ISSN:0975-9646

Fingertip Tracking and Hand Gesture Recognition by 3D Vision

De Gu[#1]

[#] *Key Laboratory of Advanced Process Control for Light Industry Ministry of Education, Jiangnan University Lihu avenue, Wuxi, China.*

***Abstract*—This paper introduces an algorithm to track palm and fingertips based on images with depth data in real-time. Some meaningful hand gesture is then recognized by the detected palm and fingertips. The images with depth information are first captured by a Kinect camera. Then foreground and background are separated to pick up the potential hand. The fingertips are then detected by the curvature of the hand boundary. Finally, we tested several meaningful hand gesture, and the result is inspiring.**

***Keywords*— depth image data; fingertip detection; posture recognition; human-machine interaction Introduction**

I. INTRODUCTION

Human-machine interaction by body language is popular in recent years. For example hand trajectory is reported to be able to interact with media player [1]-[3]. Hand gesture recognition systems by vision is developing recently[4][5]. However, hand gesture recognition by vision is very dependent by illumination and affected badly by complicated background. Another type of hand gesture recognition system is based on wearable equipment, such as data glove[6]-[10]. A data glove provides precise position data, which certainly contributes a lot for later gesture recognition. Furthermore, the data are usually labeled with finger names, if the glove is worn correctly. But it does have some drawbacks. It is not comfortable wearing something on hand. And the battery recharge for the wearing equipment is also a trouble.

Recognition algorithms based on depth information is becoming popular with the development of 3D cameras. Bergh et al. introduced an algorithm for hand gesture recognition using a 3D camera, which gets the depth information via Time of Flight (TOF)[11]. However, Bergh's gesture recognition algorithm are not widely used, because 3D cameras based on TOF technique are very expensive. Microsoft's Kinect provides a much cheaper way of getting 3D vision by using light coding technique. Yang et al. designed a palm trajectory tracking system based on Kinect, and then the system is used in a media player application[1]. This system proves that it is possible to recognize a hand gesture by a contactless infrared camera. It is not able to detect fingertips, however, so a more detailed finger level gesture cannot be recognized. Tunkakurn et al. designed a system for fingertip detection based on the skeleton information provided by Kinect, and then used it in a medical software[12]. But this system is dependent on skeleton detection. Nothing is detectable if the skeleton is not fully detected, e.g. some part of the person is outside the visual field although the hand is fully in.

This paper proposes an algorithm to recognize fingertips level gesture by depth data (without skeleton information), which provided by an infrared camera on Kinect. Fingertip tracking and several meaningful hand gesture are tested to check whether the algorithm is applicable.

II. THE ALGORITHM

A. Environment and assumptions

The resolution of the infrared camera is 640 pixels × 480 pixels. The infrared camera is put on a level desk. The hand is the nearest visible object, i.e. the hand should be stretch toward the camera. The distance between the camera and the hand to be detected is no less than 40 cm (or otherwise some part of the hand may frequently move out of vision) and no more than 2 m (or otherwise the hand is too small to detect fingers).

B. Separation of foreground and background

The task is focused on hand, so the first step is to remove the background.

The infrared camera provides all pixels in vision with depth information, but only the nearest part includes the potential object, the hand. So let us find the pixel P_0 with smallest depth value z_0 . Then remove every pixel P whose depth value z is much further than P_0 , i.e., remove P if $z > z_0 + \sigma$, where σ is the threshold. In this paper, σ=15cm , because when trying to make some gesture to the camera, the distance between P_0 (usually, a fingertip) and the wrist along z-axis never exceeds 15 cm. Thus the foreground, hand P_h , is separated from the background. If such process repeats once again, the 2nd foreground, which is probably another hand, is picked out. Fig. 1 shows the hands in vision, where the blue dots are the geometrical centers of the hands. The pixels at the boundary of hand, P_b , is very important for further study. The boundary consisted by the pixels that is in the foreground itself with at least one adjacent pixel in the background. Fig. 2 shows that a nearest hand and its boundary is detected either from a complicated background, or in a dark environment. In this step, the boundary pixels are organized in a sequenced array.

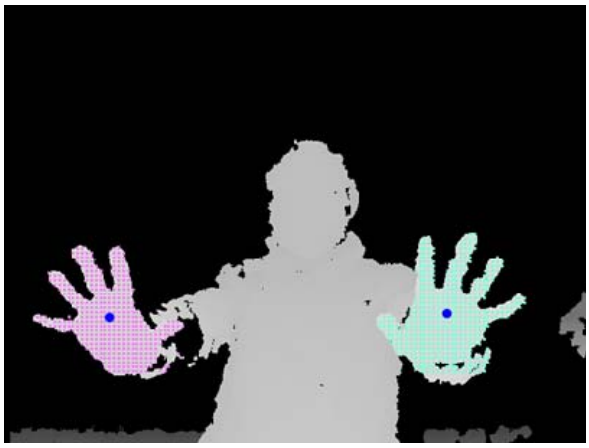

Fig. 1. two hands detected

Fig. 2. the boundary of a hand

C. Searching for the center of palm

The center of palm is indispensable in the steps afterwards. Initially, the geometrical center of the hand is regarded as the center of palm. However, the geometrical center varies a lot when fingers stretching out without any movement of the palm. In this paper, the center of the maximum inscribed circle of the hand boundary is regarded as the center of the palm, P_c. It is much stabled than the geometrical center. In this algorithm, the geometrical center is the initial value of the searching process for P_c.

D. Fingertip detection

The fingertip is a pixel where the hand boundary curves outward most locally. Fig. 3 shows how to detect a fingertip. It is a searching process in the sequenced hand boundary array, which is prepared in separation of foreground and background.

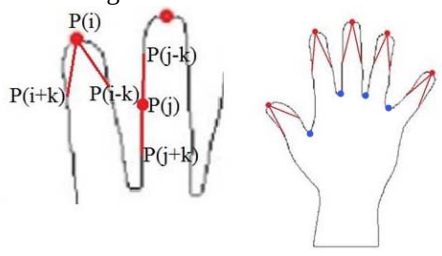

Fig.3. Fingertip detection

For any pixel $P(i)$ of the boundary array, there is a pixel $P(i+k)$ that is indexed k after $P(i)$, and $P(i-k)$ indexed k before $P(i)$. Denote $\vec{a}_i = \overrightarrow{P(i)P(i-k)}$, $\vec{b}_i = \overrightarrow{P(i)P(i+k)}$, and $\alpha_i = \angle P(i-k)P(i)P(i+k)$. The value of α is calculated by equation (1)

$$\alpha_i = \arccos \frac{\vec{a}_i \times \vec{b}_i}{\|\vec{a}_i\| \|\vec{b}_i\|} \qquad (1)$$

If $\alpha_i < \delta$ and it is a local minimum, then $P(i)$ is a fingertip candidate, where δ is threshold. In this paper, $\delta = 40°$.

There are some pixels, which is actually a fingerweb, the blue dots shown in Fig. 3, may be faulty detected as fingertip candidates. For a fingertips, $P(i)$ is further away from the palm center than both $P(i+k)$ and $P(i-k)$. While for a fingerweb $P(i)$ is nearer.

In the process discussed above, one parameter, k, has great impact on the result of finger detection. Either too big or too small k may lead failure. A proper k should guarantee both that $P(i+k)$ and $P(i-k)$ are on the boundary of a same finger if $P(i)$ is a fingertip, and that k is big enough to make $\vec{a}_i$ and $\vec{b}_i$ reasonable. As the hand may move toward or away from the camera, k is determined by the length of the hand boundary array, L. A lot of video frames are tested. And the result is shown in table 1.

TABLE 1.
THE RELATION BETWEEN OF k / L AND CORRECT DETECTION

k / L	Correct Detection
1%	0
3%	34%
4%	81%
5%	96%
6%	90%
7%	73%
9%	42%

Thus $k = L \times 5\%$ based on the test result.

E. Fingertip tracking and gesture recognition

This step records the trajectory of the fingertips. There is a tracking result shown in Fig. 6 in later paragraph.

The gesture recognition is based upon the number of detected fingertips and the distance between the fingertips.

III. RESULT AND DISCUSSION

Fig. 4 shows that the stretching index finger is detected. Interaction with computer is shown in Fig. 5: when the index finger moves onto a UI label, the color of the label changes. Fig. 6 shows that the index fingertip is tracked, and it writes a small letter “a”.

Fig. 4. A fingertip detected

Fig. 5. Interaction with UI label

Fig. 6. Tracking the fingertip

Table 2 shows that the finger detection is almost independent from illumination and background. The number of stretching fingers has slight impact on correctness.

TABLE II.
CORRECT DETECTION VS NUMBER OF FINGERS.

Number of fingers	Correct frames / Total frames (correctness in percentage)		
	Good illumination	Complicated background	Poor illumination
1	211/219(96.3)	187/193(96.9)	179/184(97.3)
2	197/205(96.1)	188/196(95.9)	181/193(93.4)
3	185/193(95.6)	236/251(94.0)	211/218(96.8)
4	245/259(94.6)	194/206(94.2)	203/215(94.4)
5	227/240(94.6)	197/211(93.4)	212/226(93.8)
Total	1065/1116(95.4)	1002/1057(94.8)	986/1036(95.2)

Finally, we tries to recognize 8 hand gesture shown in Fig. 7, and table 3 is the result.

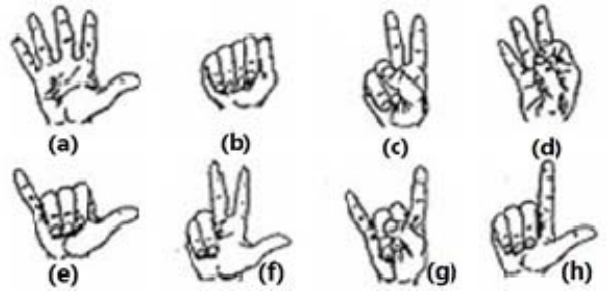

Fig. 7. Eight different hand gestures.

TABLE III.
CORRECT RECOGNITION FOR EACH GESTURE

Tested gesture	Correct frames / Total frames (correctness in percentage)
a	201/211(95.3)
b	198/201(98.5)
c	185/199(93.0)
d	190/202(94.1)
e	211/235(89.8)
f	189/203(93.1)
g	227/239(95.0)
h	185/198(93.4)
total	1586/1688(94.0)

The worst recognized gesture is (e), where the closed indexed finger is faultily detected as a stretching finger sometimes. Nevertheless, these gesture is well recognized.

IV. CONCLUSION

In this paper, an algorithm for fingertip detection is introduced. The algorithm uses 3D image frames provided by an infrared camera on Kinect. The foreground and the background is separated first. Then the palm center and the hand boundary is identified. After that, the fingertip is detected based the curvature of the hand boundary. Finally, the algorithm is tested by tracking and hand gesture recognition. The test result is good and inspiring. Dynamic gesture recognition is expected in the near future.

REFERENCES

[1] Yang C, Jang Y, Beh J, et al.Gesture recognition using depth-based hand tracking for contactless controller application. Proceedings of IEEE International Conference on Consumer Electronics ICCE)IEEE, 2012:297-298.

[2] Silanon K, Suvonvorn N.Hand motion analysis for Thaialphabet recognition using HMM.International Journal of Information and Electronics Engineering, 2011(1):65-71.

[3] Holte M B, Moeslund T B, Fihl P.View-invariant gesture recognition using 3D optical flow and harmonic motion context.Computer Vision and Image Understanding, 2010, 114(12):1353-1361.

[4] Chu S, Tanaka J.Hand gesture for taking self portrait. Human-Computer Interaction Techniques and Environments. Berlin Heidelberg:Springer, 2011:238-247.

[5] Wachs J P, Kölsch M, Stern H, et al.Vision-based handgesture applications.Communications of the ACM, 2011, 54(2):60-71.

[6] Harrison C, Benko H, Wilson A D.OmniTouch:wearable multitouch interaction everywhere. Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. ACM, 2011:441-450.

[7] Kenn H, Megen F V, Sugar R.A glove-based gesture interface for wearable computing applictions. Proceedings of the 4th International Forum on Applied Wearable Computing(IFAWC), 2007:1-10.

[8] Vicente A P, Faisal A A.Calibration of kinematic body sensor networks:Kinect-based gauging of data gloves“in the wild”. Proceedings of IEEE International Conference on Body Sensor Networks(BSN)IEEE, 2013:1-6.

[9] Reddy K, Samraj A, Rajavel M, et al.Suitability analysis of gestures for emergency response communication by patients, elderly and disabled while using data gloves. Proceedings of the 1st WSEAS International Conference on Information Technology and Computer Networks(ITCN’12), 2012.

[10] Kumar P, Rautaray S S, Agrawal A.Hand data glove:A new generation real-time mouse for Human-Computer Interaction. Proceedings of the 1st International Conference on Recent Advances in Information Technology (RAIT).IEEE, 2012:750-755.

[11] Van den Bergh M, Van Gool L.Combining RGB and ToF cameras for real-time 3D hand gesture interaction. Proceedings of IEEE Workshop on Applications of Computer Vision(WACV) IEEE, 2011:66-72.

[12] Tuntakurn A, Thongvigitmanee S S, Sa-Ing V, et al.Natural interactive 3D medical image viewer based on finger and arm gestures Proceedings of the 6th Biomedical Engineering International Conference (BMEiCON) IEEE, 2013:1-5.