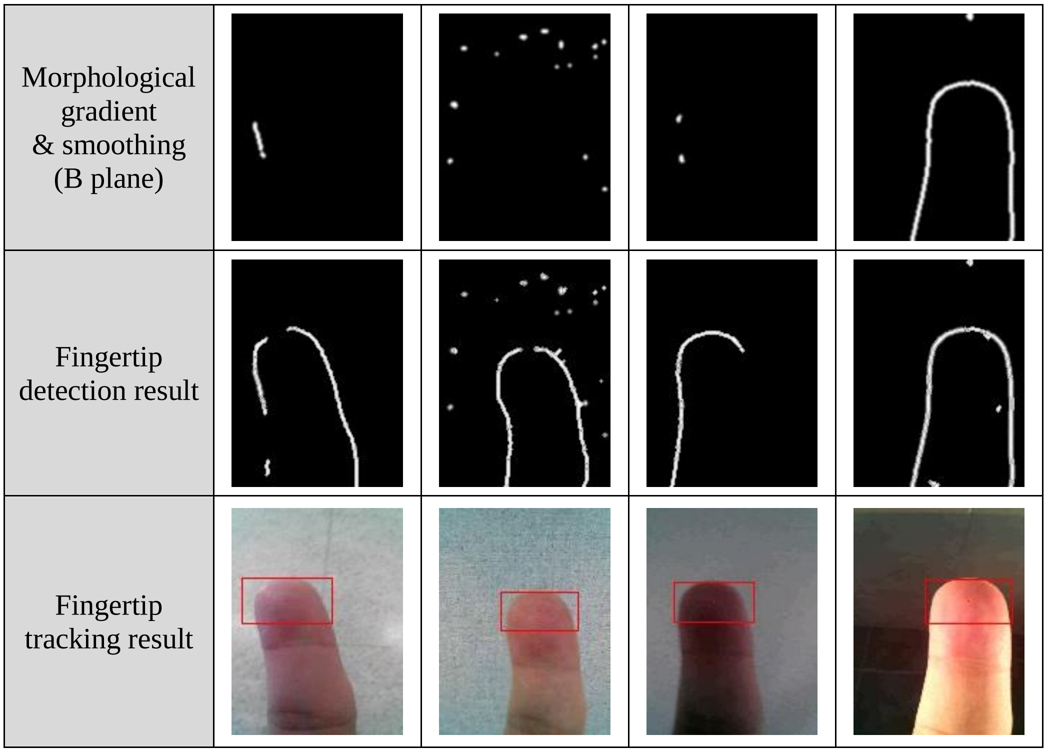


Figure 4. The step results of the proposed algorithm in four conditions

Table 2. The fingertip detection performance of the proposed algorithm

	Condition 1	Condition 2	Condition 3	Condition 4
The number of detected frames	245	260	258	291
The number of false detected frames	0	0	0	0
The number of miss detected frames	57	42	42	11
The number of total frames	302	302	300	302
Correct detection probability	81.1 %	86.1 %	86.0 %	96.4 %
False detection probability	0 %	0 %	0 %	0 %
Mean processing time	18.8 ms	17.6 ms	18.1 ms	57.2 ms

Table 3. The fingertip detection performance of the conventional algorithm

	Condition 1	Condition 2	Condition 3	Condition 4
The number of detected frames	4	262	158	275
The number of false detected frames	1	0	2	0
The number of miss detected frames	298	40	142	27
The number of total frames	302	302	300	302
Correct detection probability	1.0 %	86.8 %	52.0 %	91.1 %
False detection probability	0.3 %	0 %	0.7 %	0 %
Mean processing time	17.0 ms	17.0 ms	16.8 ms	18.1 ms

5. Conclusion

We have proposed a robust finger gesture detection algorithm that uses rear-facing cameras for mobile UIs. Using edge detection, the proposed algorithm is robust against change in lighting conditions. In addition, the proposed algorithm can be applied to various real-time applications, as it also has low complexity. It has been confirmed experimentally that the fingertip detection performance of the proposed algorithm is better than that of the conventional algorithm.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2010-0021411), and by MKE, Korea under ITRC NIPA-2012-(H0301-12-3001).

References

[1] H. He, X. Duan and Y. Wu, Int. J. Virtual Reality, vol. 9, no. 4, **(2010)**.

[2] S. Prasad, A. Sawant, R. Shettigar, K. Bhandari and S. Sinha, “Skin segmentation based face tracking independent of lighting conditions”, Int. Conf. and Workshop on Emerging Trends in Technology, **(2011)** February 25-26; Mumbai, India.

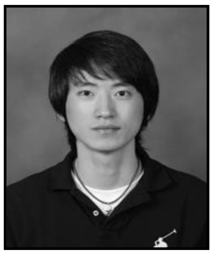
[3] J. L. Raheja, K. Das and A. Chaudhary, “An efficient real time method of fingertip detection”, Int. Conf. Trends in Industrial Measurements and Automation **(2011)** January 6-8; Chennai, India.

[4] D. -Y. Huang, W. -C. Hu and S. -H. Chang, “Expert Systems with Applications”, vol. 38, **(2011)**.

[5] C. Yang, Y. Jang, J. Beh, D. Han and H. Ko, “Gesture recognition using depth-based hand tracking for contactless controller application”, IEEE Int. Conf. Consumer Electronics **(2012)** January 13-16; Las Vegas, NV.

[6] J. -H. An, "Finger gesture based mobile user interface using a rear-facing camera", M. S. E. dissertation, Dept. Mobile Systems Engineering, University of Sungkyunkwan, Suwon, **(2011)**.
[7] L. G. Shapiro and G. C. Stockman, "Computer Vision", Prentice Hall, **(2002)**.
[8] R. Lumia, L. Shapiro and O. Zuniga, Computer Vision, Graphics, and Image Processing, vol. 22, no. 2, **(1983)**.
[9] G. Sperling, M. Landy, Y. Cohen and M. Pavel, Computer Vision, Graphics, and Image Processing, vol. 31, no. 3, **(1985)**.

Authors



Jong-In Park received his B.S. degree in electronic engineering from the Kwangwoon University, Seoul, Korea, in 2011. He is presently an M.S candidate at the College of Information and Communication Engineering at Sungkyunkwan University, Suwon, Korea. His current research focuses on digital image processing and pattern recognition.

Byung-Hun Oh received his B.S. degree in electronic engineering from the Eulji University, Seongnam, Korea, in 2011, M.S. degree in at the College of Information and Communication Engineering at Sungkyunkwan University, Suwon, Korea, respectively. He is presently a Ph.D. candidate at the College of Information and Communication Engineering at Sungkyunkwan University, Suwon, Korea. His current research focuses on digital image processing and pattern recognition.

Kwang-Seok Hong received his B.S., M.S., and Ph.D. in Electronic Engineering from Sungkyunkwan University, Seoul, Korea in 1985, 1988, and 1992, respectively. Since March 1995, he has been a professor at Sungkyunkwan University, Suwon, Korea. His current research focuses on human-computer interaction, five-sense recognition, interaction, and representation.