Depth Camera Based Real-Time Fingertip Detection Using Multi-view Projection

Weixin Yang, Zhengyang Zhong, Xin Zhang*, Lianwen Jin,
Chenlin Xiong, and Pengwei Wang

School of Electronic and Information Engineering,
South China University of Technology, Guangzhou, P.R. China
yang.wx@mail.scut.edu.cn, timothy7784@gmail.com,
eexinzhang@scut.edu.cn, eelwjin@scut.edu.cn, xcl_722@163.com,
eepwwang@163.com

Abstract. We propose a real-time fingertip detection algorithm based on depth information. It can robustly detect single fingertip regardless of the position and direction of the hand. With the depth information of front view, depth map of top view and side view is generated. Due to the difference between finger thickness and fist thickness, we use thickness histogram to segment the finger from the fist. Among finger points, the farthest point from palm center is the detected fingertip. We collected over 3,000 frames writing-in-the-air sequences to test our algorithm. From our experiments, the proposed algorithm can detect the fingertip with robustness and accuracy.

Keywords: Kinect, depth image, finger detection, fingertip detection, multi-view projection.

1 Introduction

Natural Human Computer Interface (N-HCI) has long been a heated research topic. Traditional finger detection methods are usually vision-based. When the finger is pointing out, finger area is relatively small and traditional vision-based fingertip detection methods face several challenges. The local maximum curvature [2] and the fingertip template matching [3, 4] are typical successful vision-based methods. These methods heavily relied on good segmentation results and are sensitive to noise and occlusion. Therefore, prior information is introduced, like the arm-shoulder structure [9] and hand skeletal model [10]. Such approaches can provide the finger joint location and estimate the hand pose, but it costs extra computational load. When the hand is close to the skin or skin-like color objects, they could be regarded as hand, which may affect the result. Furthermore, it is difficult to detect fingertip precisely with traditional vision-based methods, due to the variety of position and angle of fingers. The launch of Microsoft Kinect[1] camera provides new possibilities of natural human computer interaction. Several recent researches have made promising progress in HCI using depth information

[*] Corresponding author.

[1] http://www.microsoft.com/en-us/kinectforwindows/

M. Kurosu (Ed.): Human-Computer Interaction, Part V, HCII 2013, LNCS 8008, pp. 254–261, 2013.
© Springer-Verlag Berlin Heidelberg 2013

provided by Kinect [5-8]. In order to improve effectiveness and usability of device-free HCI systems, robust methods of fingertip detection become important.

In this work, we propose a real-time fingertip detection method via multi-view projection using depth information. Our method has the following main steps: extracting the user from the background, segmenting the hand from user's torso using depth threshold, employing multi-view projection to detect candidate pixels of the finger, finding the fingertip according to the distance between candidate fingertip pixels and the palm center. Our algorithm can robustly and effectively locate the fingertip of a pointing-out finger, which overcomes traditional vision-based problems like hand pose variation, skin-color similarity and hand self-occlusion. Moreover, the accurate fingertip detection framework enables us to further design a fingertip-based device-free writing system.

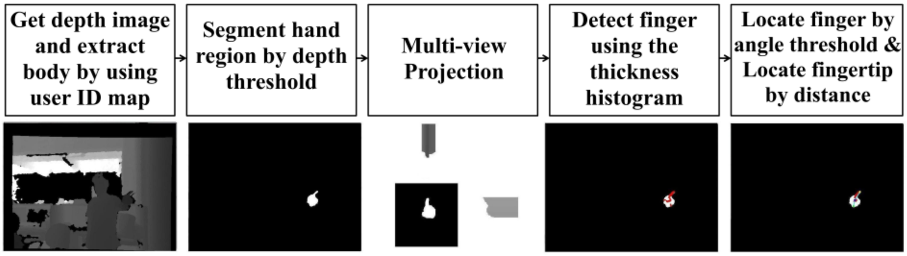
Get depth image and extract body by using user ID map
Segment hand region by depth threshold
Multi-view Projection
Detect finger using the thickness histogram
Locate finger by angle threshold & Locate fingertip by distance

Fig. 1. The flowchart of proposed algorithm

2 Proposed Framework

When the finger is pointing out, finger area is relatively small and traditional vision-based fingertip detection methods face several challenges. Our multi-view projection method converts frontal view depth image into top and side view depth images. By combining different views, we can effectively distinguish the fist and pointing-out finger using histograms of hand thickness from different views. Further, we employ a simple physical model to eliminate points that are wrongly classified as finger. Finally, the farthest finger point from the palm center is marked as the fingertip. The flowchart is shown in Fig.1.

2.1 Hand Segmentation

We get depth image captured by the Kinect camera and extract the user from the background using user ID map provided by OpenNI[2]. And then depth threshold is employed to segment hand from user's torso on the basis of the assumption that the writing hand is always in front of torso. By increasing the threshold of hand segmentation, we can get extra forearm pixels and mark the spatial center of them as forearm point for further process.

2.2 Finger Identification

For the proposed multi-view projection, we firstly convert pixels of depth image into 3D points [1] and project each point onto top and side view (Fig.2 (a)-(c)). When a finger is

[2] http://www.openni.org/Downloads/OpenNIModules.aspx/

pointing out, it is reasonable to assume that the finger is thinner than palm and wrist (Fig.2 (d)). Thickness histograms can roughly segment fist and finger. Thickness histograms are generated by calculating the number of pixels of each thickness value. According to our observation, there are two different histogram patterns. As shown in Fig. 3, without the pointing-out finger, thickness histograms from different views have only one peak; while in the other case, the histogram of one view might show only one peak but histogram of the other view would have two peaks, indicating the fist and finger. We select finger pixels accordingly for further process.

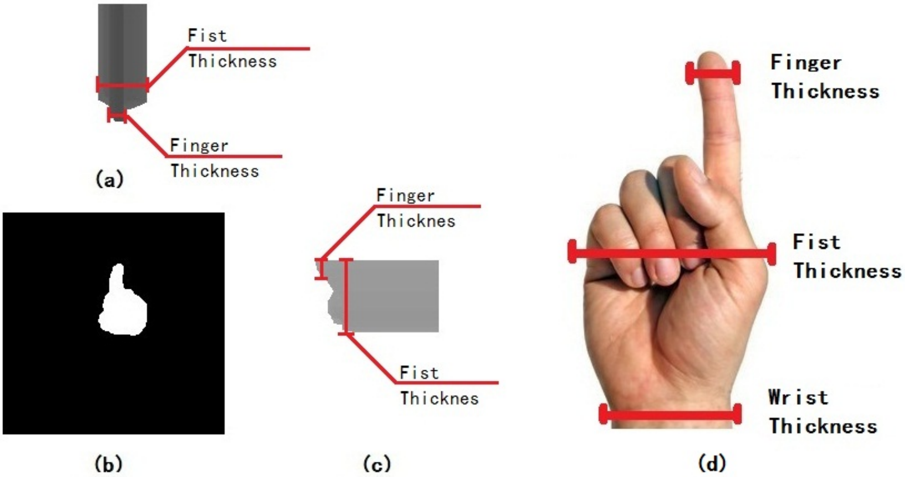

Fig. 2. Three views demonstration and hand thickness illustration：(a) top view (b) front view (c) side view (d) thicknesses of finger, fist and wrist

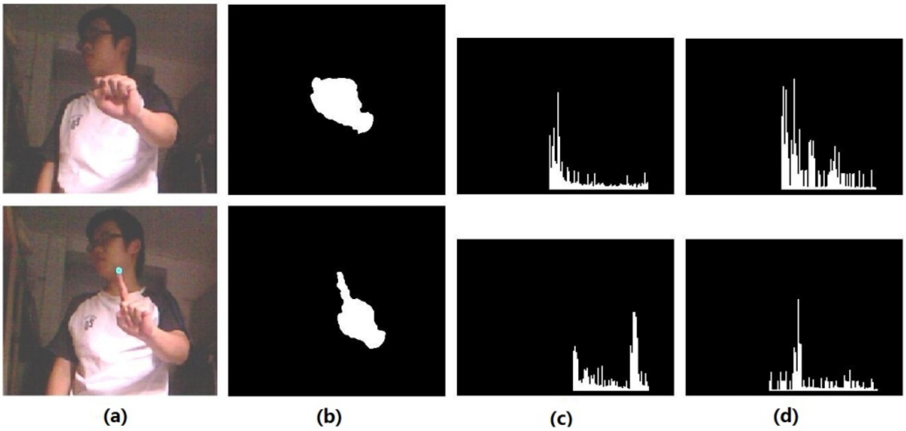

Fig. 3. Thickness histograms of two different situations: (a) RGB image, (b) Extracted hand, (c) Thickness histogram of side view, (d) Thickness histogram of top view

2.3 Fingertip Identification

According to our experiments, some fist points would be wrongly classified as finger points. This might lead to false fingertip detection. We take the following steps to remove these interference points. Considering the writing gesture of hand, the angle between finger and forearm is greater than 90° and this becomes a simple physical constraint for finger point classification, as shown in Fig.4. The palm center is defined as the average of palm pixels, marked as blue triangle in Fig. 4. By enlarging the threshold of hand segmentation, we can get extra forearm pixels and calculate the average position of these points, and then mark it as forearm point, as shown in fig.4. By connecting forearm point and candidate finger point with palm center, we have two vectors and define the angle between them as θ. Candidate points which formed angles less than 90° are regarded as interference points.

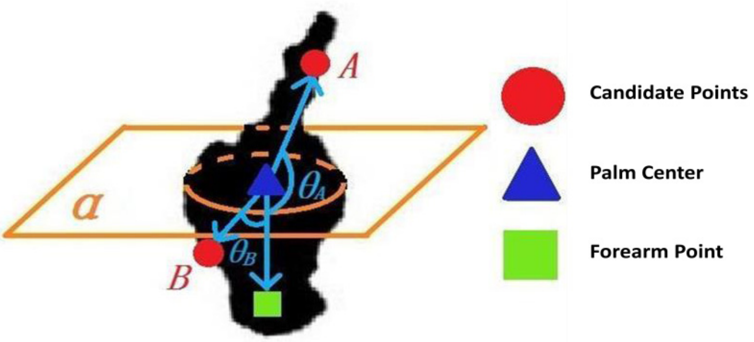

Fig. 4. Physical constraint for further finger point classification using angle-based threshold. (***A*** is the finger point and ***B*** is wrongly classified fist point.)

P(x,y,z) is a candidate point, then finger point can be identified by the formula below:

$$P(x, y, z) = \begin{cases} 1 & (\theta \geq 90°) \\ 0 & (\theta < 90°) \end{cases}$$

As illustrated in Fig.5, point A and point B are both candidate points and formed two angles: θA and θB. Angle θA is greater than 90° while θB is less than 90°, so point A is finger point and point B is interference point. Plane α is formed by 90°angles, and is employed to distinguish space of finger points from the space of interference points.

Among finger points, the farthest point from palm center is the detected fingertip. Detailed illustration and final fingertip detection results are shown in Fig.5. Fig.5 (a) is the color image with detected fingertip (yellow point) corresponding to the recognition result in Fig.5 (c). Fig.5 (b) is the depth image with candidate finger points (red points) generated by multi-view projection and thickness histogram. Fig.5 (c) indicates the process of fingertip identification.

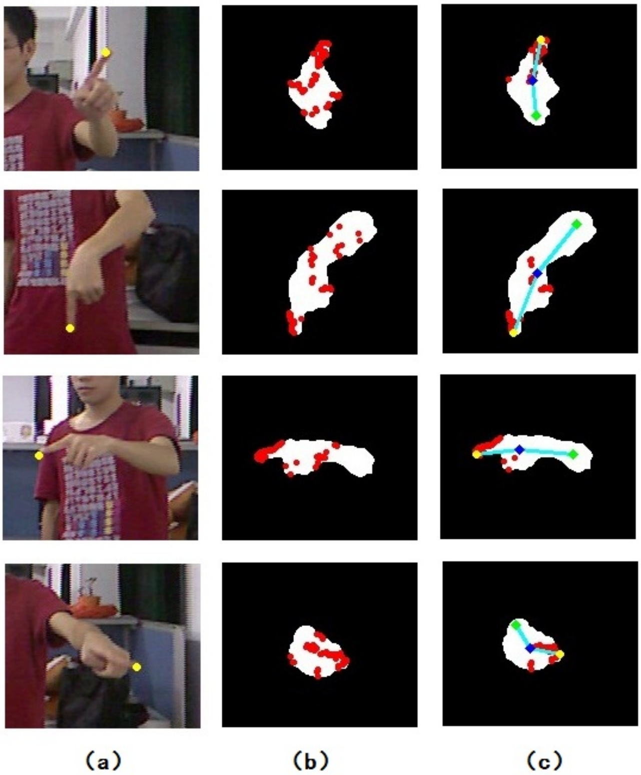

Fig. 5. Finger point classification and final fingertip detection result. (a) Color image (b) Candidate finger points (RED) (c) Results (RED for finger points, BLUE for palm center, GREEN for forearm point and YELLOW for fingertip).

3 Experimental Result and Discussion

Testing data was captured at 30 fps using OpenNI, with a resolution of 640×480. We focus on the hand and finger tracking, therefore the test subject was required to stand straight to the camera and write in the air with one finger out. We collected 3,010 frames (shown in Table 1) writing-in-the-air sequences, including writing in numbers, English and Chinese characters. We manually labeled these videos. The data includes location of fingertip and a bounding box of the hand in each frame.

Table 1. Overview of testing data set

	Uppercase	Lowercase	Character	Number	Total
Frame Number	507	485	1503	515	3,010

Detailed comparison is shown in Table 2. Curvature fitting [2] and template matching [3, 4] methods have been proposed for the vision-based fingertip detection. Both methods face common 2D vision challenges like noisy and clutter background, similar skin color and hand pose variation. Nearest method uses the depth image and aims at finding the pixel with the smallest depth value. It has largest pixel error, because fingertip is not always the closest point to the camera. The cluster method [1] is the most recent and relevant work using depth data. It has relatively few pixel errors but its performance is unstable. Pixel error greatly increases when dealing with complex inputs like Chinese character. The proposed method has the best fingertip detection accuracy rate of 93.69% (within 10 pixels) and an average deviation of 4.76 pixels with a stable performance for various inputs.

Table 2. Accuracy comparison within 10 pixels (<10) and 5 pixels (<5)

Method	Nearest	Curvature[2]	Template[3,4]	Cluster[1]	Proposed
<10	71.05%	81.04%	88.59%	90.41%	**93.69%**
<5	57.48%	52.24%	64.65%	**67.58%**	64.82%

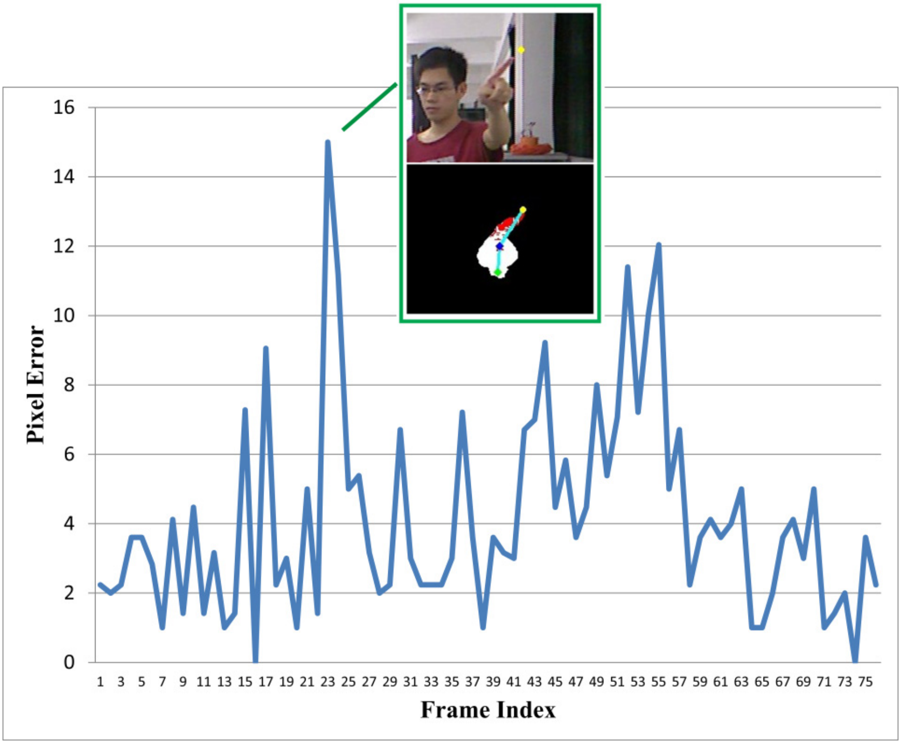

Fig. 6. Error plot of every frame in one sequence. The color and depth images of the frame with the highest pixel error are shown.

An example of distribution of pixel errors in a sequence is shown in Fig. 6. The errors are distributed on an average of 4.15 pixels. The color image and depth image of the frame with the most error are shown. Due to the time synchronization issue between color and depth images when the user is writing fast, the detected fingertip (yellow point) from the depth image is different from the fingertip in the color image. Skin color detection is an available method to solve this issue. The finger pixels that are not skin-colored can be removed from the fingertip candidate pixels.

4 Conclusion

This paper proposes a real-time fingertip detection method that overcomes traditional vision-based problems like hand pose variation, skin-color similarity and hand self-occlusion. We employed depth threshold to segment hand from user's torso, and then use the depth information of front view to generate depth map of top view and side view. Due to the difference between finger thickness and fist thickness, thickness histogram is employed to segment the finger from the fist, and the farthest finger point from the palm center is marked as the detected fingertip. According to the result of our experiments, the proposed algorithm can detect the fingertip with robustness and accuracy.

Acknowledgement. This work is supported by National Undergraduate Innovative and Entrepreneurial Training Program (No.111056115), National Science and Technology Support Plan (2013BAH65F01, 2013BAH65F04), GDSTP (No. 2012A010701001), Natural Science Foundation of China (No. 61202292) and Fundamental Research Funds for Central Universities of China (No. 2012ZM0022).

References

1. Feng, Z., Xu, S., Zhang, X., Jin, L., Ye, Z., Yang, W.: Real-time fingertip tracking and detection using Kinect depth sensor for a new writing-in-the air system. In: International Conference on Internet Multimedia Computing and Service (ICIMCS 2012), pp. 70–74 (2012)
2. Malik, S., Laszlo, J.: Visual touchpad: a two-handed gestural input device. In: Proceedings of international Conference on Multimodal Interfaces, pp. 289–296. ACM (2004)
3. Jin, L., Yang, D., Zhen, L., Huang, J.: A novel vision based finger-writing character recognition system. Journal of Circuits, Systems, and Computers (JCSC) 16(3), 421–436 (2007)
4. Crowley, J., Berard, F., Coutaz, J.: Finger tracking as an input device for augmented reality. In: International Workshop on Gesture and Face Recognition, pp. 195–200 (1995)
5. Minnen, D., Zafrulla, Z.: Towards robust cross-user hand tracking and shape recognition. In: IEEE International Conference on Computer Vision Workshops, pp. 1235–1241 (2011)
6. Pugeault, N., Bowden, R.: Spelling it out: Real-time asl fingerspelling recognition. In: IEEE International Conference on Computer Vision Workshops, pp. 1114–1119 (2011)
7. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, p. 7 (2011)

8. Tang, Y., Sun, Z., Tan, T.: Real-time head pose estimation using random regression forests. In: Sun, Z., Lai, J., Chen, X., Tan, T. (eds.) CCBR 2011. LNCS, vol. 7098, pp. 66–73. Springer, Heidelberg (2011)
9. Wu, A., Shah, M., Da Vitoria Lobo, N.: A virtual 3D blackboard: 3D finger tracking using a single camera. In: International Conference on Automatic Face and Gesture Recognition, pp. 536–543 (2000)
10. Kang, S., Nam, M., Rhee, P.: Color based hand and finger detection technology for user interaction. In: International Conference on Convergence and Hybrid Information Technology, pp. 229–236 (2008)