Real-Time 3D Hand Tracking from Depth Images

Lin Song[1,a], Ruimin Hu[1,b], Yulian Xiao[2,c] and Liyu Gong[2,d]

Correspondence author: Ruimin Hu

[1]National Engineering Research Center for Multimedia Software,
Computer School, Wuhan University, China

[2]Eedoo Inc., China

[a]lin_song511@163.com, [b]hrm1964@163.com, [c]xiaoyl@eedoo.cn, [d]gongliyu@gmail.com

Keywords: Hand Tracking; 3D Image Processing; Depth Images; Kinect; Depth Image Segmentation

Abstract. In this paper, we propose a depth image based real-time 3D hand tracking method. Our method is based on the fact that human hand is an end point of human body. Therefore, we locate human hand by finding the end point from a predicted position of hand based on the hand position of the previous frame. We iteratively grow a region around the predicted position. The end point on the major axis of the region which stops moving with region growing is selected as the final position of human hand. Experiments on Microsoft Kinect for Xbox captured sequences show the effectiveness and efficiency of our proposed method.

Introduction

Human-computer interaction (HCI) technology is an active and important research topic which attracts interest from various research areas, including computing power, various sensors, and display techniques [1]. The movements of human hands are useful information for HCI. Compared with traditional point input device such as mouse, controlling with human hand is much more naturally, especially for gaming. As a non-verbal communication modality, human hand motion is a promising HCI approach because of its ability to interaction at a distance [2]. To capture the movements of human hand, tracking algorithms plays a very important role in hand motion based HCI systems. Currently, various kinds of sensors are used in HCI systems, e.g., special gloves which are directly attached to hand, color cameras etc. [3-4]. Since special gloves are intrusive, their application scenarios are limited. On the other hand, hand tracking from a non-intrusive camera is more natural for users. Various color histogram based [5-6] and template based [7] hand tracking algorithm have been proposed. However, all these methods are working in 2D space. Since hand movements generally occur in 3D space, 2D based methods are not robust enough for real applications. However, obtaining 3D information using 2D cameras are not robust and computing intensive. Traditional 3D scanner are slow and expensive thus not suitable for ordinary applications. Recently, 3D cameras are becoming cheaper and faster. Many companies (e.g. Microsoft, PrimeSense [8] etc) have launched their 3D cameras which can capture depth information as well as RGB color information. These affordable and fast cameras open a door for researchers to design new algorithms for hand motion based HCI. For example, Breuer et al. [9] used an infra-red ToF camera to create a near real-time gesture recognition system. Grest et al. [10] proposed a human motion tracking method using a combination of depth and silhouette information.

In this paper, we propose a novel real-time 3D hand tracking method in depth space using Microsoft Xbox Kinect camera. Our method is based on the fact that human hand is an end point of the full hand body. Therefore, we design a segmentation based algorithm to locate a hand from a predicted position of hand. By iteratively growing a region around the predicted position, we observe two end points along the major axis of the region. The one which stops moving with region growing is selected as the final position of human hand. We implemented the algorithm on a Microsoft Kinect for Xbox based HCI system, our algorithms are robust (ready for real-world applications) and efficient (fast enough for make real-time systems).

Published by Atlantis Press, Paris, France.

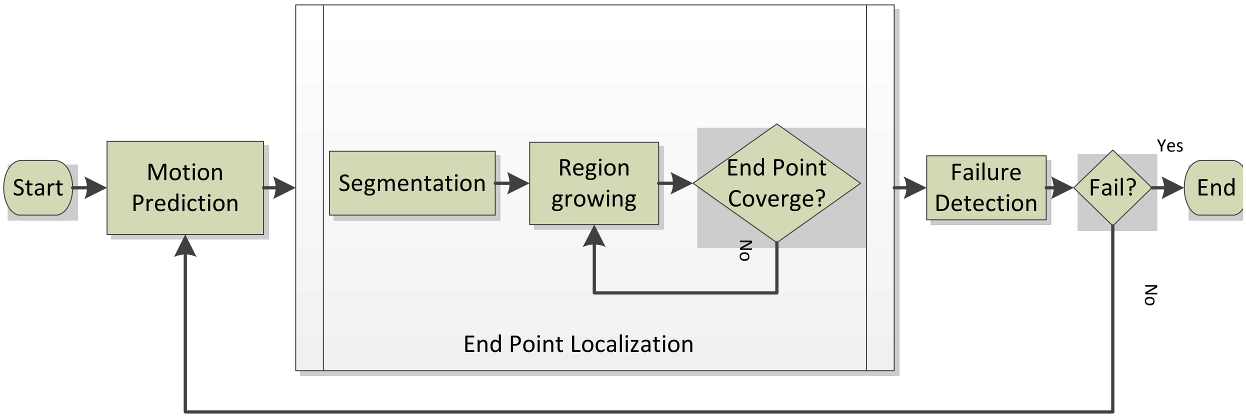


Figure 1. Flow chart of the proposed hand tracking system

Proposed method

The whole tracking system is composed by three major components: a motion prediction module, an end point localization module and a failure detection module. Figure 1 illustrates the flow chart of our proposed system. A typical use case of the system can be described as follow: An external hand detection system detect a focus hand gesture and get the initial position of the hand. Then, the tracking system is initialized using the initial hand position. Then, for each new frame, the motion prediction module predicts a hand position. The predicted position is just a rough guess of the actual hand position. It may be the exact correct position of the end point of hand (e.g. palm), but it will located at a point near the correct point (e.g. arm) in 3D space. Then, the end point localization module locates the end point of hand using the predicted hand position as input. Finally, the failure detection module checks the hand position obtained to see whether it makes any sense. If the obtained position doesn't make any sense at all (e.g. too far from the hand position at the previous frame), the tracking system is restarted and ready for initialization. Otherwise, the tracking system updated the current and previous hand position and step to the next frame.

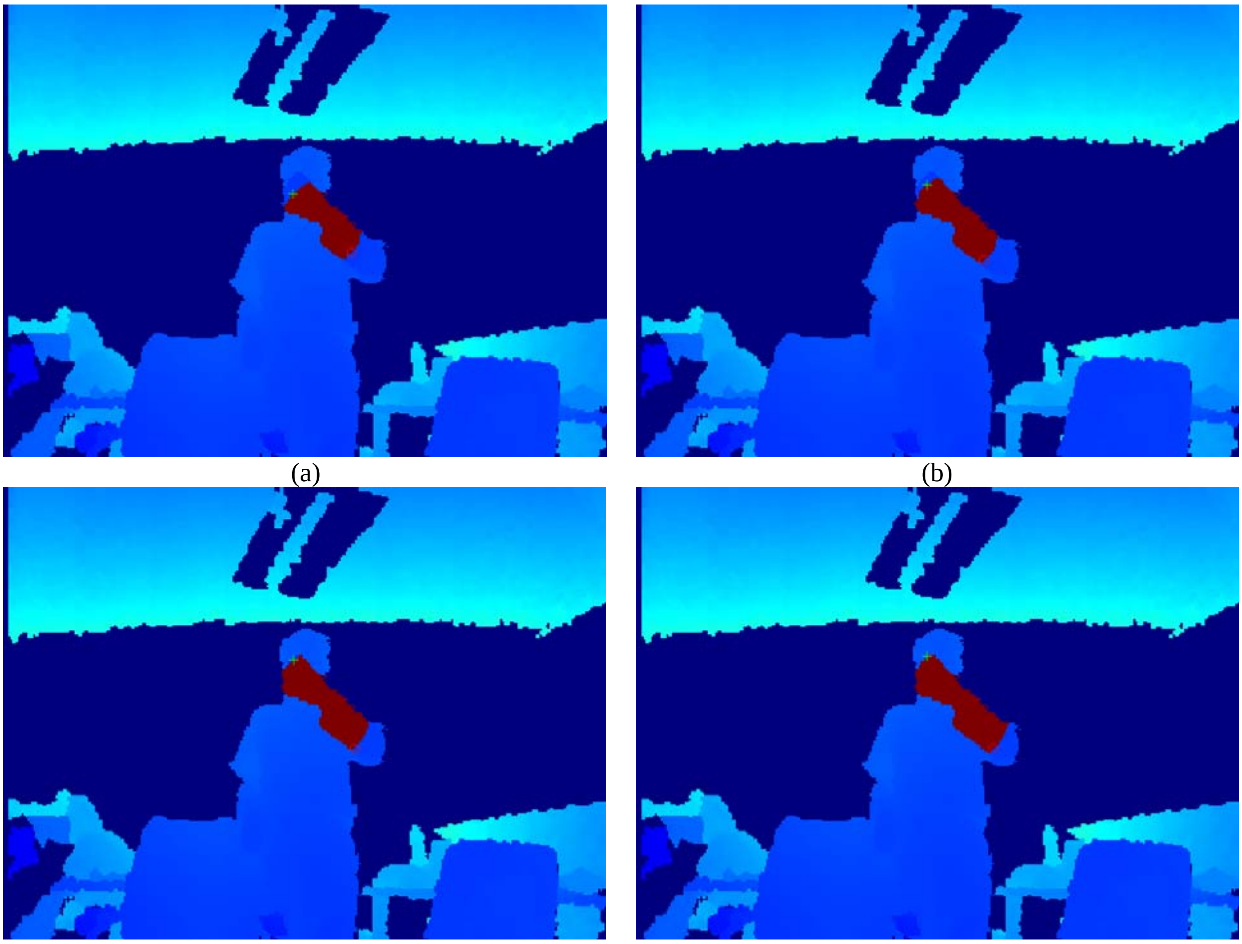
Hand position prediction. The motion prediction module is a top-down process dealing with the dynamics of the tracked hand point. It models the prior of hand motion. To be precise, the motion prediction module predicts the hand position of the current frame using the hand position of the previous frame. It can be formulated as

$$x_i = f(x_{i-1}) \tag{1}$$

Where x_i is the hand position of the i_{th} frame. The most abstract formulation of the motion prediction module is through the state space approach for modeling discrete-time dynamic systems, such as Kalman Filter. Because 3D depth information is much more discriminative than traditional RGB color information, simple motion model can be used instead of computing intensive sophisticated model. In our implementation, we employ a much simpler model: we convert the depth image into a 3D point cloud and find a point with smallest distance to the hand position of the previous frame in the 3D space. In our experiments, such a simple model shows results which are robust enough for real-world applications.

End point localization. The end point localization module is a bottom-up process which copes with the changes in the appearance of the tracked hand point. It models the prior of hand shape. Since the shape of human hand is a rigid stick with one end point (another end is connected with human body), we localize human hand by finding the end point from a rough guess of the hand position. We propose an iterative point cloud segmentation based end point localization algorithm:

1) Firstly, we segment out a region around the predicted point in 3D space with radius R.



(a) (b)

(c) (d)

Figure 2. Illustration of the end point localization module. Green cross and red circle indicate two end points along the major axes. From (a) to (c), the two end points moves with the region growing. From (c) to (d), the green cross stay unchanged because the region reaches the end of hand. But the red circle keeps moving because another end of the region still grows to the human body. Therefore, the point represented by the green cross is returned at (d).

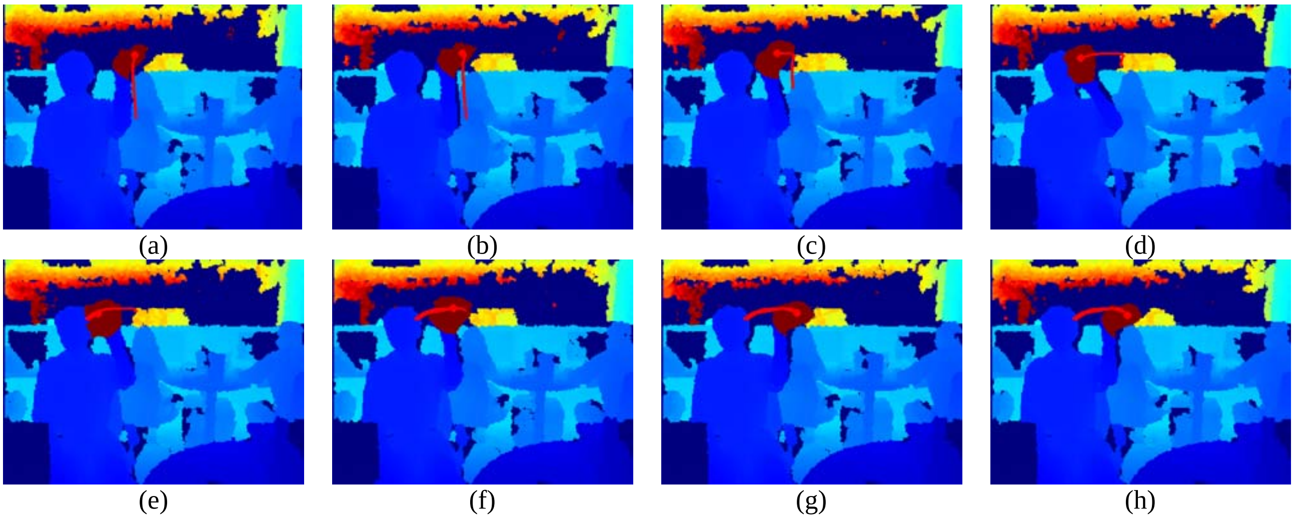
2) Then, we calculate the major axes using principal component analysis on the points which are segmented out and store the two end point x_1 and x_2 of the region segmented out along the major axes.
3) Next, we grow the region segmented by re-segment with a larger radius $R^{'}=R+r$;
4) Next, we compare the new end points of the grown region along the same major axes obtained by 2) with the old end points obtained by 2). If one of the end point is the same as its old counterpart, then the end point is returned as the hand position; else if $R^{'} > R_{max}$, return the end point which is closer to the predicted point; else goto 3).

Figure 2 gives an illustration of the end point localization procedure.

Failure detection. Failure detection module imposes constraints on the motion and shape of human hand. We decide a failure is occurred (i.e. the hand is lost by the tracker) while three constraints are met: 1) the velocity of the hand moving are too high (e.g. 50cm/second which makes no sense). 2) the ratio between largest and second largest eigen value of the segmented points is too small, which means the segmented region does not look like a stick.

Experimental results

We run our system on a personal computer with Microsoft Kinect for Xbox. Figure 3 show some results of the system, which is promising for practical applications. Moreover, our unoptimized Matlab code process 20 frames per second on a notebook with Core 2-Duo 2.4G Hz CPU and 2GB



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3. Experimental results of our proposed system.

memory. Note that the Kinect sensor captures 30 frames per second. We believe that a C++ version of the algorithm will achieve 30 fps.

References

[1] A. Just and S. Marcel, A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition. Computer Vision and Image Understanding, 113(4):532-543, 2009/

[2] B. Ionescu, D. Coquin, P. Lambert and V. Buzuloiu, Dynamic hand gesture recognition using the skeleton of the hand. EURASIP Journal of Applied Signal Processing, 2005:2101-2109, 2005.

[3] D. J. Sturman and D. Zelter, A survey of glove-based Input. IEEE Computer Graphics and Applications. 14(1):30-39, 1994.

[4] R.Y. Wang and J. Popovic, Real-time hand-tracking with a color glove. ACM Transaction on Graphics. 28(3):1-8, 2009.

[5] R. Kjeldsen and J. Kender, Toward the use of gesture in traditional user interfaces, in Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA, 1996.

[6] K. Imagawa, S. Lu and S. Igi, Color-based hands tracking system for sign language recognition, in IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998.

[7] B. Stenger, A. Thayananthan, P.H.S. Torr and R. Cipolla, Model-based hand tracking using a hierarchical bayesian filter. IEEE Transaction on Pattern Analysis and Machine Intelligence. 28(9):1372-1384, 2006.

[8] PrimeSensor http://www.primesense.com

[9] P. Breuer, C. Eckes and S. Muller, Hand gesture recognition with a novel IR time-of-flight range camera-a pilot study. Lecture Note on Computer Science, 4418,247, 2007.

[10] D. Grest, V. Kruger and R. Loch, Single view motion tracking by depth and silhouette information, in Proceedings of the Scandinavian Conference on Image Analysis, Aalborg, Denmark, pp. 719-729, June 2007.