Journal of the Institute of Industrial Applications Engineers Vol.2, No.3, pp.108–112, (2014.7.25)
DOI: 10.12792/JIIAE.2.108 ISSN:2187-8811

Paper

Development of Single Camera Finger Touch Position Detection System

Shenjing Chen* Member, Lifeng Zhang* Member
Seiichi Serikawa* Member

(Received May 18, 2014, revised July 15, 2014)

Abstract: This research proposes a Single Camera Finger Touch Position Detection System. Image processing based single camera figure touch position detection approach has several significant advantages: (1) No sensing devices need to be instrumented on the surface of the touch screen. (2) Minimum sensor construction can reduce the failure rate to realize maintenance free system. (3) This approach enables an easy installation and a low-cost touch sensing. The problem of using a single camera is how to detect the touch action and the position of the finger from a single view image. In order to solve this problem, we use the reflected fingertip image appears on the back of the screen. Detecting the fingertip and the reflected image on the screen effectively, we enable to detect touch position only using single camera. In this paper, we provide concrete method, algorithm, implement details, and several experimental results.

Keywords: Finger detection, Finger tracking, Image processing, Computer vision, Touch panel.

1. Introduction

Many products with touch screen technologies appeared such as smartphones, tables and PCs because touch interface provides more flexibility and convenience than traditional interfaces such as mouse by allowing user directly and intuitively interact through finger actions. In addition to these small devices, public equipment with large touch screen such as digital signage and vending machines are also increased. The major methods of touch screen have some weak points: (1) Resistive membrane-based systems are cheap to implement, but easy to break and hard to provide sufficient sensing accuracy and support more sophisticated user operations. (2) Capacitive-based systems can provide more accurate sensing results using multiple touch contacts on the screen, but Capacitive-based sensing systems require some specific attachments under the screen [1].

Therefore, Image processing-based touch sensing systems have paid much attention recently. Image processing-based touch sensing systems have the following advantages: (1) No sensing devices need to be instrumented on the surface of the touch screen. (2) This approach enables various operations not only by touch but also by gesture operations. Nowadays, many of these approaches being researched using multiple cameras to detect finger position. However, multiple cameras could increase failure rate and installation costs. Therefore, we propose the Single Camera Finger "Touch Detection" and "Position Detection" System. Image processing based single camera figure touch position detection approach has several significant advantages: (1) No sensing devices need to be instrumented on the surface of the touch screen. (2) Minimum sensor construction can reduce the failure rate to realize maintenance free system. (3) This approach enables an easy installation and a low-cost touch sensing.

Using a single view camera to detect the touch action and touch position is a challenging task because we can not get the depth information from a single view image.

We use the reflected finger image appears on the back of the screen. Detecting the fingertip and the reflected image on the screen effectively, we enable to detect touch position only using single camera.

In this paper, chapter 2 describe the proposed system, algorithm, chapter 3 shows the implementation and several simulations. Finally, the conclusion and future work is mentioned in chapter 4.

2. Proposed System

2.1 System organization Figure 1 shows the concept of proposed system. A single camera is fixed at the top of the screen. From the camera image, recognize the fingertip to get x-coordinate and y-coordinate and recognize fingertip and reflected image using Haar-like features and Adaboost learning algorithm to get z-coordinate. If coordinates of fingertip and reflected image overlapping, determine finger has touched the screen.

Figure 2 shows process flow of our proposed system. Our system performs process as follows:

1. Capture a single view image includes fingertip and reflected image from the camera fixed at the top of the screen.

* Department of Electrical Engineering and Electronics,
Kyushu Institute of Technology, Fukuoka, Japan
chenshenjing@gmail.com

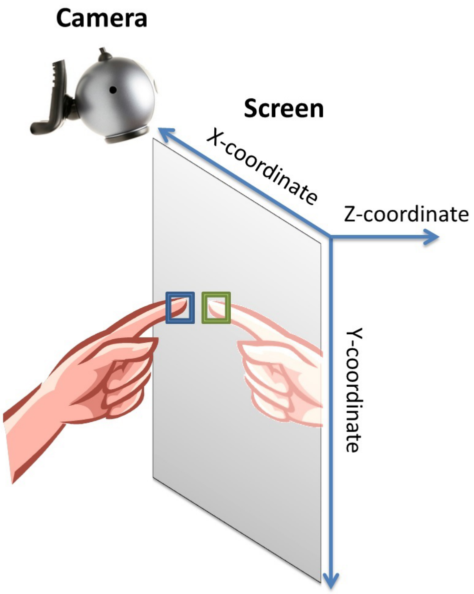


Figure 1: Concept of proposed system.

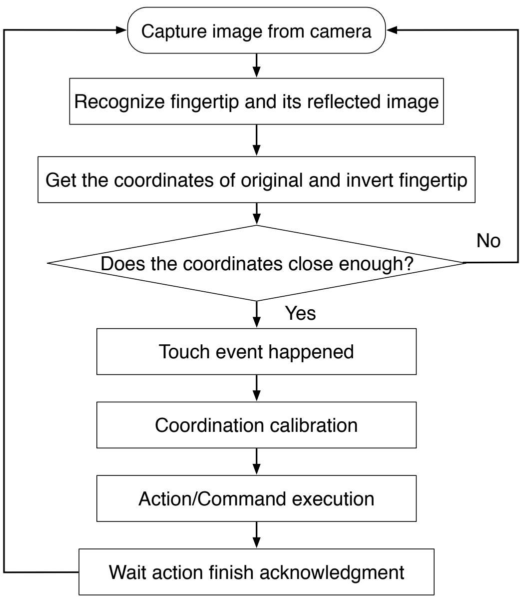


Figure 2: Process flow.

2. From the capture image, recognize fingertip and its reflected image using Haar-like features and Adaboost learning algorithm.
3. Get fingertip and reflected image position on the camera frame coordinates.
4. From two obtained coordinates, determine whether these two coordinates overlapped or closed enough.
5. If the answer is "Yes", then the finger touched the screen, the touch event happens. Else, catch a new frame and make a new detection.
6. Once a touch event happens, convert the coordinate of fingertip on the camera into the coordinate on the screen.

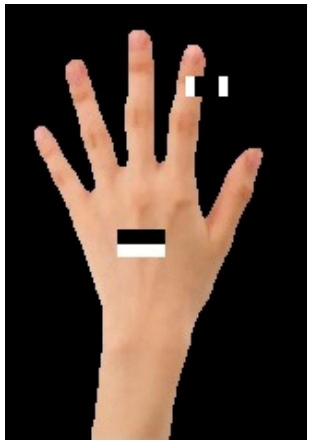


Figure 3: The extended Haar-like features set.

7. Take an action or sent a command according to the poisition of fingertip.
8. Wait until the action finish acknowledgment comes, then redo a new detection cycle.

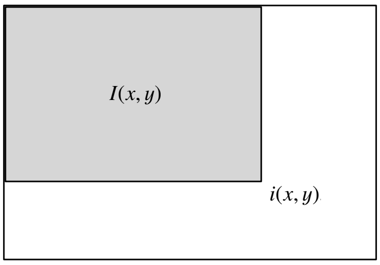
2.2 Fingertip and the reflected image recognition algorithm The method of recognizing the fingertip and the reflected image based on Haar-like features and the AdaBoost learning algorithm. Haar-like features are digital image features used in object recognition [1] [2] [3]. Haar-like feature consists of two or three "black" and "white" rectangles. Haar-like features are based on the value of simple features. This value shows characteristics like edges or borders between light and dark regions. The Haar-like features has some characteristics: (1) This approach enables very fast computation, because it depends only on the sum of pixels within a rectangle instead of every pixel value. And using an "integral image" for calculating the sum, one rectangle can be computed with only four references. According to the definition of the "integral image", the location of any point (x, y) is just the sum of all the pixels above and to the left of this pixel (x, y), which can be computed by Equation 1.

$$I(x, y) = \sum_{x' \leq x, y' \leq y)} i(x', y') \quad (1)$$

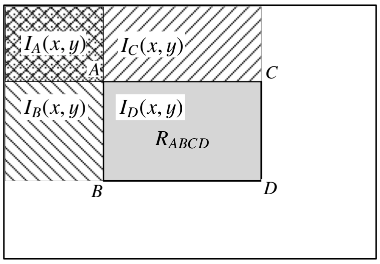
Once the summed area table has been computed, the sum of the pixel values within the shaded rectangular area in Figure 2 can be computed by Equation 2.

$$SUM = I(C) + I(A) - I(B) - I(D) \quad (2)$$

(2) Haar-like features are also relatively strong to noise and lighting changes. The noise and lighting variations affect the pixel values on the whole feature area. Haar-like features compute the gray-level difference between the white and black rectangles, and this influence can be



(a) Defination of integral image



(b) Rectangle area integral value calculation using integral image

Figure 4: Concept of the "integral image".

counteracted. Figure 3 shows the extended Haar-like features set. Figure 4 shows concept of the "integral image" [2]. In Fig. 4(a), let the gray area $I(x, y)$ means the integral value of point $i(x, y)$, then the integral value of the gray area R_{ABCD} in Fig. 4(b) can be calculated simply as $I_D(x, y) - I_B(x, y) - I_C(x, y) + I_A(x, y)$. This means if we pre-calculate the integral image, any rectangle areas integral value can be obtained by four arithmetic operations 3 times.

Single Haar-like feature can hardly identify the object with high accuracy. However, The AdaBoost learning algorithm can construct a strong classifier to improve the overall accuracy by using a linear combination of these individually weak classifiers. This machine learning algorithm is formulated by Yoav Freund and Robert Schapire [3]. The algorithm is as follow the steps: Using the probability distribution $D_t(i)$ ($t = 1, 2, \cdots, T$), we combine a weak classifier for the training data. We define this weak classifier by h_t.

Step 1: Set all sample weights equal.

$$D_{1(i)} = \frac{1}{m} \quad (i = 1, 2, \ldots, m)$$

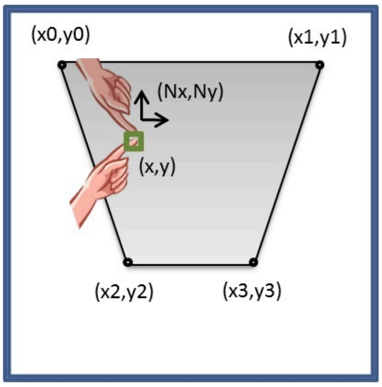
We construct the mapping.

$$h_t : X \rightarrow \{-1, 1\}$$

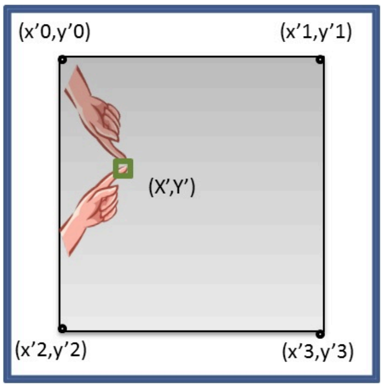
Step 2: We estimate the classification error rate for the ht classifier.

$$\epsilon_t = \frac{1}{2}\sum_{i=1}^{m} D_t(x_i) * |h_t(x_i) - y_i|$$

Step 3: Calculate the trust factor for ht. Find the weight of classifier α.



(a) Camera coordinate



(b) Screen coordinate

Figure 5: Image of coordinate transformation.

$$\alpha_t = \frac{1}{2} ln(\frac{1 - \epsilon_t}{\epsilon_t})$$

Step 4: Update the probability distribution over the training data.

$$D_{t+1}(x_i) = \frac{D_t(x_i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

Z_t is a normalization factor.

Step 5: Go to Step 2 for the next iteration. After T times iteration, the final classifier will be Equation (3):

$$g(x) = \sin\left(\sum_{i=0}^{T} \alpha_t h_t(x)\right) \tag{3}$$

2.3 Fingertip coordinate transform algorithm After detecting fingertip, it is necessary to convert the coordinate of fingertip on the camera into the coordinate on the screen because the fingertip image on the camera is inclined. Figure 5 shows the Image of coordinate transformation. The method of fingertip coordinates transform algorithm based on the idea of projective transformation. However, we add the expression (k, N_x, N_y) as revision parameters to projection transformation because normal projective transformation cannot cope with the noise and the gap of camera. Parameter k works as scale parameter and Parameter N_x, N_y works as sift parameter. Converted fingertip coordinate X', Y' can be computed by following Equation (4) using original fingertip coordinate x,y, conversion parameters a, b, c, d, e, f, g, h and revision parameters (k, N_x, N_y).

$$
\begin{aligned}
X' &= k\frac{ax + by + c}{gx + hy + 1} + Nx \\
Y' &= k\frac{dx + ey + f}{gx + hy + 1} + Ny
\end{aligned} \tag{4}
$$

Converted fingertip coordinate X', Y' can be expressed as a matrix as following Equation (5).

$$
\begin{aligned}
&\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = K \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + \begin{pmatrix} Nx \\ Ny \\ 1 \end{pmatrix} \\
&= k \begin{pmatrix} a + gN_y & b + hN_x & c + N_x \\ d + gN_y & e + hN_y & f + N_y \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}
\end{aligned} \tag{5}
$$

$$
\begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{pmatrix} = \begin{pmatrix} x_0 & y_0 & -x'_0x_0 & -y'_0y_0 \\ x_1 & y_1 & -x'_1x_1 & -y'_1y_1 \\ x_2 & y_2 & -x'_2x_2 & -y'_2y_2 \\ x_3 & y_3 & -x'_3x_3 & -y'_3y_3 \\ x_0 & y_0 & -x'_0x_0 & -y'_0y_0 \\ x_1 & y_1 & -x'_1x_1 & -y'_1y_1 \\ x_2 & y_2 & -x'_2x_2 & -y'_2y_2 \\ x_3 & y_3 & -x'_3x_3 & -y'_3y_3 \end{pmatrix}^{-1} \begin{pmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \\ x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \tag{6}
$$

Conversion parameters a, b, c, d, e, f, g, h can be computed by 8 known points (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x'_0, y'_0), (x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3) by following Equation (6). Revision parameters k, N_x, N_y are set by manual .

3. Preliminary Experiment

In order to confirm the performance of the proposed system, we have created an experimental program using C++ on Visual Studio 2012. The experimental environment is using a web camera to capture 320×240 pixel image, OS:Windows7, CPU:Core-i3 2.93GHz, Memory: 2GB.

3.1 Fingertip and the reflected image recognition experiment result

We investigated the recognition performance of fingertip and its reflected image on an anti-glare screen. Figure 6 shows the recognition and tracking result of the fingertip and its reflected image when a finger closed to the screen. Figure 7 shows the state when a finger and its reflected image overlapped on the web camera caught image. The system determines that a finger touch action happens on the screen in this state. Table. 1 shows Resolution and processing time of finger touch state detection process.

Form experiment result, in a relatively ideal environment, the proposed system can tracking the fingertip and reflected image correctly.

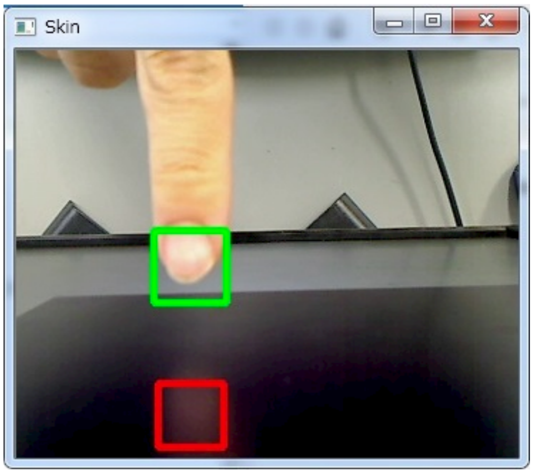


Figure 6: Tracking the fingertip and reflected image.

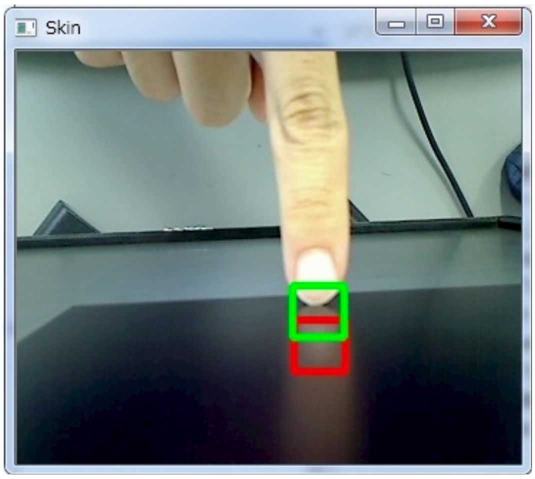


Figure 7: The state when finger touches the screen.

Table 1: Resolution and processing time of finger touch state detection process.

Resolution [pixel]	Processing Time [ms]
320×240	$100 \sim 150$

Table 2: Resolution and Processing time of proposed coordinate transform algorithm.

Resolution [pixel]	Calculation Time [ms]
320×240	$0.12 \sim 0.14$

3.2 Coordinate transformation experiment result

In order to confirm the speed of our coordinate transform algorithm, we fixed a web camera at the top of the tablet and tracking the fingertip using fingertip recognition algorithm. Execute coordinate transformation after a touch action happened to confirm the speed of our coordinate transform algorithm. Table. 2 shows the image size and the calculation time of proposed coordinate transform algorithm.

4. Conclusion

In this paper, we proposed a Finger Touch Detection and Position Detection System using a Single Camera. To detect the figure object, we use Haar-like features and Adaboost algorithm. From fingertip and the reflected image recognition experiment result, our system can tracking the fingertip and reflected image in a real time processing. Furthermore, the coordinate transformation experiment result shows that our coordinate transform algorithm obtained a very fast calculation time, and converts the coordinate of fingertip on the camera image into the coordinate on the screen correctly. However, the unstable of finger object detection is a problem to be solved. As a future work, we would like to removal of uncertainties in the surrounding environment to get improved recognition accuracy and processing time.

References

[1] Y.-L. Chen, W.-Y. Liang, C.-Y. Chiang, T.-J. Hsieh, D.-C. Lee, S.-M. Yuan and Y.-L. Chang, “Vision-based finger detection, tracking, and event identification techniques for multi-touch sensing and displaying systems”, *Sensors*, Vol.11, pp.6868-6892, 2011.

[2] Q. Chen, N. D. Georganas and E. M. Petriu, “Hand gesture recognition using Haar-like features and a stochastic context-free grammar”, *IEEE Trans. Instrum. Meas.*, vol.57, no.8, pp.1562-1571, 2008.

[3] F. Yoav and E. S. Robert. “A decision-theoretic generalization of on-line learning and an application to boosting”. *Journal of Computer and System Sciences*, Vol.55, No.1, pp.119-139, 1997.

[4] S. K. Kang, M. Y. Nam and P. Rhee, “Color Based Hand and Finger Detection Technology for User Interaction”, *Proc.IEEE Int’l Conf. Convergence and Hybrid Information Technology*, pp.229-236, 2008.

[5] M. K. Bhuyan, R. N. Debanga and K. K. Mithun, “Fingertip detection for Hand pose recognition”, *International Journal of Computer Science and Engineering (IJCSE)*, Vol.4, No.3, 2012.

[6] L. Rainer, and M. Jochen, “An extended set of haar-like features for rapid object detection”, *Image Processing. 2002. Proceedings. 2002 International Conference on*, Vol.1, pp.I-900-I-903, 2002.

[7] M. Chris and B. Andre, “Fast and efficient rotated haar-like features using rotated integral images”, *Australasian Conference on Robotics and Automation*, Vol.1, 2006.

Shenjing Chen (Member) He was born on March, 1991. He received the B.S. degree in Electrical Engineering from Kyushu Institute of Technology in 2013. He is studying in Graduate School of Engineering, Kyushu Institute of Technology, Japan. His current research interests include sensor application systems, computer vision and image processing.

Lifeng Zhang (Member) was born in Liaoning, China, on July, 1972. He received the B.S. degree in Electronic Engineering from Southeast University in 1994. And he received M.S. and Ph.D. degrees in Electrical Engineering from Kyushu Institute of Technology, in 1999 and 2001, respectively. Recently. he is an Associate Professor at Kyushu Institute of Technology. His current research interests include computer vision, image processing, biometrics authentication and remote sensing system. He is a member of IIAE, IEEE, and IEICE.

Seiichi Serikawa (Member) was born in Kumamoto, Japan, on June, 1961. He received the B.S. and M.S. degrees in Electronic Engineering from Kumamoto University in 1984 and 1986. During 1986 -1990, he stayed in Tokyo Electron Company. From 1990 to 1994, he was an assistant in Kyushu Institute of Technology. He received the Ph.D. degree in Electronic Engineering from Kyushu Institute of Technology, in 1994. From 1994 to 2000, he was an assistant Professor at the Kyushu Institute of Technology. From 2000 to 2004, he was an associate Professor at the Kyushu Institute of Technology. Since 2004, he has been a Professor at the Kyushu Institute of Technology. Recently, he is the Dean of Department of Electrical Engineering and Electronics of School of Engineering in Kyushu Institute of Technology. His current research interests include computer vision, sensors, and robotics. He is a member of IIAE,JIEE, and IEICE.