An image sensor based virtual mouse including fingertip detection in face mask algorithm

Chung-Wen Hung, Hsuan T. Chang, and Cheng-Yang Chen

National Yunlin University of Science and Technology, Yunlin 640-02, Taiwan
(Tel: 886-5-534-2601)

wenhung@yuntech.edu.tw

Abstract: An image sensor based virtual mouse with fingertip detection in face mark (FDIFM) function is proposed in this paper. As the image sensors or webcams are more and more cheap and popular, the algorithms are developed to detect the fingertip location and its motions. The motion information could be used to define the mouse functions and perform a virtual mouse. When the fingertip locates in the face area, it is difficult to detected, due to similar skin colors. The FDIFM algorithm is developed to handle this exceptional situation, using the red component image of the face area, the fingertip could be detected successfully. In this paper, the algorithms and simulation results are described, and the experimental results are also presented to verify the virtual mouse workable.

Keywords: virtual mouse, image sensor, webcam, fingertip detection in face mask (FDIFM).

1 INTRODUCTION

Improving the efficiency of human computer interface (HCI) devices, such as keyboard and mouse, has been an important issue for a long time. Due to the development of new technologies and cost down, there are a lot of new HCI devices and methods are proposed, such as touch panel and its user friendly control. However, with the price reduction and popularity of a digital image capture sensor, another HCI technology which is based on image processing and detection made progress, and the Microsoft's Kinect is a good example[1], but the function focuses on TV games. Chen [2] proposed an interface method which is based on hand-gesture and facial expression, and Qzawa [3] utilized the data glove to control a robotic hand. These researches tacked a lot of CPU resource or needed some extra equipment.

A webcam based virtual mouse is proposed in this paper. As the price reduction of CMOS image sensors and the popularity of social communication software, the webcam became a regular peripheral. Based on such a low cost webcam, the simplified algorithms are development to perform the functions of a traditional mouse in this research. First, every single image frame is captured and processed with several operations, which include the background removal, the skin color detection, the background updating, and the fingertip detection. Then the fingertips are tracked, and the traveling direction and distance are measured. Furthermore, when the fingertip overlaps the user's face region, a special algorithm is proposed to detect. Finally, the specific fingertip motions are defined as to the corresponding functions of the virtual mouse. The algorithm will be discussed as following, and the experimental results also will be presented.

2 The algorithm

The algorithm includes several procedures: background removal, skin detection, hand detection, connected regions labeling, hand region extraction, fingertip detection, fingertip motion tracking, and fingertip extracting from overlapping facial region. The details and operations are described as following:

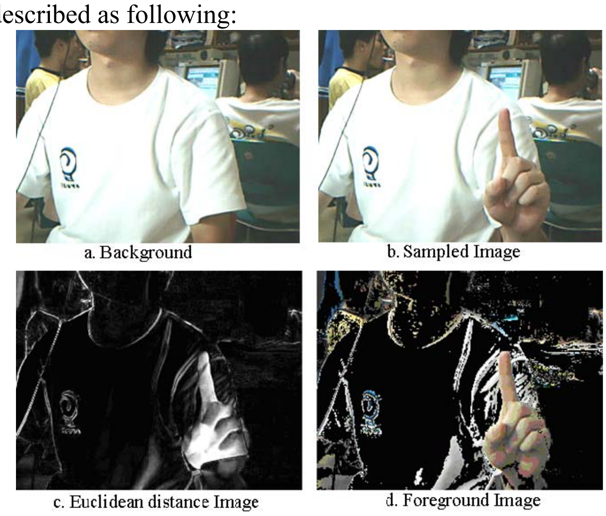


Fig. 1 The background, sampled and background-removed Images

2.1 Background Removal

First, take an image in advance as background, as shown in Fig. 1.a; then calculate the RGB Euclidean distance between sampled image, shown in Fig. 1.b, and background for every pixels as (1), Y. Cui [4]:

$$A = \sqrt{(I_{CR}(x,y) - I_{BR}(x,y))^2 + (I_{CG}(x,y) - I_{BG}(x,y))^2 + (I_{CB}(x,y) - I_{BB}(x,y))^2} \quad (1)$$

Here, I is the brightness, C and B show current sampled image and background, respectively, and R, G and B indicate red, green and blue, respectively. If the Euclidean distance of a pixel is greater than threshold, it means that this pixel is not similar to the background and will be retained. Finally, the background is removed, and the foreground is obtained, as shown in Fig. 1.c.

2.2 Skin Detection

Based on the database proposed by Maricor Sorian [5], [6], the skin detection processing is used to detect the skin region on the previous foreground image. The normalization values of red and green are determined from (2) and (3), respectively.

$$r = \frac{R}{R+G+B} \quad (2)$$

$$g = \frac{G}{R+G+B} \quad (3)$$

If the r and g values of a pixel are located in the skin region as shown in Fig. 2, in which the region is bound in a pair of 2nd order functions as (4)~(7), the pixel will be treated as the skin.

$$f_1(r) = -1.3767r^2 + 1.0743r + 0.1452 \quad (4)$$

$$f_2(r) = -0.776r^2 + 0.5601r + 0.1766 \quad (5)$$

$$W = (r - 0.33)^2 + (g - 0.33)^2 \quad (6)$$

$$s = \begin{cases} 1, & \text{if}\,(g < f_1(r))\,\&\,(g > f_2(r))\,\&\,(r > 0.2) \\ & \&\,(r < 0.6)\,\&\,(W > 0.0004) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The S=1 marks the pixel as the skin. The result is shown in Fig. 3.

Fig. 3 The binary image of skin detection.

2.3 Hand Detection

The morphological erosion and dilation operators are used to detect the hand region in the above binary image, H. J. Lee [7]. The erosion is utilized to eliminate the small noise area, and then the dilation operation is applied to reconstruct the hand region which is shrunken and broken by the previous erosion operator. The result image of erosion and dilation operators is shown in Fig. 4, and the hand region is successfully detected with some noise blocks.

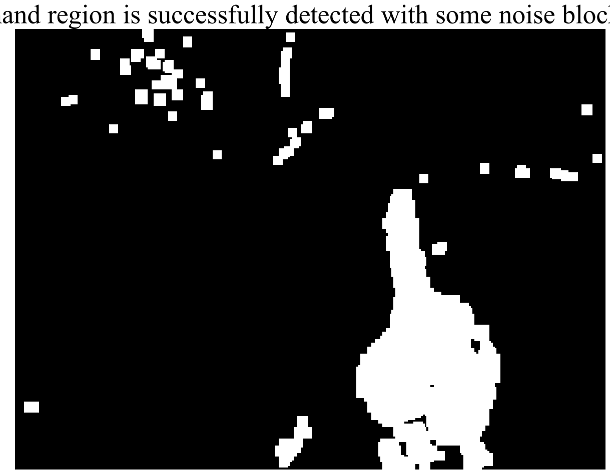


Fig. 4 The result image of hand detection.

2.4 Connected regions labeling

The result image of erosion and dilation operators would contain only few bigger skin regions. They will be labeled for every connected region for the calculation of the region size.

2.5 Hand Region Extraction

For all the connected skin labeled region, their sizes of regions are calculated. The maximum is treated as the hand region, because the user's hand is supposed to be closest to the webcam, and the other region is supposed to be noise or the region closed to the skin color. The extraction result is shown in Fig. 5, it proves that the extraction algorithm is workable.

Fig. 5 The extracted hand region.

2.6 Fingertip Detection

When the hand region is extracted, the edge detection is next processed. The Prewitt algorithm is employed to detect the edge boundary, and then the contour will be more obvious, D. Wang [8]. Next, the contour search criteria are proposed, H. Soltanian-Zadeh [9]: the lowest and leftest point of the hand region edge is the starting point; and the left-to-right and bottom-to-top search criteria are used to find the next connected pixel of edge; then the searching is repeated until back to the starting point.

When the contour of a hand is detected, the pixels are compared one-by-one to find the top point. Because the default command finger is fixed to the forefinger, and this point is treated as a candidate fingertip point. However, some other gestures would lead to a wrong detection.

To confirm this point is the fingertip point and filter out wrong hand gestures, a checking processing is necessary. Fist, based on this point, the next 10-points pair and next 15-points pair are located on the contour by forward and backward searching; then, the angles of a candidate fingertip point with the next 10-points pair and next 15-points are calculated. Based on the characteristic of a forefinger, the former angle should be bigger than latter's. Otherwise, the image will be treated as an invalid image. The fingertip detection result is shown in Fig. 6.

2.7 Fingertip Motion Tracking

If a fingertip is marked, the fingertip is tracked and the traveling direction and distance are measured. Then the direction and distance will be used to define the mouse action, like a left-key click. The Lucas-Kanade optical flow method is used to track the motions, which will be applied to measure the direction and distance, S. N. Tamgade [10].

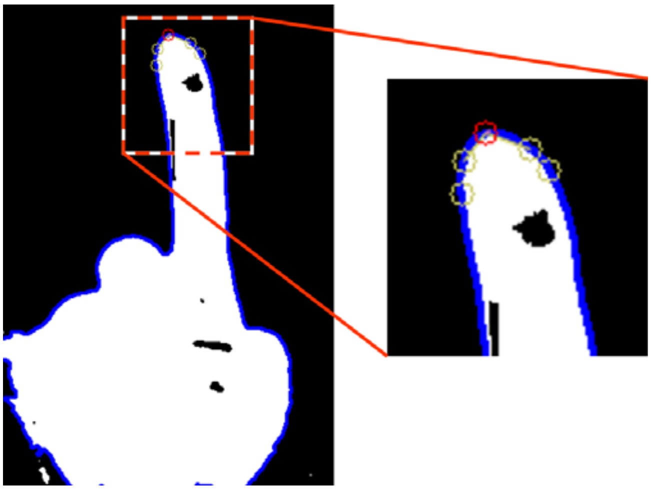


Fig. 6 The fingertip detection result.

2.8 Fingertip Motion Tracking

In normal cases, the mentioned procedures could successfully detect the fingertip and track the motions. However, some errors happen when the finger locates in the face region, because the face and finger have similar colors and the gesture region will be lost in above methods. The FDIFM algorithm is proposed to handle such situation.

First, the face mask sensing area is defined, which is based on a rectangular face mask with extended outward 50 pixels. It is used to decide that processing should fast switch to the FDIFM procedure. When the fingertip moves into the mask area, the mask area is extracted, names region of interest (ROI). Next, the negative image of the ROI is calculated, as shown in Fig. 7.a. Based on the red component of ROI negative image, the fingertip will be found, due to it is a obvious dark region in face area as shown in Fig. 7.b. Finally, the grayscale image is transformed into a binary image, and the fingertip will be detected more easily, as shown in Fig. 7.c.

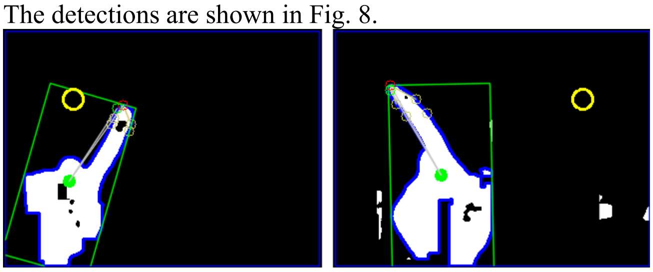
a. Negative image　b. Red component image　c. Binary image

Fig. 7 The processed images in FDIFM algorithm.

2.8 The function definition of the virtual mouse

When the fingertip and its motion are tracking successfully, the functions of the virtual mouse could be defined by the motion of fingertip, such as the double click of fingertip indicates a click action of a mouse, and the angles of fingertips could be used to define other functions of a mouse.

To differentiate between the vertical movement and the click action, the mouse click should be defined as double click of the fingertip. If the locations of fingertip in several consecutive images show fingertip motion is in "up-down-up-down" moving, the mouse click message will be send to the PC system. Similarly, the angle of the line from the center of the minimal rectangular to the fingertip could be measurement, and then the clockwise and counterclockwise angles could be used defined as other function of a mouse. The detections are shown in Fig. 8.



a.The fingertip move clockwise　b. The fingertip moves counterclockwise

Fig. 8 The images of clockwise and counterclockwise motions detection.

3 Experimental Results

The above algorithms are developed in desktop or notebook personal computers. So the experiments are implemented on following specifications: the hardware specification is Intel Core 2 Quad 2.4GHz CPU with 3 GB DDR2 DRAM; software is programmed under Microsoft Visual Studio 2008 and program runs in Windows 7 professional 32bit OS; the image is captured by Logitech QuickCam which is set to 24bit RGB colors, 640×480 resolution, and 12 frames per second. The program occupies about 28% of CPU resource.

The fingertip detection is the base of the proposed algorithm, and the experimental result is shown in Table. 1, the average success rate is about 94.4%.

Table. 1 The correct fingertip detection rate

	Total frame	Correct fingertip frame	Success rate
User 1	600	575	95.8%
User 2	600	574	95.6%
User 3	600	540	90.0%
User 4	600	565	94.2%
User 5	600	567	94.5%
User 6	600	564	94.0%
User 7	600	573	95.5%
User 8	600	554	92.3%
User 9	600	570	95.0%
User 10	600	580	96.7%
Average			94.4%

The experimental results of the left-key click function for a virtual mouse are shown in Table. 2. Here, the rates of the User 1 and 2 are lower, which should be caused from too fast clicks. The average successful rate is about 83.5%. The other two functional experiments for clockwise- and counterclockwise-motion detections are respectively listed in Tables 3 and 4, in which the successful rates are about 95% and 100%, respectively. Due to distinct shapes, the detection rates are higher than other functions.

Table. 2 The successful click detection ratio

	Total frame	Correct fingertip frame	Success rate
User 1	20	15	75%
User 2	20	13	65%
User 3	20	18	90%
User 4	20	18	90%
User 5	20	16	80%
User 6	20	16	80%
User 7	20	18	90%
User 8	20	17	85%
User 9	20	18	90%
User 10	20	18	90%
Average			83.5%

Table. 3 The successful ratio of the clockwise motion detection

	Total frame	Correct fingertip frame	Success rate
User 1	20	19	95%
User 2	20	18	90%
User 3	20	19	95%
User 4	20	18	90%
User 5	20	18	90%
User 6	20	19	95%
User 7	20	20	100%
User 8	20	20	100%
User 9	20	19	95%
User 10	20	20	100%
Average			95%

Table. 4 The successful ratio of the counterclockwise motion detection

	Total frame	Correct fingertip frame	Success rate
User 1	20	20	100%
User 2	20	20	100%
User 3	20	20	100%
User 4	20	20	100%
User 5	20	20	100%
User 6	20	20	100%
User 7	20	20	100%
User 8	20	20	100%
User 9	20	20	100%
User 10	20	20	100%
Average			100%

4 CONCLUSION

An image sensor based virtual mouse including the FDIFM algorithm is proposed in this paper. Based on popular image sensors or webcams, the proposed algorithms could efficiently detect the fingertip locations and motions, which information could be used to replace the mouse and act as the virtual mouse. The FDIFM algorithm is also developed to handle the exceptional situation that the fingertip locates in the facial region. From the experimental results, the virtual mouse is workable, and the algorithms could be extended to other new HCI devices.

Acknowledgments: This work is partly supported by the National Science Council, Taiwan, under contract NSC 100-2622-E-224-006-CC3

REFERENCES

[1] T. Leyvand, C.Meekhof , Y.-C. Wei, J. Sun, B. Guo (2011), Kinect Identity: Technology and Experience, Computer, 44(4):94-96

[2] Q. Chen, M. D. Cordeal, E. M. Petriu, T. E. Whalen, I. J. Rudas, and A. Varkonyi-Koczy (2008), Hand-gesture and facial-expression human-computer interfaces for intelligent space applications, Proceedings of the IEEE International Workshop on Medical Measurements and Applications, Ottawa, Canada, May 9-10, 2008, pp. 1-6

[3] R. Ozawa, N. Ueda (2007), Supervisory control of a multi-fingered robotic hand system with data glove, IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, California U.S.A, Oct. 29 - Nov. 2, 2007, pp.1606-1611

[4] Y. Cui and J. J. Weng (1996), Hand sign recognition from intensity image sequences with complex background, Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Killington, VT , USA, Oct. 14-16, 1996, pp.259–264

[5] E. Marszalec, B. Martinkauppi, M. Soriano, and M. Pietikäinen (2000), A physics-based face database for color research, Journal of Electronic Imaging, Zadar, Croatia, 9(1):32-38

[6] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen (2000), Using the Skin Locus to Cope with Changing Illumination Conditions in Color-Based Face Tracking, Proc. IEEE Nordic Signal Processing Symposium, Kolmården, Sweden, Jun. 13-15, 2000, pp.383-386

[7] H. J. Lee and J. H. Chung (1999), Hand gesture recognition using orientation histogram, Proceedings of the IEEE Region 10 Conference TENCON 99, Cheju Island, Korea, Sept 15-17, 1999, Vol. 2, pp.1355-1358

[8] D. Wang and S. Zhou (2008), Color Image Recognition Method Based on the Prewitt Operator, International Conference on Computer Science and Software Engineering, Wuhan, China, Dec. 12-14, 2008, pp.170-173

[9] H. Soltanian-Zadeh, J. P. Windham, and F. Chen (1994), Automated contour extraction using a multi-scale approach, 1994 IEEE Conference on Nuclear Science Symposium and Medical Imaging, Norfolk, VA , USA, Oct 30- Nov 5, 1994, vol. 4, pp.1797-1801

[10] S. N. Tamgade and V. R. Bora (2009), Motion Vector Estimation of Video Image by Pyramidal Implementation of Lucas Kanade Optical Flow, 2009 2nd International Conference on Emerging Trends in Engineering and Technology, Nagpur, India, Dec.16-18, 2009, pp.914-917