Shadow Touching for Interactive Projectors

Lan-Rong Dung, Guan-Ying Lai, and Yin-Yi Wu

Department of Electrical and Computer Engineering, National Chiao-Tung University
1001 Ta-Hsueh Rd., Hsinchu, Taiwan
lennon@faculty.nctu.edu.tw

Abstract—*Touching devices have become one of the major elements in today's most electronic devices. As the increasing demands of large touching area, the state-of-the-art touching approaches become costly and infeasible. Therefore, it is essential to design a new kind of touch techniques with high touching accuracy and scalability with touching panel size. The purpose of this paper is to provide a touch system that uses the distance between the object (finger or stylus) and its shadow to detect the touch-timing and position. It can be applied to an interactive projection system without using large amount of touch-detecting elements. The proposed touch system only requires a camera and an IR source with an interactive projector to detect the occurrence of touching and its location. The proposed system achieves an average detection rate of 97.53% when the error tolerance is 10 pixels.*

Index Terms—*shadow detection, touching detection, image processing, object segmentation, interactive system*

1. Introduction

Touch panel, which is different from other input devices that need the learning process as keyboard, is easy to use and quite popular due to its instinctive input method as a man-machine interface. There are three units in this kind of system. Most of them have the sensor and coordinate calculation unit, and some have the special signal project unit. The sensor unit is used to receiving signal and generating information that can define the coordinate. The coordinate calculation unit takes care of the information generated by sensor unit and transforms the information to coordinate or some other signal that can judge the location. And there are some special touch system such as electromagnetic touch system must have additional project unit to generating specific signal received by sensor unit.

The representative of medium and small touch system covered the entire displaying surface with a matrix of resistors or capacitors. However, with the increasing of panel size, the number of sensor device is also increased and cause the yield los and large cost. Therefore, these types of touch systems are not suitable for large touch screen.

With the techniques improved, the screen size promoted bigger and bigger. But touch system face to more problems than general screen. The first problem is that the sensor unit is increased with screen size such as resistors or capacitors touch screen. The second problem is the yield issue which requests a higher process technique with the increase of panel size. And this problem decreases the probability of mass production and increases the cost. On the contrary, the type of ultrasonic, optical, electromagnetic, infrared, and vision based touch system which sensor device is not increased with the panel size are more feasible to use as a large or super large touch system. Nevertheless, these touch systems still have drawbacks. For example, the cost of ultrasonic based touch system is still expensive. The electromagnetic based touch system is limited to an electromagnetic pen. The infrared and vision based system need more than one camera to locate the dimension.

This paper target at the vision based touch system to use as a super large size touch system. The vision based touch system is quite suitable for large touch screen. It has camera that can catch images to calculate the location and decide the event of touching. There are many similar technique proposed as a vision based touch system [1]-[5].

The key issue of this touch system is that the movement of gesture is just a move of transpose or really a touch event [2].

The simplest solution is to use two dimension cameras. One is located parallel to the screen to detect whether the finger is touch the screen or not, and the other is located with a specific angle in front of screen to locate the coordinate or judge the gesture. The second solution is to use the gesture to define the touch event and movement [1]. But the main drawback is that the operation speed is limited by the switching time of touch-mode and move-mode. Also, this solution need user to learn and fit and it is not instinctive and not user-friendly for user.

The third solution [6] is to use the amount of time that finger stay to judge whether finger is on the touch point or just moving in front of the screen. In this solution, finger need to stay a long time at the begging and ending to judge the touch point and trace, therefore; the detecting speed is also limited.

This paper proposed a vision based touch technique that use the shadow information to detect the touch event and location. This system using only one camera and do not use the gesture judgment to tell apart the touch event and moment that is more instinctive, user-friendly, and save more cost opposite to two camera system.

2. Proposed Touch System

2.1. The framework of proposed touch system

The proposed touch system is depicted in Figure 1. The touch system includes an interactive projector which mounts an infrared camera, an IR filter, a projection screen and an infrared light source. The interactive projector projects contents on projection screen, and the camera capture user's action in front of projection screen. The IR filter is installed

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

in front of the camera lens to filter out interferences from the projection contents and other visible light. The infrared light source supply illumination to the system. The images captured by camera will be analysed. Therefore, when the user touches projection screen, the system will detect user's touch, and estimating the touched location.

Comparing to other optical imaging touch screen, the proposed touch system only needs one camera. Therefore, it is superior in cost and establishment.

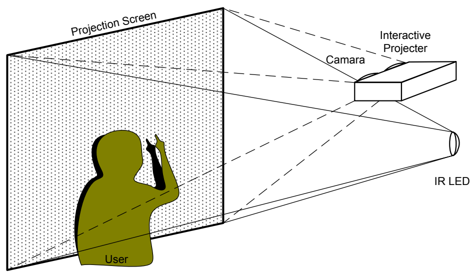
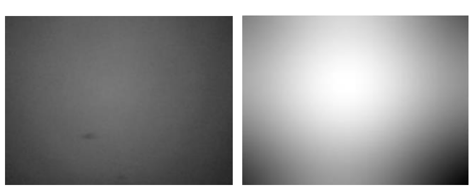


Figure 1. Proposed touch system

2.2. Initialization

The system must be initialized after setting up at the first time. The first reason is the illumination supplied by infrared light source is non-uniform; hence, we must correct the non-uniform illumination, or the extracted object and shadow will be inaccurate. The second reason is we have to estimate the threshold of extracting shadow and object.

For non-uniform illumination correction, we refer to the method proposed in [7]. Paper [7] proposed a method of illumination estimation based on the intensity gradients instead of intensity means. After estimating illumination, we can correct the captured image's illumination by the estimated illumination. Figure 2 shows an example of illumination correction, Figure 2(a) is the background without illumination estimation, Figure 2(b) is the estimated illumination of Figure 2(a), and Figure 2(c) is the background with illumination correction. The estimated illumination and corrected background will be saved for detecting user's touch.

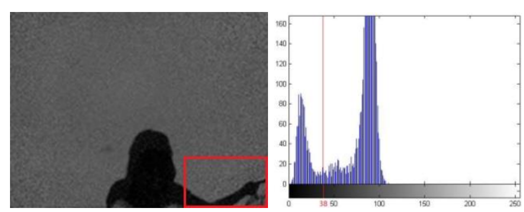


(a)Original background (b)Estimated illumination

(c)Corrected background

Figure 2. Illumination correction

For estimating the threshold of extracting shadow and object, we refer to the OTSU's method [8]. OTSU's method assumes that the image to be thresholded contains two classes of pixels, and then calculates an optimum threshold can separate these two classes optimally. It involves iterating through all the possible threshold values and finding a threshold to let intra-class variance of two classes is minimal. We require an user to touch projection screen in a limited range and making histogram of the limited range, then we use OTSU's method to find the threshold of extracting shadow and object. Figure 3 is an example of calculating the threshold. As seen in Figure 3(a), an user touches projection screen in a limited range (red frame). Figure 3(b) is the histogram of Figure 3(a), the optimum threshold calculated by OTSU's method is 38 in this example. After that, the threshold of extracting shadow and object will be measured.



(a) (b)

Figure 3. Calculating threshold of extracting shadow and object
(a)To touch screen in a limited range
(b)Finding optimum threshold by OTSU's method

2.3. Detecting touch and estimating touch location

The touch system can detect user's touch after initialization. The image captured by camera will have much noise, so we use a 3×3 mean filter to reduce noise. Besides, illumination of image is non-uniform, so we have to correct illumination of every frame by estimated illumination. Figure 4 is an example of the frame after noise reduction and illumination correction.

After noise reduction and illumination correction, we use thresholding and background subtraction to get preliminary bi-level image of object and shadow, as shown in Figure 4(b) and Figure 4(c). The threshold we used is estimated in initialization.

Preliminary segmentation of object and shadow may have many errors; therefore, we should use some methods such as morphology and majority filter to reduce these errors. First we use erosion to erode away the boundaries of regions of foreground pixels, it can reduce some error of segmentation. Second, majority filter is used on the image after erosion to reduce some noise. Third, closing is applied to the image after majority filter, the small holes or breaks in foreground will be filled up. Finally, we use connected component labelling [9] to label the different region in image, and we count the area of every region. The regions which are too small will be eliminated, it can reduce remaining errors. The final bi-level image of object and shadow is shown in Figure 4(d) and Figure 4(e).

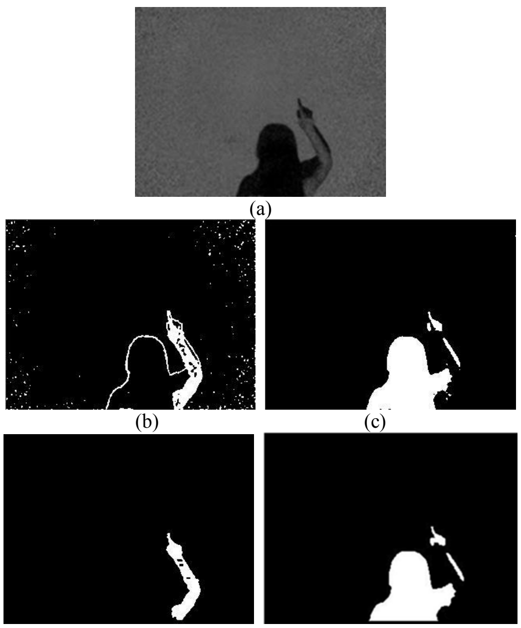


Figure 4. Extracting object and shadow

(a) The frame after noise reduction and illumination correction
(b) Bi-level image of object after thresholding
(c) Bi-level image of shadow after thresholding
(d) Final bi-level image of object
(e) Final bi-level image of shadow

After extracting the bi-level image of object and shadow, we have to find the fingertip's possible location. We detect the tips location of object and shadow by scanning the four edge of a 16×16 mask, as shown in Figure 5. If there is only one edge touches the object or shadow, the tip is detected; otherwise, there is no tip in the block contained by mask. When the tip is detected, we will calculate height and width of the tip. If the tip's height is large enough and width is small enough, the direction and the coordinate of the tip will be calculated and recorded. Figure 6 is an example of detected tips of object and shadow. The pink points is the detected tips.

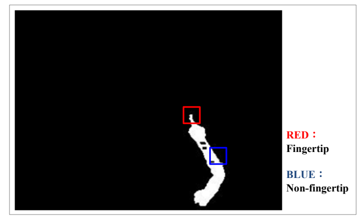


Figure 5. Tip detection

(a)Detected shadow tips (b)Detected object tips

Figure 6. Detected tips

After detecting the tip's of object and shadow, we use the information of tips to determine touch-timing and touch-point. We trace the tips for more than one frame, and make use of four parameters to judge whether the object's tip touches the projection screen. The four parameters are tip's *Direction*; the *Distance* of object's tip and shadow's tip; *Frame Counts*, which is the number of frames that include the same tip; and the *Approaching Velocity* of object's tip and shadow's tip. If the object's tip and shadow's tip is same direction and close enough, there is a great possibility that the object's tip is the peak of finger or stylus. If two successive frames have the closing object's tip and shadow's tip, the *Frames Counts* will increase one. Then we calculate the *Approaching Velocity* of the object's tip and shadow's tip, if the object's tip and shadow's tip is closer of two successive frames, the approaching distance is added by the difference of the distance of object's tip and shadow's tip in two frames. *Approaching Velocity* is approaching distance divided by *Frames Counts*, if object's tip and shadow's tip is closer of these successive frames, the *Approaching Velocity* will be positive. The formula of calculating Approaching Velocity is shown as Equation(1).

$$\begin{aligned} &\text{Approaching Distance}_{(\text{current})} \\ &= \text{Approaching Distance}_{(\text{previous})} \\ &+ [\text{Distance}_{(\text{previous})} - \text{Distance}_{(\text{current})}] \end{aligned} \quad (1)$$

$$\text{Approaching Velocity} = \frac{\text{Approaching Distance}}{\text{Frame Count}}$$

We use two conditions to determine whether the object's tip touches the projection screen. If we judge that the object touches the screen, the touch-point is the position of object's tip. The first condition is the *Frames Counts* is more than seven, the *Approaching Velocity* is larger than threshold, and the *Distance* of current frame is smaller than threshold. This means the same tip appears more than half second, and the object's tip is closer during this time period and the object's tip is very close to the projection screen. The second condition is the shadow tip is disappeared in current frame, and *Frames Counts* is more than four before current frame, and *Approaching Velocity* and *Distance* is satisfied the threshold. The second condition is for faster touch; that is, the movement of touch is too fast so that before the *Frames Counts* reach seven, the shadow's tip is covered with finger or stylus.

3. Result and Discussion

The experiment environment is shown as Figure 7. There are an interactive projector, and a camera which mounts an IR filter in front of camera lens, and an infrared light source and a projection screen which diagonal distance is about 100 inches in the touch system.

Figure 7. Experiment environment

We use 11 test patterns to test the performance of the proposed touch system. The test patterns are videos that frame rate is 15 frames per second, and the frame size is 320 by 240 pixels. The performance of the proposed method is shown as Table 1. The proposed system achieves an average detection rate(DR) of 97.53% and false alarm ratio(FAR) of 0% if the tolerance of detected touch coordinate is +/- 10 pixels. The formula for calculating detection rate and false alarm ratio is shown as Equation(2). Touch number is the number of touching in test pattern, and Hits is the number of correct touching detected by the proposed method. False Alarms is the number of detecting error, and Detected touch number is the number of detected touching by the proposed method.

$$DR = \frac{Hits}{Touch\ number} \times 100\%$$
$$FAR = \frac{False\ Alarms}{Detected\ touch\ number} \times 100\% \qquad (2)$$

Table 1

Detection rate and False Alarm Ratio

Test Pattern	*Touch number*	*Hits*	*Detection Rate (DR)*	*False Alarms*	*False Alarm Ratio (FAR)*
1	81	81	100.00%	0	0%
2	122	119	95.90%	0	0%
3	51	49	96.08%	0	0%
4	72	72	100.00%	0	0%
5	104	101	97.16%	0	0%
6	83	79	95.18%	0	0%
7	87	84	96.55%	0	0%
8	58	58	100.00%	0	0%
9	59	59	100.00%	0	0%
10	61	58	95.08%	0	0%
11	73	70	95.89%	0	0%
Total	851	830	97.53%	0	0%

The processor is Intel Pentium Dual CPU E2180 2GHz, and the proposed method is developed by C. The average computation time of dealing one frame is about 0.029 seconds, so we can detect the touch-timing and position at 33 frames per second video. That is, our method can achieve real-time computing for an ordinary computing system.

4. Conclusion

This paper proposed an optical imaging touch system which takes advantages of the relative relationship between finger or stylus and its shadow to determine whether a finger or stylus touches the screen. Touch system proposed by this thesis needs only one camera and one IR source, and acts with an interactive projector to detect the occurrence of touching and its location. The proposed system does not use more touch detecting elements for a larger touch screen. That is, no additional costs are needed for a great touch screen. Comparing to other large-size touch panels, the proposed touch system needs only one camera and can be moved easily, therefore the touch system proposed in this thesis is superior in cost and establishment.

Experiments show that our method achieves average detection rate of 97.53% if the tolerance of detected touch coordinate is +/- 10 pixels. Moreover, our method can achieve real-time computing for an ordinary computing system.

References

[1] Z. Mo, J.P. Lewis and U. Neumann, "Smart Canvas: A Gesture- Driven Intelligent Drawing Desk System," In Proc. of the 5th ACM symposium on Intelligent User Interfaces, pp. 239-243, 2005.

[2] L. Bretzner, I. Laptev, and T. Lindeberg., "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," In Proc. of Face and Gesture, pp. 423–428, 2002. (Demo video is available at http://www.nada.kth.se/cvap/gvmdi/DrawBoard2.mpg.)

[3] X. Chen, H. Koike, Y. Nakanishi, K. Oka, and Y. Sato, "Two-handed drawing on augmented desk system," In Proc. of Working Conference on Advanced Visual Interfaces, May 2002.

[4] D. Hall, C.L. Gal, J. Martin, O. Chomat, and J.L. Crowley, "Magicboard: a contribution to an intelligent office environment," In Proc. of the International Symposium on Intelligent Robotic Systems, vol. 35, no. 3, pp. 215–223, Jun. 1999.

[5] P. Wellner., "Interacting with paper on the digital desk," ACM Trans. on Communications, vol. 36, no.7, pp. 86–97, Jul. 1993.

[6] C. Hardenberg, "Fingertracking and Handposture Recognition for Real-Time Human-Computer Interaction,"

http://iihm.imag.fr/hardenbe/, Jul. 2001.
[7] T. Tasdizen, E. Jurrus, and R. T. Whitaker., "Non-uniform illumination correction in transmission electron microscopy," In MICCAI Workshop on Microscopic Image Analysis with Applications in Biology, Sep. 2008.
[8] N. Otsu, "A Threshold Selection Method from Gray Level Histogram," IEEE Trans. on Systems, Man and Cybernetics, vol. 9, pp. 62–66, Jan. 1979.
[9] K. Wu, E. Otoo, and K. Suzuki., "Optimizing two-pass connected-component labeling algorithms," Pattern Analysis & Applications, vol. 12, no. 2, pp. 117 – 135, 2009.
[10] L. R. Barnes, D. M. Schultz, E. C. Gruntfest, M. H. Hayden, and C. C. Benight, "Corrigendum: False alarm rate or false alarm ratio?," Weather Forecasting, 2009.