Fingertips Tracking Based Active Contour for General HCI Application

Kittasil Silanon[1] , Nikom Suvonvorn[1]

[1] Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla, Thailand 90112

{kittasil.silanon, nikom.suvonvorn}@gmail.com

Abstract. This paper presents a real time estimation method for 3D trajectory of fingertips. Our approach is based on depth vision, with Kinect depth sensor. The hand is extracted using hand detector and depth image from sensor. The fingertips are located by the analysis of the curvature of hand contour. The fingertips detector is implemented using concept of active contour which combine the energy of continuity, curvature, direction, depth and distance. The trajectory of fingertips is filtered to reduce the tracking error. The experiment is evaluated on the fingers movement sequences. Besides, the capabilities of the method are demonstrated on the real-time Human-Computer Interaction (HCI) application.

Keywords: Fingertips Detection and Tracking, Hand Posture Estimation, Human-Computer Interaction (HCI).

1 Introduction

Hand gesture recognition has been a popular research in recent year. It provides more natural human-computer interaction. Many researches in this field related to real-time hand gesture recognition are proposed. Most of these system use trajectories of hand motion to recognize the commands [1,2,3,4,5]. However, the most important aspect of hand gesture recognition is to recognize commands accurately can be done with the accurate position of fingertips. Thus, some works have introduced on contours-based method for 2D fingertips tracking [6,7,8,9]. However, this approach cannot track fingertips robustly and usually design to track only stretched fingertips. Other systems for tracking the fingertips are using finger kinematic model [10,11,12] which search for the special form of the fingertips. These systems can work robustly. However, the computation cost is still too high. Using stereo vision has proposed to analyze the 3D fingertip positions [13,14,15,16]. The most problems in 3D fingertips are failure tracking when the fingertips are bending into the palm or overlapped each other. Therefore, this paper presented system to deal with such situations by using the depth image from Kinect. The hand region is segmented from the depth image and initial hand features are detected. The 3D fingertips are tracked using concept of active contour which occurs from internal and external energy that use features of continuity,

adfa, p. 1, 2011.

curvature, depth and distance from candidate fingertips that are detected at hand region in each frame. The energy represents the possible of candidate fingertips to be the fingertip in next frame. The tracking experiments are tested on basic finger movement. In addition, we develop an HCI application based on the fingertips tracking result. The rest of paper is organized in four sections by the following: initial hand segmentation, finger detection and tracking, experimentation results and conclusion respectively.

2 Initial Hand Segmentation

2.1 Hand Segmentation

From our previous work [5], we proposed hand detector by using object detection method which provides accurate result (Fig.1.a). The hand detector will be used to search hand's region in image. From experimentation, we found that hand detector failed when other parts of body such as face, arm move close to hand's region (Fig.1.b).

(a) (b)

Fig. 1. Hand segmentation: (a) hand detector (b) wrong segmentation.

Therefore, depth image (Fig.2.c) is used in the system. The depth is a 3D position vector (x,y,z) which obtained from the Kinect camera, where the x and y are the rows and columns in an image and z is the depth readings that are stored in the pixels, for detecting how pixel far away from camera.

(a) (b) (c)

Fig. 2. Depth information: (a) complex background (b) depth image (c) hand's region.

The depth can be solved the problem of complex background (Fig.2.a) by setting initial depth value from hand detector to remove any object behind the hand. Thus, our system takes depth information to extract only hand's region in image. After extraction hand's region, the initial hand's features will be estimated. For example, hand center, fingertips position, palm size etc. All initial hand features are used to be reference value to compare changing of hand gesture in command recognition system.

2.2 Hand Center Position

We obtained the center point of hand's region that can be easily computed from the moments of pixels in hand's region, which is defined as:

$$X_c = \frac{M_{10}}{M_{00}}, \; Y_c = \frac{M_{01}}{M_{00}} \quad (M_{ij} = \sum_x \sum_y x^i y^j I(x,y)) \tag{1}$$

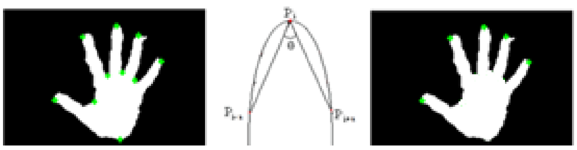
In the above equations, *I(x,y)* is the pixel value at the position *(x,y)* of the image, x and y are range over the hand's region. The center point is calculated as *Xc and Yc* (Fig.3.a). The palm size is defined as the distance between the center point and the closest pixel on hand contour (Fig.3.b).

(a) (b)

Fig. 3. Initial hand feature: (a) hand center (b) palm size.

2.3 Fingertip Positions

Since the user is required to initialize the system by producing the pose of an "open" right hand facing the camera. Therefore, it is simple to locate fingertips from curvature of boundary point of hand's region. However, it may not be necessary to consider all the boundary points of hand's region. Thus, we use the polygon approximation method to extract key point [18] and is stored in a new series of key point P1,…,Pn (Fig.4.a). Each key point Pi has two parameters, the angle (θ) and slope. The angle can be estimated by using k-curvature [6] which calculates the angle of key point by two vectors [*P(i-k)P(i)*] and [*P(i)P(i+k)*] with the same range (*k*) (Fig.4.b). The key point, with curvature value is in the threshold and slope is positive, is an initial fingertip (Fig.4.c).



(a) (b) (c)

Fig. 4. Fingertips position: (a) key point (b) curvature calculation (c) fingertip points.

The initial positions of the five fingertips are detected. Each of them will be given a label that corresponds to the thumb, index, middle, ring and pinky fingers (f_j). As we control initial open hand posture to frontal view. We can simply label fingers based on sorting the five points by clockwise arranging around palm center. Nevertheless, it can only detect fingertip in open hand. Thus, tracking method is used for tracking fingertips which can be changed position at all time in hand gesture sequence.

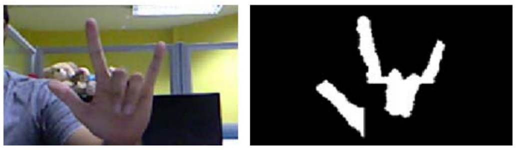
3 Finger Detection and Tracking

3.1 Finger Locations

Most movements in hand gesture are finger movements (Fig.5.a). Therefore, we defined two conditions to segment finger locations (Fig.5.b). For a stretching finger, use distance condition and a bending finger, use depth conditions follow the rules:

$$D(p, C_0) > k.R \quad || \quad |Z(p) - Z(C_0)| < \tau_D \tag{2}$$

Where k is a scaling factor, D (p, C_0) is distance between considering pixel and hand center, Z is depth value and τ_D is a pre-defined depth threshold.

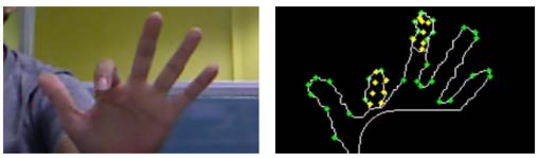


(a) (b)

Fig. 5. Finger locations: (a) origin hand posture (b) extraction finger location.

3.2 Candidate Fingertips

We define searching area to locate the candidate fingertip positions (Cf_i). We assume that fingertip positions should be points on hand contour. Therefore, it may not be necessary to consider all points on hand contour. Hence, the polygon approximation algorithm is used again to find candidate fingertips. Finger.6.b shows points on contour for stretching finger. In addition, candidate fingertip positions can be found by assuming that they are closest to the camera in each finger region. Thus, we use depth to find point which has minimum depth to be candidate fingertips. Fig. 6.b shows these points inside contour.



(a) (b)

Fig. 6. Candidate fingertips: (a) origin hand posture (b) candidate fingertips.

3.3 Fingertip Tracking

The tracking of the fingertip position between successive frames is built by concept of active contour [18,19]. The possible candidate fingertips of each fingertip will be assigned energy and then the maximum energy point is chosen to be the fingertip in the next frame. The discrete formulation of energy function which can be written as:

$$E(Cf_i) = E_{internal}(Cf_i) + E_{external}(Cf_i) \quad (3)$$

The energy for each candidate fingertips can be decomposed into two basic energy term. $E_{internal}$ represents the internal energy of the candidate due to bending or stretching of finger and $E_{external}$ is the external constraint introduced by user. The internal Energy of candidate fingertips is defined as:

$$E_{internal} = E_{continuity} + E_{curvature} + E_{depth} + E_{direction} \quad (4)$$

The $E_{continuity}$, It forces the candidate fingertip points to be continuous, because the fingertip should not change much from the current point to the next one. Therefore, this term tries to keep point which has appropriate distance between the average distance ($\bar{d}$) and candidate fingertip point. The form for $E_{continuity}$ is the following:

$$E_{continuity} = \bar{d} - \left\| f_j - Cf_i \right\|^2 \quad (5)$$

The $E_{curvature}$, this term will find smoothness of the candidate fingertips by considering contour curvature between the two vectors, A = ($x_i - x_{i-k}$, $y_i - y_{i-k}$) and B = ($x_{i+k} - x_i$, $y_{i+k} - y_i$), k is constraint. If the candidate fingertips have the curvature value that fall under a threshold, these points will be kept to be possible candidate fingertips. The formula for $E_{curvature}$ is given by:

$$E_{curvature} = \cos^{-1} \frac{\mathbf{A}.\mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} \quad (6)$$

Because we have use property of depth, the E_{depth} has been established. The E_{depth} is distance between candidate fingertips to camera that represent in 16-bit depth data units of millimeter. As we have assumed, the fingertips should be found at the closest point to the camera. Therefore, the E_{depth} becomes maximum value when the candidate fingertips get close to a camera. The closest point will be given more priority than the rest points in order of depth. As we mentioned previously, The $E_{external}$ is the external constraint. Thus, in our system, the distances from all fingertips to considering candidate fingertip in image are used to describe about the $E_{external}$. For instance, if we are considering the movement of index fingertip, the distances from other fingertips to considering candidate fingertip point are equivalent to the external energy which will be used to estimate suitability for choosing the considering candidate fingertip to be the index fingertips. The distance from index fingertip to considering candidate fingertip should be shorter than other fingertip. On the other hand, if the distances from other fingertips are shorter than index fingertip distance, the considering candidate fingertip should be assigned to another fingertip. The energy function represents the importance of candidate fingertips relative to each fingertip. The candidate fingertip with maximum energy is selected to be new location of fingertip in next frame. In order to reduce the tracking error due to losing depth value, a simple low-pass filter is applied for the smoothness trajectory of fingertip tracking. The average point between selected point and current fingertip point will be estimated.

4 Experimentation Result

4.1 Fingertips Tracking Precision

We evaluate the fingertip tracking precision on basic finger movements between the tracked fingertips and the ground truth. We have defined the ground truth using the end point contour of each finger. The basic finger movements have included five sequences (Fig.7), bending finger (seq.1), moving finger into the palm (Seq.2), for crossing finger (Seq.3), hand movement (up, down, left and right) (Seq.4). 45^0 hand rotation (counterclockwise and clockwise) (Seq.5). Each sequence is tested at 10 rounds. Table1 shows the precision in terms of the Euclidean distance.

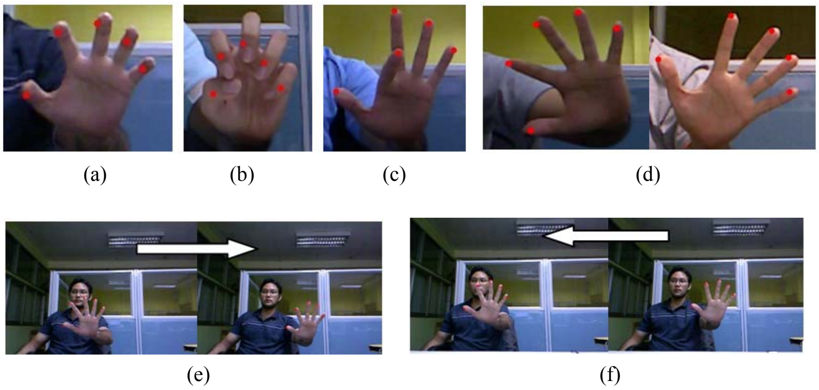


Fig. 7. Basic fingertip movements: (a) bending finger (b) moving finger into the palm (c) crossing finger (d) 45^0 hand rotation (e-f) hand movement.

Table 1. Fingertips Tacking Precision

Seq.	Tracking Precision(Pixel)					Avg.
	Thumb	Index	Middle	Ring	Pinky	
1	7.94	7.65	8.50	9.25	8.51	8.37
2	25.35	14.28	14.96	23.12	15.69	18.68
3	7.47	6.06	5.12	5.60	5.10	5.87
4	5.60	4.16	3.06	5.05	5.77	4.72
5	4.17	3.44	5.80	3.76	7.19	4.87

From experimentation result, we found that the tracking cases for crossing, rotation and movement (Seq.3, Seq.4 and Seq.5) give a good precision (≈ 4-6 pixels error) because the candidate fingertips obtained from polygon approximation algorithm are quite well located to the fingertip ground truth. But the tracking cases for bending and moving into palm (Seq.1 and Seq.2) give less precision (≈ 9-20 pixels error) due to the imperfect depth data received from Kinect that is not suitable for near mode function. Hence, depth data of finger may lose in some frames. From these factors, there might produce some errors in the fingertips tracking process.

4.2 Human-Computer Interaction Application

We carried out experiments on a general HCI Application that used as the commands to interface with Window Media Center on Window 7. For example, pressing up, down, left, right button. You can see video showing the real-time interactive at: http://www.youtube.com/watch?v=OnQra4We-4o

Fig. 8. Our method is implemented in Window7 for controlling Window Media Center.

5 Conclusion

Fingertips and palm positions are significant features for hand gesture recognition. The most of previous works cannot track 3D fingertip positions because the complexity of finger movement. In this paper, we present the method to deal with these issues by using depth data feature for correctly hand segmentation and apply concept of active contour to track fingertips over finger movement. Our method shows good performance in term of real-time and also has capability to expansion to Human-Computer Interaction application. However, our method still has some limitation in tracking fingertips. The fingertips tracking procedure fails if the neighborhood candidate fingertips are lost, which is the case of the finger movements are too fast. For the possible applications, our system can be combined with other Human-Computer Interaction applications, such as finger-spelling process, robot controlling, visual input device and etc.

6 Acknowledgments

We would like to thank the National Research University Project of Thailand's Office of the Higher Education Commission for financial support.

References

1. Deng, J.W., Tsui, H.T.: An HMM-based approach for gesture segmentation and recognition. In: 15th International Conference on Pattern Recognition Proceedings, vol. 3, pp. 679-682 (2000)
2. Ho-Sub Yoon, Jung Soh, Younglae J. Bae, Hyun Seung Yang: Hand gesture recognition using combined features of location, angle and velocity. In: Pattern Recognition, vol. 34, pp. 1491–1501 (2001)

3. Feng-Sheng Chen, Chih-Ming Fu, Chung-Lin Huang: Hand gesture recognition using a real-time tracking method and hidden Markov models. In: Image and Vision Computing, vol. 21, pp. 745–758 (2003)
4. Elmezain M., Al-Hamadi: Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Model. In: IEEE International Symposium on Signal Processing and information Technology, pp.1192-1197 (2007)
5. Kittasil Silanon, Nikom Suvonvorn: Hand Motion Analysis for Thai Alphabet Recognition using HMM. In: International Journal of Information and Electronics Engineering vol. 1, pp. 65-71 (2011)
6. Wei Du, Hua Li: Vision based gesture recognition system with single camera. In: 5th International Conference on Signal Processing Proceedings WCCC-ICSP, vol. 2, pp.1351-1357 (2000)
7. Antonis A. Argyros, Manolis I. A. Lourakis: Vision-based interpretation of hand gestures for remote control of a computer mouse. In: Computer Vision in Human-Computer Interaction, pp. 40-51 (2006)
8. Ko-Jen Hsiao, Tse-Wei Chen, Shao-Yi Chien: Fast fingertip positioning by combining particle filtering with particle random diffusion. In: IEEE International Conference on Multimedia and Expo, pp. 977-980 (2008)
9. J. Ravikiran, Mahesh Kavi, Mahishi Suhas, R. Dheeraj, S. Sudheender, Pujari Nitin V.: Finger Detection for Sign Language Recognition In: International MultiConference of Engineers & Computer Scientists, pp. 489 (2009)
10. Lee, J., Kunii, T.L.: Model-based analysis of hand posture. In: IEEE Computer Graphics and Applications, vol.15, no.5, pp.77-86 (1995)
11. Cheng-Chang Lien, Chung-Lin Huang: Model-based articulated hand motion tracking for gesture recognition. In: Image and Vision Computing, vol. 16, Issue 2, pp. 121-134, (February 1998)
12. Lathuiliere, F., Herve, J. Y.: Visual tracking of hand posture with occlusion handling. In: 15th International Conference on Pattern Recognition Proceedings, vol.3, pp.1129-1133 (2000)
13. Dung Duc Nguyen, Thien Cong Pham, Jae Wook Jeon: Fingertip detection with morphology and geometric calculation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1460-1465 (2009)
14. M. Do, T. Asfour, R. Dillman: Partical filter-based fingertips tracking with circular hough transform feature. In: Proceedings of the 12th IAPR Conference on Machine Vision Application (2011)
15. Raheja, J.L., Chaudhary, A., Singal, K.: Tracking of Fingertips and Centers of Palm Using KINECT Computational Intelligence. Third International Conference on Modeling and Simulation (CIMSiM), pp. 248-252 (2011)
16. Hui Liang, Junsong Yuan, Daniel Thalmann: 3D fingertip and palm tracking in depth image sequences. In: Proceedings of the 20th ACM international conference on Multimedia (MM '12). ACM, New York, NY, USA, pp.785-788 (2012)
17. Ramer-Douglas-Peucker algorithm, http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm
18. Michael Kass, Andrew Witkin, Demetri Terzopoulos: Snakes: Active contour models. In: International journal of computer vision, vol. 1, no. 4, pp. 321-331 (1988)
19. Donna J. Williams, Mubarak Shah: A Fast algorithm for active contours and curvature estimation. In: CVGIP: Image Understanding, vol. 55, no. 1, pp. 14-26 (January 1992)