A New Method Of Detecting Fingertip Touch For The Projector-Camera HCI System

Xilei Cai [1], Xiang Xie[1], Guolin Li[2], Wei Song[1], Yi Zheng[1], Zhihua Wang[1]
Tsinghua National Laboratory for Information Science and Technology
[1]Institute of Microelectronics, Tsinghua University, Beijing 100084, China
[2]Electronic Engineering Department, Tsinghua University, Beijing 100084, China

***Abstract*—This paper proposes a new fingertip touch detecting method with high accuracy for the human-computer interaction based on one projector and a single camera. A finger model composed of a quarter-sphere and half a cylinder is introduced, and the relationship between the finger contour and its shadow based on this model is given out. The way to estimate the distance between finger and the projected surface is presented and discussed. Experimental results show that our approach can achieve a precision of about 2mm when the distance from the projected surface to the projector is 50cm and the button size will not influence the precision.**

I. INTRODUCTION

Traditional way of Human-Computer Interaction (HCI), for instance, keyboard, mouse and LCD display screen, has become the bottleneck of the miniaturization of electronic devices with HCI such as today's smart phones. Recently, many researches have been done to explore new possibilities to make the HCI more natural and portable. With the emergence of the pico-projectors, some of the researchers try to make use of it as well as the camera to make the projected surface into the touch-sensitive input device. Andrew D.Wilson's PlayAnywhere[1,2] and OmniTouch[3] adopted a mini-projector as output device, and a depth camera, Microsoft Kinect, as the input device. Since Kinect can acquire the depth of every pixel in the background image, it can be used on not only flat plane, but also on any non-flat surface. However, the input device seems not easy to be further miniaturized and the depth resolution also makes the touch detection accuracy not enough. Its precision is 7mm when the distance from Kinect to the projected surface is 0.75m. Jingwen Dai[4] employed a projector and single camera to make any planar surface into touch-sensitive display. Simple computer vision algorithm makes it fast, but its precision is about 3 pixel when the system was configured for a working distance of about 500mm. J.Hu[5] came up with a novel finger touch detection method by using the button's distortion. However, the size of button will influence the finger touch precision.

This paper mainly focuses on measuring the distance between the finger and the projected surface. Based on a finger model and the geometry of fingertip point and its shadow point, a new touch detection method with high precision is proposed and applied for a virtual keyboard application. Moreover, the button size will not influence the detection precision in this method.

The remained parts are organized as follows. In section II, a fingertip model is introduced and the relationship between finger and its shadow is discussed. In section III, the distance computation for the touch detection is analyzed. In section IV, the proposed method is applied for the projected virtual keyboard with big buttons. Experimental results and a conclusion are given out in section V and VI respectively.

II. MODEL OF A FINGER AND THE RELATIONSHIP BETWEEN THE FINGER AND ITS SHADOW

The proposed finger model, which consists of a quarter sphere and half a cylinder, is shown in Fig.1. The sphere center is S_C and the radius is r. If point B is on the surface of the finger model and line S_CB is perpendicular to the projected surface, then point B is the point on finger model closest to the projected surface. Distance from point B to the projected surface represents the distance between finger model and the projected surface.

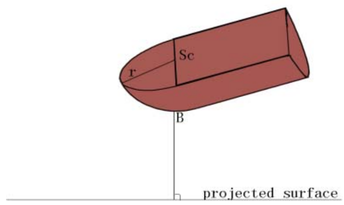

Figure.1. Finger model

Based on the finger model, the relationship between the finger and its shadow under the pro-cam system can be shown in Fig.2. In general, the finger shadow is mainly caused by the projector in the pro-cam system. The edge of shadow is produced by the tangent points between the finger and the projected light. For example, the edge point of shadow, M, is generated by the projected light PA, and A is the tangent

Supported by the National Natural Science Foundation of China (grant No. 61373073).

point on the finger. Point B' is the tangent point between finger and the incident ray of the camera, and it will be an edge point of the finger in the image captured by the camera C. When points P,C,A,B' are on the same plane, the length of PC and radius r are far shorter than the length of PS_C, then the intersection D between line PA and B'C has almost the same distance from the projected surface as point B' because point A and point B' are very close.

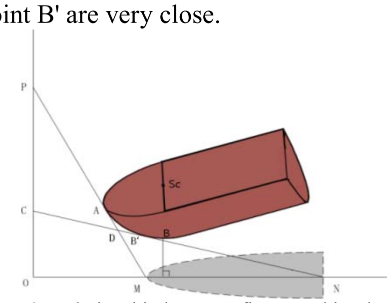

Figure.2. Relationship between finger and its shadow

The relationship between the finger and the optical center of camera can be illustrated in Fig.3. In order to calculate tangent points between the finger and point C, the coordinate is built in the following way: the sphere center of the finger, S_C, is taken as the origin with z-axis perpendicular to the projected surface, y-axis is perpendicular to z-axis as well as in the plane $S_C CC'$, x-axis can be determined by the right hand rule. So, the z-axis value of one point can represent its distance to the projected surface. The point on the finger surface closest to the projected surface, which the camera can get, must be a tangent point between optical center C and the finger.

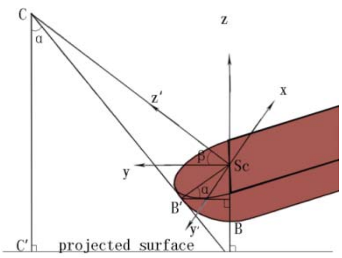

Figure.3. Relationship between finger model and optical center of camera

The coordinate system is rotated to simplify the calculation of those tangent points. The coordinate system is rotated around x-axis so that z'-axis in the rotated coordinate system will be collinear with line CS_C. The rotation matrix is expressed in equation (1):

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin\beta & -\cos\beta \\ 0 & \cos\beta & \sin\beta \end{bmatrix} \tag{1}$$

Assuming the coordinate value of tangent points in this new coordinate is (x,y',z') and they should fit the following equation:

$$x = r\frac{\sqrt{d^2-r^2}}{d}\cos\gamma, y' = r\frac{\sqrt{d^2-r^2}}{d}\sin\gamma, z' = \frac{r^2}{d} \tag{2}$$
$$\theta_1 \le \gamma \le \theta_2$$

γ is the included angle between x-axis and the line passing through the tangent point and the circle center of those tangent points. d is the length of CS_C. θ_1 and θ_2 are determined by the attitude of the finger model in this coordinate, as is shown in Fig.4.

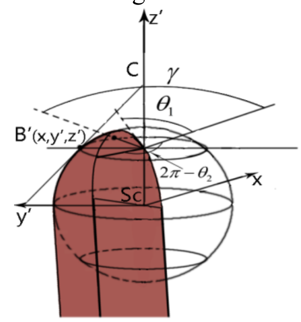

Figure.4. Relationship in the rotated coordinate

So the z coordinate value of tangent points in the xyz coordinate will be :

$$z = -r\frac{\sqrt{d^2-r^2}}{d}\sin\gamma\cos\beta + \frac{r^2}{d}\sin\beta \tag{3}$$
$$\theta_1 \le \gamma \le \theta_2$$

z gets its extreme value when $\gamma = \pm\frac{\pi}{2}$. In plane $S_C CC'$, the tangent point B' is the closest point to the projected surface among all finger points captured by camera. The distance between real closest point B and point B' in the z axis direction is $r - r\sin\alpha$. In the real implementation, the included angle between CC' and CB', α, is determined by the attitude of finger and greater than 60 degree, the radius of proposed finger model is about 10mm, so their distance will be smaller than 1.34mm. Therefore, point B' can be used to represent point B. The conclusion that intersection D can be used to approximate the distance from finger to the projected surface can be got by the analysis above, and it will have nothing to do with the thickness of finger. Detail calculation procedure will be shown in the next section.

III. Computing The Distance

Since the information of point B' and M is from the image data, the relationship between finger and its shadow in the real world will be corresponded to the image plane for the distance calculation, see Fig.5.

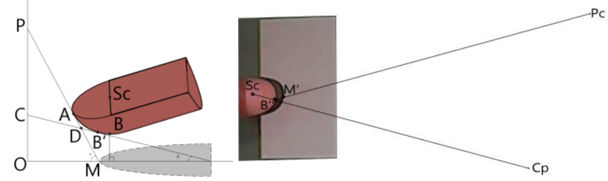

Figure.5. Corresponding relationship in image plane

S_C is the sphere center, C_P is the intersection between line CC' and image plane and P_C is the intersection between line PC and image plane. Since the finger point in image closest to the projected surface is in plane $S_C CC'$, the point will be a finger edge point on the line $C_P S_C$ in the image. So in the image plane, B'' is the image point of the finger point B'. The camera optical center C, point B' and B'' are on the same line in the real world. In order to calculate the intersection D between PM and CB', M should be in the plane consists of point P, C and B'. So in the image plane, the edge point of shadow on the line $P_C B''$, M', will be the image point of shadow point M. Assuming 2D position of point B'' and M' are $(x^C_{B''}, y^C_{B''})$ and $(x^C_{M'}, y^C_{M'})$ respectively, then the position of point M' in the projector's display panel can be derived by the equation (4):

$$K\left[x^P_{M'} \quad y^P_{M'} \quad 1\right]^T = H\left[x^C_{M'} \quad y^C_{M'} \quad 1\right]^T \tag{4}$$

$(x^P_{M'}, y^P_{M'})$ is the 2D position of point M' in the projector's display panel, which is as same as 2D position of point A in the projector's display panel. K is a scale coefficient, H is the homography[9] from camera to projector. After this homography transformation, projector optical center P, point M', point A and point M will be on the same line in the real world.

3D position of image point B'' in camera coordinate system is $(x^C_{B''}, y^C_{B''}, f_C)$, and the 3D position of point M' will be $(x^P_{M'}, y^P_{M'}, f_P)$ in the projector coordinate system. f_C and f_P are the focal length of camera and projector respectively and they can be got by their internal parameters. The rotation and translation matrix from camera coordinate system to projector coordinate system can be obtained by the procedure proposed in [7,8]. So the 3D position of image point B'' in projector coordinate system can be computed:

$$\left[x^P_{B''} \quad y^P_{B''} \quad z^P_{B''}\right]^T = R_{CP}\left[x^C_{B''} \quad y^C_{B''} \quad f_C\right]^T + T_{CP} \tag{5}$$

T_{CP} and R_{CP} represent the rotation and translation matrix from camera coordinate system to projector coordinate system respectively.

So the coordinate value of camera optical center C in the projector coordinate system is T_{CP}, equation of line CB' and PM will be:

$$CB': \quad \left[x^P \quad y^P \quad z^P\right]^T = \lambda_1 v_1 + q_1$$
$$PM: \quad \left[x^P \quad y^P \quad z^P\right]^T = \lambda_2 v_2 \tag{6}$$
$$v_1 = (\left[x^P_{B''} \quad y^P_{B''} \quad z^P_{B''}\right]^T - T_{CP}),\ q_1 = T_{CP},\ v_2 = \left[x^P_{M'} \quad y^P_{M'} \quad z^P_{M'}\right]^T$$

λ_1 and λ_2 are scale coefficients. So the intersection D can be determined by the solution of equation $\lambda_1 v_1 + q_1 = \lambda_2 v_2$. Since line CB' and PM are linear independence, so the solution to λ_1 and λ_2 is[6]:

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \|v_1\|^2 & -v_1^t v_2 \\ -v_2^t v_1 & \|v_2\|^2 \end{pmatrix}^{-1} \begin{pmatrix} -v_1^t q_1 \\ v_2^t q_1 \end{pmatrix} \tag{7}$$

So the 3D position of intersection D is $\lambda_2 v_2$. Assuming the normal vector and a point of the projected surface in projector coordinate system are n^P and p^P respectively, so the distance from intersection D to the projected surface is :

$$\text{distance} = \left|\frac{(\lambda_2 v_2 - p^P)^t n^P}{(n^P)^t n^P}\right| \times \left\|n^P\right\| \tag{8}$$

IV. Implementation of The Distance Calculation for The Projected Virtual Keyboard

Based on the model above, a new method is proposed to improve the accuracy of finger touch detection as illustrated in Fig.6. First of all, the internal and external parameters of projector and camera are obtained by the procedure proposed in [7,8] after they are mounted together and will not change afterwards. Then the procedure begins with calibrating the projected surface, homography[9] from camera to projector and the information of the projected surface will be got. Keyboard with big buttons is projected to test our algorithm after calibration. Distortion of any button in the projected keyboard means a potential touch event[5], and if the deformation is under a certain value, then the finger should be in the button area. These button images will be sent to the following procedures as the input images.

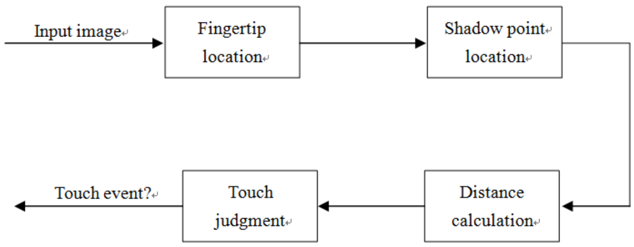

Figure.6. main structure of the algorithm

In order to acquire the position of the sphere center mentioned above, the finger should be detected first. A coarse to fine strategy is adopted. Since there must be a finger in the distorted button area and the skin color is more salient than the background button image in the Cr channel of the YCbCr color model, the OTSU threshold is used in the button area to get coarse finger area. Then a fine finger area is got by the region growing algorithm in the Munsell color space[10] because of the homogeneous color feature on the finger surface.

a b c

Figure 7. a) the origin input image; b) blue area is the coarse finger area detected by the otsu threshold in the cr channel. c) blue area is the fine finger area detected by the region growing algorithm in Munsell color space

The fingertip should be in the button area nearest to the projector. We get the edge of fingertip and fit it with a circle by the least squares algorithm, and center of the circle is also the sphere center of our finger model when ignoring the error resulted from the affine transformation. Then the position of S_C in image plane can be got. Position of C_P and P_C can be obtained by the internal and external parameters. Point B'' can be located by S_C, C_P and the fingertip edge, then line $P_C B''$ can be acquired. Through the feature in HSV color space[11], the shadow point M' on the line $P_C B''$ can be got. By the 3D line-line intersection equation (7), the approximate 3D coordinate of intersection D can be obtained and its distance to projected surface can be derived by equation (8), which can be regarded as the height of finger above projected surface and used for touch judgment.

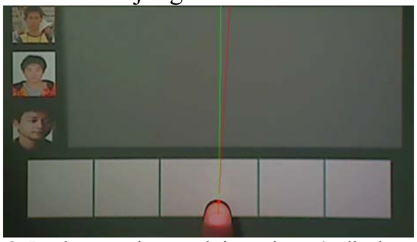

Figure.8. Implementation result in projector's display panel

V. EXPERIMENTAL RESULTS

In order to test the precision of the proposed method and verify whether it is influenced by the button size, the virtual keyboard with big buttons is used, see Fig.9. The height of buttons is 60mm, and the width of area 1, 2, 4 and 5 is 50mm, width of area 3 is 70mm. An Optoma PK301 DLP pico-projector with a resolution of 848×480 and a Logitech Pro9000 camera which provides 960×720 image at 30FPS are used in the experiment. The system is configured as follows, camera and projector are mounted together to make the connection of their optical centers perpendicular to the projected surface, pro-cam system is working at the distance of about 0.5m. The height of a prosthetic finger fixed on a platform is measured first, and the platform is moved to put the prosthetic finger into our projected keyboard buttons to measure its height afterwards. Each button is tested 35 times to make sure each position in the button area is measured and the data is shown in Table.1 when the prosthetic finger is 2.1mm above the projected surface. Precision is evaluated by the worst case, which is formulated as:

Precision= max (|real height- Max|, |real height- Min|).

Area	1	2	3	4	5
Max/mm	4.07	3.36	2.97	3.90	3.95
Min/mm	0.81	0.62	0.78	1.21	1.08
Precision/mm	1.97	1.48	1.32	1.80	1.85
Average/mm	2.50	2.28	2.09	2.34	2.53
Variance	0.72	0.54	0.44	0.57	0.67

Table.1. Measured results.

Error becomes bigger with the position further from the center button. The range of measured results varies from 0.62mm to 4.07mm, the accuracy can be about 2mm and will not be influenced by the button size. In [1], Wilson uses a Kinect to measure the height of finger above surface, and its worst case error is about 7mm when the system is 0.75m height above the projected surface. In [5], the precision is influenced by the button size. So we can get the conclusion that our method is better.

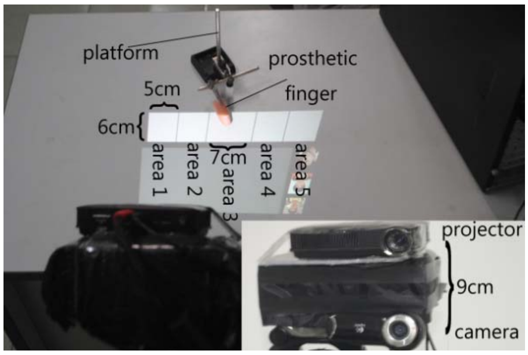

Figure.9. the hardware of our experiment

VI. CONCLUSION

In this paper, a new method for estimating the distance from finger to the projected surface by the geometric relationship of finger and its shadow based on the pro-cam system is introduced. By exploring the finger model we built, the point on finger closest to projected surface and its coplanar shadow point can be located in image plane, which is used to approximate finger height. Experiments show high accuracy of about 2mm when projector is about 50cm away from the projected image center. In future work, we will try to use our method to realize the goal of making any planar surface touch-sensitive no matter what projected.

REFERENCES

[1] A. D. Wilson, "Using a depth camera as a touch sensor", in ACM international conference on interactive tabletops and surfaces, 2010, pp 69–72.

[2] A. D. Wilson, "PlayAnywhere: a compact interactive tabletop projection-vision system", in Proceedings of the 18th annual ACM symposium on User interface software and technology, 2005, pp 83–92.

[3] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: wearable multitouch interaction everywhere", in Proceedings of the 24th annual ACM symposium on User interface software and technology, 2011, pp 441–450.

[4] J. Dai and R. Chung, "Making any planar surface into a touch-sensitive display by a mere projector and camera", in Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, 2012, pp 35–42.

[5] J. Hu, G. Li, X. Xie, Z. Lv and Z. Wang, "Bare-fingers Touch Detection by the Button's Distortion in Projector-camera System", IEEE Transactions on Circuits and Systems for Video Technology, Early Access Online, 2013.

[6] D. Lanman and G. Taubin, "Build your own 3D scanner: 3D photography for beginners", in ACM SIGGRAPH 2009 Courses, 2009, p 8.

[7] R. Legarda-Sa, T. Bothe, and W. P. Ju, "Accurate procedure for the calibration of a structured light system", Optical Engineering, vol 43, no 2, pp 464–471, 2004.

[8] S. Zhang and P. S. Huang, "Novel method for structured light system calibration", Optical Engineering, vol 45, no 8, pp 083601–083601, 2006.

[9] R. I. Hartley, "In defense of the eight-point algorithm", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol 19, no 6, pp 580–593, 1997.

[10] H. Wan, H. Hu, and Z. Shi, "Perceptually color clustering for image segmentation", in Proceedings of SPIE, 2001, vol 4554, pp 159–164.

[11] V. J. Tsai, "A comparative study on shadow compensation of color aerial images in invariant color models", Geoscience and Remote Sensing, IEEE Transactions on, vol 44, no 6, pp 1661–1671, 2006.