VTouch: A Vision-Base Dual Finger Touched Inputs for Large Displays

Ching-Han Chen and Cun-Xian Nian

National Central University, Computer Science & Information Engineering Department, No.300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan
pierre@csie.ncu.edu.tw, yalight@hotmail.com

Abstract. In this work, we present a complete architecture to implement a dual touch function for large sized displays. The architecture includes the hardware device and the software algorithm. The system has lower cost and smaller size than previous works for large sized display. And the proposed approach eliminates the requirement of special hardware. The system can be plugged to a normal display and change the display to equip the dual touch function. At the final of this paper, we give an experiment to demonstrate the system and show its usability.

Keywords: Human-computer interaction, dual touch screen, multi-touch screen, user interface.

1 Introduction

To touch and manipulate data on screens directly without any intermediary device provides more convenient and instinctive using experience to the user. There are several implements of multi-touch displays, but most current techniques need special hardware for the touching ability. For instance, spatially segmented antennas to sense the changes in electronic capacity are employed in [1], [2] and projected electronic capacities are used in iPhone [3]. Another approach proposed in [4] uses a frustrated total internal reflection (FTIR) device to detect the fingers optically. Moreover, current implements are either still too experience or too bulky for widespread adoption especially to large displays.

This work proposed a vision-base system to achieve the dual fingertip touching ability for large displays. This system eliminates the need of special hardware. It only employs two cameras with wide-angle lens and applies image processing techniques to detect the fingertip touched on the display. It can be mounted on a general monitor and turn it to a touch screen. It is cheap and thin compared with previous works.

With this system, user can manipulate the screen directly with their one or two fingertips. For examples, a dual touché display can be installed on a museum wall to be a versatile screen. The user can drag, zoom or rotate photos on the screen by their fingertips, or play two people interacted games on the screen.

J.A. Jacko (Ed.): Human-Computer Interaction, Part II, HCII 2009, LNCS 5611, pp. 426–434, 2009.
© Springer-Verlag Berlin Heidelberg 2009

2 Approach

In the main idea, there are two cameras to capture the two scenes above the surface of the display. Then, the system applies the detection algorithm to detect whether there is any object touched the display. If there is any object touched the display, it will compute the coordinates of the objects on the display. The following are the detail descriptions of hardware and software in this system, respectively.

2.1 Hardware Device

The major components are two cameras in the hardware mechanism. The two cameras are installed on the two cutting edges of the display and the optical axis of each camera is parallel with the surface of the display. For capturing the full scene above the surface of the display, the wide-angle lenses are employed. Besides, there is a set infrared rays (IR) LED modules at the side of each camera. The IR LED provides light source when the environment is dark. A thin border which can absorb infrared rays is appended on the display. The border is to reduce environmental interferences and make the system more robust. Fig. 1 shows the illustration of the hardware device.

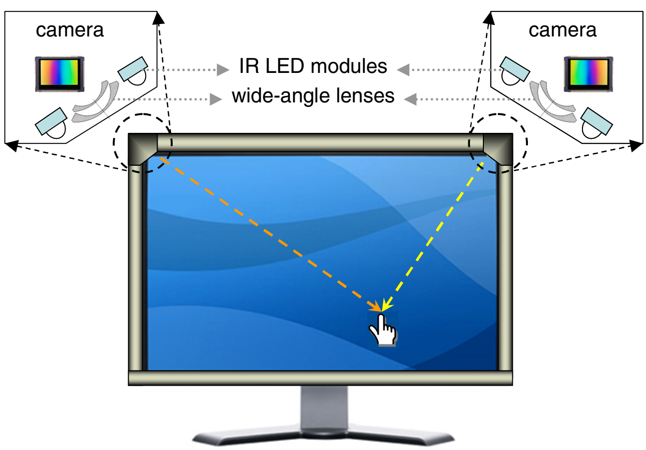

Fig. 1. Two cameras with wide-angle lenses are on the top-left and top-right corner of the display. The IR LED modules are at the side of the camera. The optical axes of cameras and IR LED are all parallel to the surface of the display. The display is surrounded by a thin border which can absorb the infrared rays.

2.2 Software Algorithm

When the two cameras capture a set of scenes, the system receives the left and right images from the cameras and applies the algorithm on the images to achieve the touch ability of the screen. The algorithm detects whether there is any object touched on the display. If so, it calculates the coordinate of the object. The main procedure of the algorithm can be separated to four stages and they are image pre-processing, locating

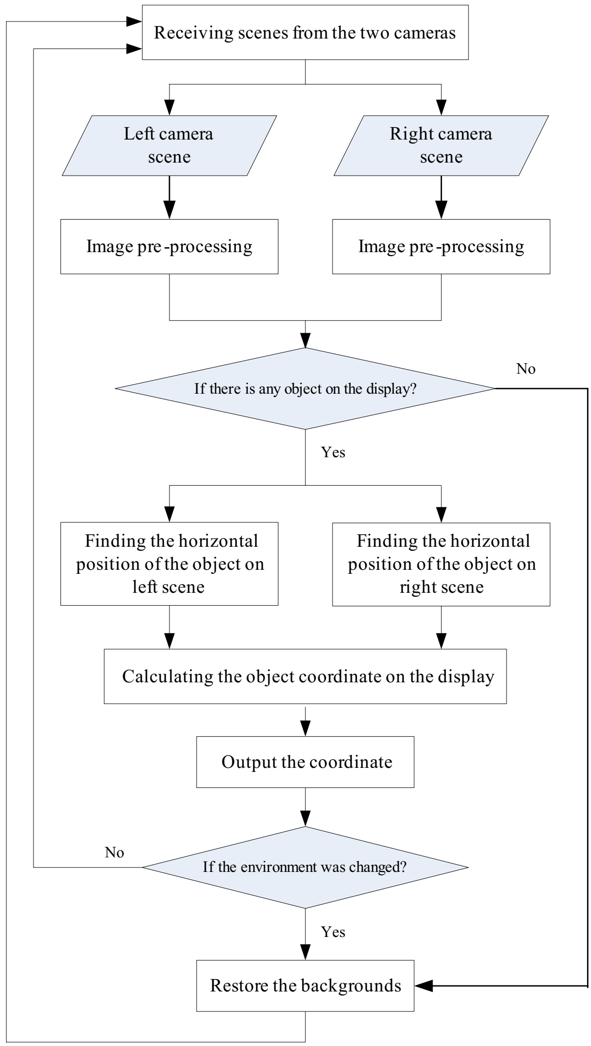

Fig. 2. Overall flowchart of the algorithm in the system

coordinate, object tracking, and restoring background. Fig. 2 shows the procedure of the algorithm and following is the detail descriptions for each stage.

For the camera, it can capture a full size image. However, the lens axis of the camera is parallel with the surface of the display. We can detect whether any object touched the display by the intersection between the screen surface and the surrounding. The intersected region is small and thin in the full image. This reduces size of data needed to process and enhance the performance. The red rectangle in fig. 3 is the region used in our system implementation.

Fig. 3. The red rectangle is the region is the intersection of screen surface and the surrounding and it is applied to our algorithm

Image Pre-processing. In this stage, a serial image processing procedure is applied on the two received scenes. It includes the color transformation, subtracting background, image median filter, image bi-level, horizontal image profile converter, and equalization. Fig. 4 shows the flowchart and the sample images of each step of the image pre-processing.

The camera sensor can capture color images, but we lose most color information of the captured image due to the IR filter. The captured images are looked like the source image in Fig. 4. For the reason, the color transformation in image pre-processing simply translates the received images to gray level images. The gray level image is easier and faster to process. Then, we take the current image to subtract the background image, and we can get a clear image to identify if there is object touched the display. The background image is stored at the system initialization. For reducing the noise, which may be produced by camera or environmental interference, a median filter is employed. In the next, we project the image to a horizontal image profile which counting how many bright pixels in each column of the image. Subsequently, we equalize the horizontal image profile to smooth it. The equalization replaces each

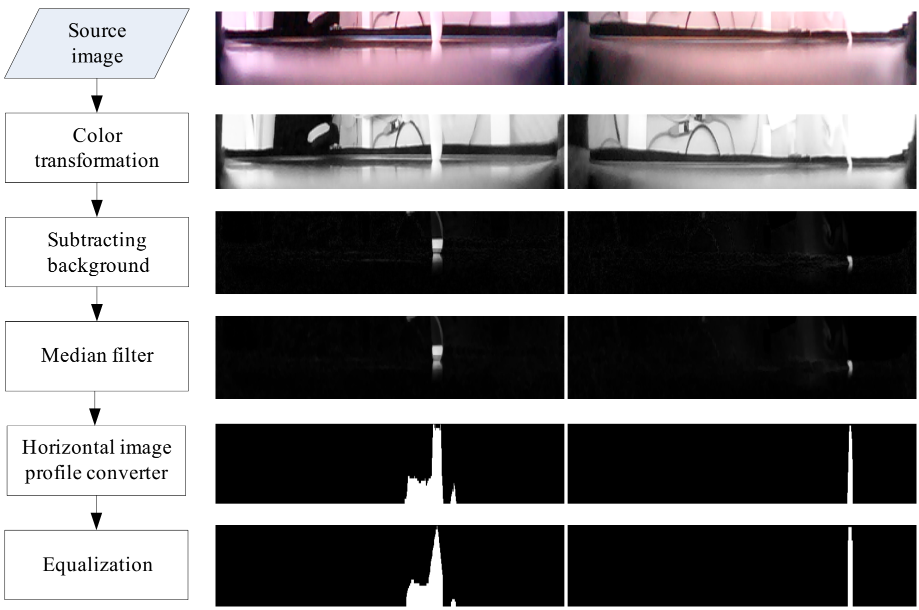

Fig. 4. The flowchart of the image pre-processing and the corresponding sample images of each step

element in the horizontal image profile with the average value of near the elements. The equalization can be expressed as the following equation.

$$\overline{H}_i = \frac{1}{k}\sum_{j=1}^{k} H_{i-j+k/2} \tag{1}$$

Where H_i is the value of the *i*th element in original horizontal image profile and $\overline{H}_i$ is the value of the *i*th element in the equalized horizontal image profile, and k is the degree of equalization. The degree of equalization decides the degree of smooth.

Locating Coordinate. After image pre-processing, the system detects whether there is any peak in the two horizontal image profiles. If so, it finds out the horizontal positions of the peaks in the two horizontal image profiles. Fig. 5(b) is the procedure of finding the peaks and Fig. 5(a) is the sample images of the procedure. At the beginning of the procedure, it searches the element $\overline{H}_i$ with the maximum value in the horizontal image profile $\overline{H}$ which got from image pre-processing. If $\overline{H}_i$ is greater than the threshold, the i is considered as the horizontal position of the peak in $\overline{H}$. If not, the system considers there is no peak exists. In other words, it represents there is no object touched the display. After the first peak is found, the system eliminates the decreasing region from the *i*th element in $\overline{H}$, and the residual horizontal image profile is $\overline{H}'$. To find the second peak, the system applies the similar procedure on $\overline{H}'$ as finding the peak in $\overline{H}$. It searches the element $\overline{H}_j$ with maximum value in $\overline{H}'$, and the j is considered as the horizontal position of the second peak if $\overline{H}_j$ is greater than the threshold. Otherwise, there is considered only one peak existed in $\overline{H}$.

After the system gets the horizontal positions of the peaks in two horizontal image profiles, it translates the horizontal positions to the angles between the touched object and the two cameras. The angles are as the θ_1 and θ_2 in Fig. 6. Due to the distortion of the wide lens, we build a lookup table which maps the horizontal position to the angle. The mapping looks like Fig. 7.

With having the θ_1 and θ_2, the coordinate of the object touched the display can be calculated by the following equation.

$$x = \frac{W \tan\theta_2}{\tan\theta_1 + \tan\theta_2}, \quad y = x \tan\theta_1 \tag{2}$$

In equation (2), the W is the width of the display. If only one object touched the display, the x and y in equation (2) is the result coordinate of the object. However, for two objects on the display, equation (2) gives two pairs of coordinates. Fig. 8 shows the diagram. The correct coordinates of the two objects are among the two pairs of coordinates. To solve the problem, this system adopts the object tracking technique to eliminate the false coordinate and output the true one.

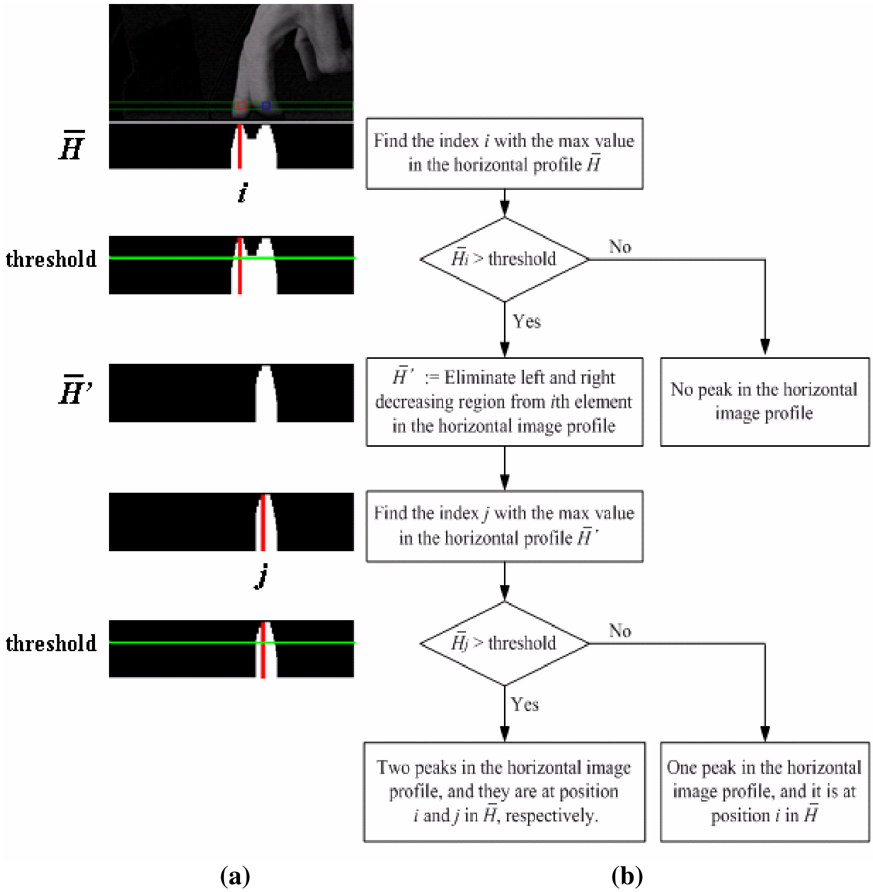

Fig. 5. (a) The sample images of the procedure of finding peaks. (b) The flowchart of the finding peaks.

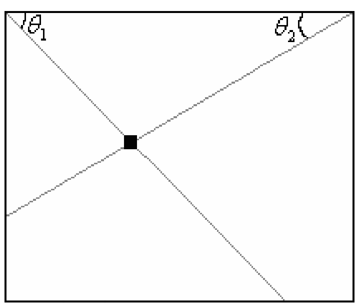

Fig. 6. The square point is the location of the object touched the display. The system can calculate the coordinate of the location with the two angles θ_1 and θ_2 .

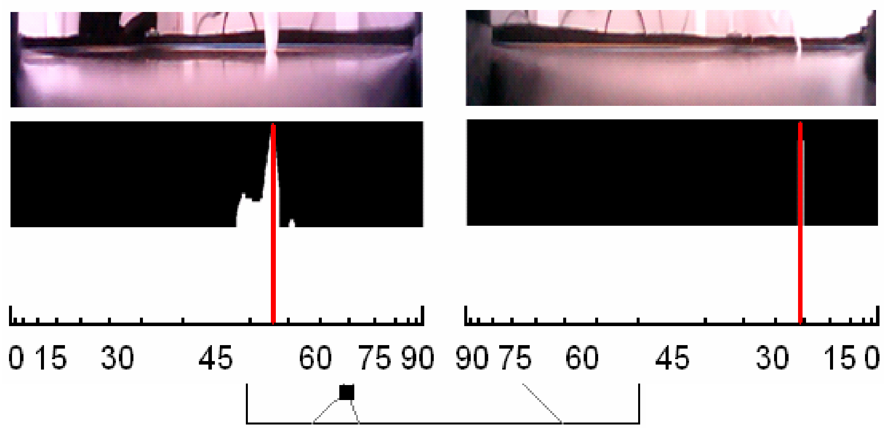

Fig. 7. The two square points are the true positions of the two objects on the display and the circle points are the false positions

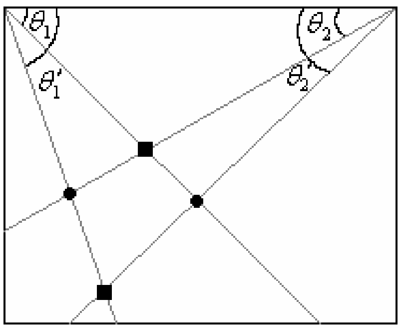

Fig. 8. The two square points are the true positions of the two objects on the display and the circle points are the false positions

Object Tracking. The object tracking is used when the system detects there are two object touched the display. For tracking the object, we use the previous coordinate of the object touched the display to compare with the two possible pairs of coordinates which got from previous stage. The scheme of comparison is to select the closer pair of coordinates to be the true coordinates that object touched. The closer pair has the shorter distance with previous coordinates.

With this object tracking scheme, there are two distinct situations need to be handled separately. The first case is there is one object touched the display in previous detection. In this case, the system calculates the distances between previous coordinate and the two possible pairs of coordinates, and takes the pair of coordinates with the shortest distance as the output. Fig. 9(a) shows the diagram of this case. The second case is there are two objects touched the display in previous detection. For this case, the system calculates the distances between previous pair of coordinates and the two possible pairs of coordinates, and select the pair of coordinates has shorter distance as the output. Fig. 9(b) is the diagram of this case.

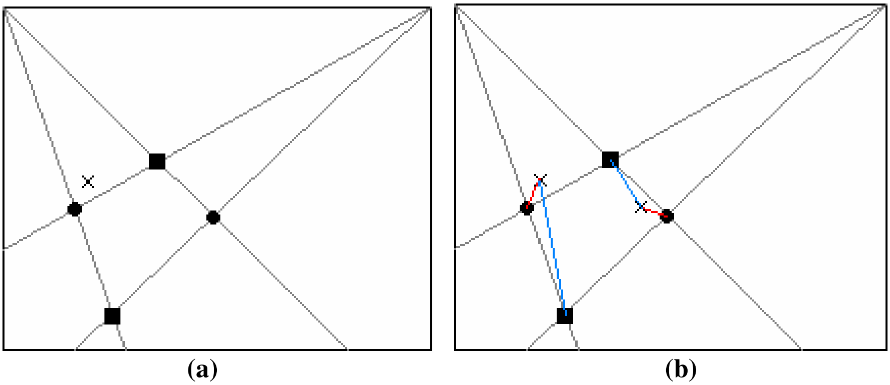

Fig. 9. (a) The circle points and square points are two possible pairs of coordinates, and the cross point is previous coordinate. The circle point is closer to the cross point, so the circle points are selected as the output. (b) The red and blue lines are the two distances between two possible pairs of coordinates and previous pair of coordinates. The length of red line is shorter, so the circle points are selected as the output.

Restoring Background. In the system, the changes of the environment will affect the accuracy of the object detection. The changes of the environment may come from the changes of light, the moves of the display, camera noises and so on. To overcome these interferences, the system measures the changes of the environment and decides to whether restores the background scenes in the end of the algorithm.

The scheme of measurement is to count how many columns have a great change. The comparison is based on the background image and current image. A column is considered has a great change if there are many pixels changed in this column. If a lot of columns are counted to the changed columns, then the system replaces the background image with current image, and uses the new background to process.

Fig. 10. The user scrolled the pohoto wall by one finger and selected a interested photo

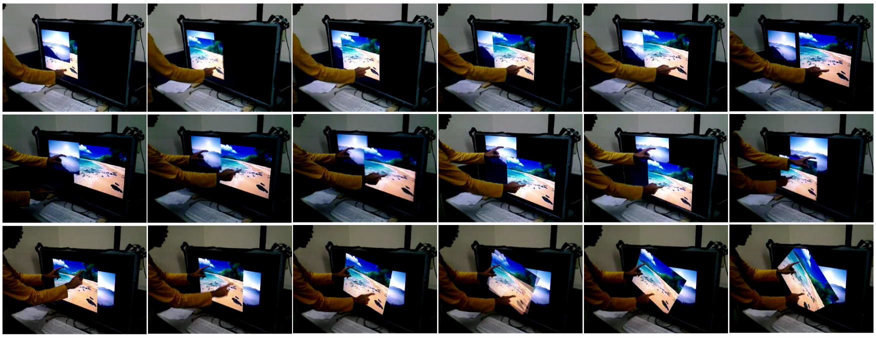

Fig. 11. At the top row, the user moved a photo with his one finger. At the second row, he moved the two photos simultaneously with his two fingers. The last row is the rotation of the photo with his two fingers.

3 Experimental Results

Two experimental results with this system are given in this section. The fist one is the manipulations of PicLens [5] which is a photo viewer application. In this experiment, it shows the ability of single touch. Fig. 10 is serial images of the demonstration. The second one is to show the ability of dual touch. The second one demonstrates the move, scale and rotation on photos with two fingertips. Fig.11 shows serial images of the second demonstration.

4 Conclusion

A multi-touch display can bring an instinctive interaction between the user and their display. In this paper, we proposed a hardware device and software algorithm to implement a dual touch for large displays. It improves the disadvantages of expensive cost and bulk, and eliminates the requirement of special hardware. We have demonstrated the system with some applications and vilified the usability of the system.

References

1. Dietz, P., Leigh, D.: DiamondTouch: a multi-user touch technology. In: Proc. of UIST 2001, Orlando, FL, pp. 219–226. ACM, NY (2001)
2. Rekimoto, J.: SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In: Proc. of SIGCHI 2002, Minneapolis, MN, pp. 113–120. ACM, NY (2002)
3. Hotelling, S., Strickon, J.A., Huppi, B.Q.: Multipoint touchscreen. U. S. Patent Application 20060097991 (2006)
4. Han, J.Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In: Proc. of UIST 2005, Seattle, WA, pp. 115–118. ACM, NY (2005)
5. [Online] PicLens, `http://www.cooliris.com/`