A Novel Fingertip Detection Algorithm for Mobile User Interfaces

Jong-In Park [1], Byung-Hun Oh[1], and Kwang-Seok Hong[1],

[1] 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 novel 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

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. There are several types of camera-based UI methods that have been proposed over the years [1-3], employing the face, hands, and fingers for interaction. Most contactless UI methods have involved skin-color-based detection [4]. Recently, a novel UI method using a rear-facing camera was proposed [5]. 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 novel 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 [5] in various environments, which is demonstrated experimentally.

ISA 2013, ASTL Vol. 21, pp. 164 - 167, 2013

2 Proposed Algorithm

Fig. 1 shows a flow chart of the proposed algorithm, which consists of a fingertip detection part and a fingertip tracking part.

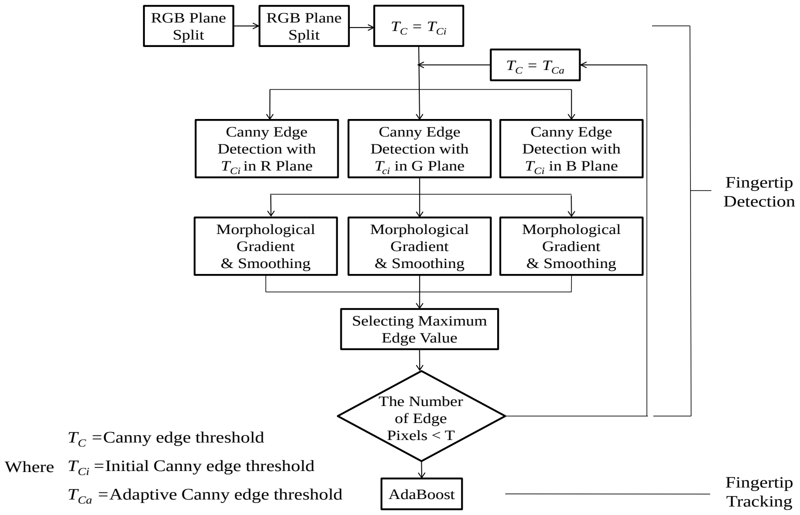

Fig. 1. The flow chart of the proposed algorithm

2.1 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.

2.2 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.

3 Experimental Results

In the experiments, four conditions were considered: Bright lighting condition I, II, dark lighting condition, and sunset lighting condition, and about 300 images of each condition (10 frames/sec) which has the size of 120×160 pixels were used.

The results of fingertip detection performance of the proposed and conventional algorithms are shown in Tables 1. 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 [6], 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.

4 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.

Table 1. The fingertip detection performance of the proposed and the conventional algorithm (proposed algorithm / conventional algorithm)

	Condition 1	Condition 2	Condition 3	Condition 4
The number of detected frames	245 / 4	260 / 262	258 / 158	291 / 275
The number of false detected frames	0 / 1	0 / 0	0 / 2	0 / 0
The number of miss detected frames	57 / 298	42 / 40	42 / 142	11 / 27
The number of total frames	302	302	300	302
Correct detection Probability	81.1 / 1.0 (%)	86.1 / 86.8 (%)	86.0 / 52.0 (%)	96.4 / 91.1 (%)
False detection Probability	0 / 0.3 (%)	0 / 0 (%)	0 / 0.7 (%)	0 / 0 (%)
Mean processing time	18.8 / 17.0 (ms)	17.6 / 17.0 (ms)	18.1 / 16.8 (ms)	57.2 / 18.1 (ms)

Acknowledgments. 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. S. Prasad, A. Sawant, R. Shettigar, K. Bhandari, and S. Sinha.: Skin segmentation bsed face tracking independent of lighting conditions. In: Int. Conf. and Workshop on Emerging Trends in Technology (ICWET), pp. 123--126, Mumbai (2011)
2. J. L. Raheja., K. Das, A. Chaudhary.: An efficient real time method of fingertip detection. In: TIMA 2011, pp. 447--450, Chennai (2011)
3. D.-Y. Huang, W.-C. Hu, and S.-H. Chang.: Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. Expert Systems with Applications, vol. 38, pp. 6031--6042 (2011)
4. C. Yang, Y. Jang, J. Beh, D. Han, and H. Ko.: Gesture recognition using depth-based hand tracking for contactless controller application. In: IEEE Int. Conf. Consumer Electronics (ICCE), pp. 297--298, Las Vegas (2012)
5. 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)
6. G. Sperling, M. Landy, Y. Cohen, and M. Pavel.: Intelligible encoding of ASL image sequences at extremely low information rates. Computer Vision, Graphics, and Image Processing, vol. 31, no. 3, pp. 335-391 (1985)