Real-Time 3D Hand Gesture Detection from Depth Images

Lin Song, Ruimin Hu, Hua Zhang
National Engineering Research Center for Multimedia Software
Computer School, Wuhan University, Wuhan, Hubei, China
{lin_song511,hrm1964}@163.com
zhanghua@whu.edu.cn
correspondence author: Ruimin Hu

Yulian Xiao
Eedoo Inc.
Beijing, China
xiaoyl@eedoo.cn

Liyu Gong
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, Hubei, China
gongliyu@gmail.com

***Abstract*—In this paper, we describe an real-time algorithm to detect 3D hand gestures from depth images. Firstly, we detect moving regions by frame difference; then, regions are refined by removing small regions and boundary regions; finally, foremost region is selected and its trajectories are classified using an automatic state machine. Experiments on Microsoft Kinect for Xbox captured sequences show the effectiveness and efficiency of our system.**

Keywords-3D Hand Gesture Detection; Depth Images; Kinect; Image Recognition; Human Computer Interface

I. INTRODUCTION

Human hand motion is an effective and natural way for human computer interaction (HCI). However, hand detection in images or videos is a challenging problem due to variations in pose, lighting conditions and complexity of the backgrounds. Although the hand detection problem has been investigated intensively, most of the work is based on images taken by visible-light cameras. Some of them detect hands using skin color features [1,2,3]. Another kind of methods detect hands using Viola & Jones like cascade detectors built from Harr features [4,5,6]. Some methods detect hands as part of a human pictorial structure [7,8,9] which provides spatial context for the hand position. All these methods are not robust enough and their computation is pretty time-consuming. Depth information is an important cue when human recognizes objects because the objects may not have consistent color and texture but most occupy an integrated region in space. Although object recognition from depth images captured by 3D sensors has been investigated in the past decades, 3D sensors in the early age are very expensive and slow thus not practical in really time HCI applications. Now, Microsoft has launched the Kinect, which is cheap and very easy to use. Also, it does not have the disadvantages of laser so it can be used in human environment and facilitate the research in human computer interaction.

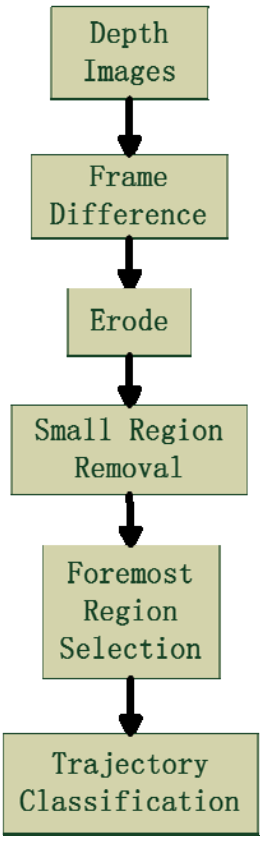

Figure 1. overview of our proposed system. Firstly, frame difference is employed to detect moving regions from two consecutive depth images. Then, the rough regions are refined by erode and small region removal. Next, the foremost region is selected as candidate moving hand. Finally, the trajectory of the foremost moving region is classified to verify the focus gesture.

Published by Atlantis Press, Paris, France.

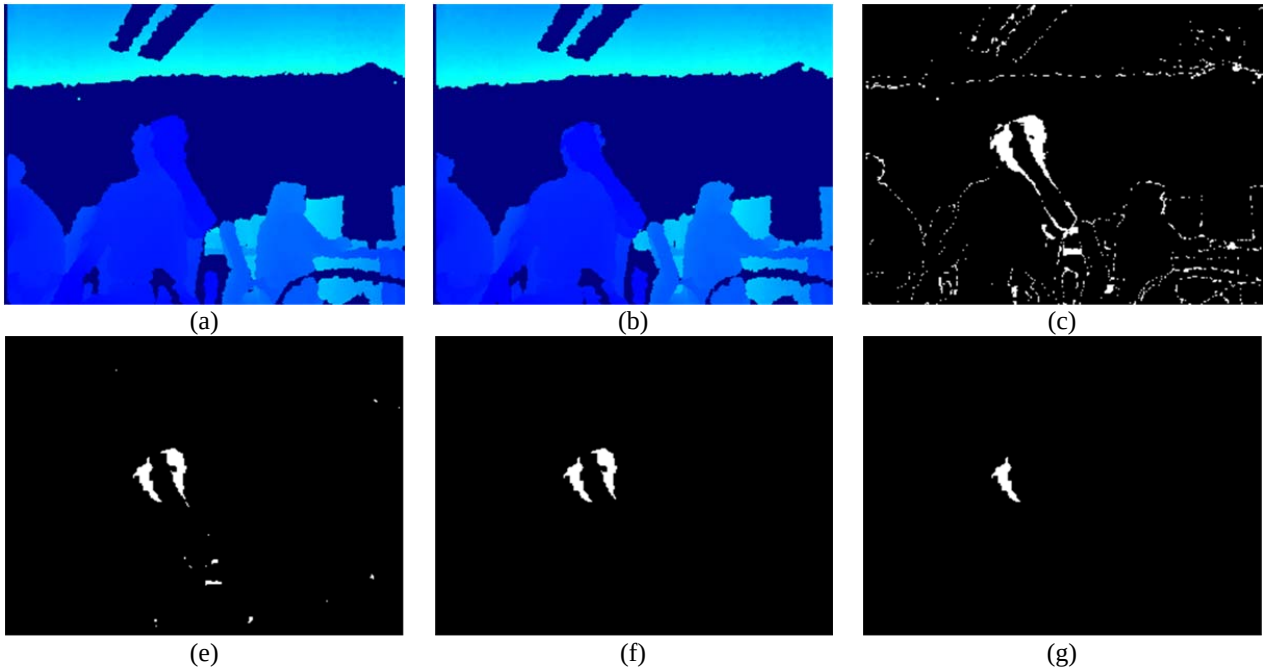

Figure 2. Illustration of moving hand detection procedure. (a) and (b) are two consecutive depth frames. (c) is the resultant mask map of frame difference operation. (e) is the result of erode operation. (f) is the result of small region removal. (g) is the mask map of the foremost region.

In this paper, we propose a real-time hand detection approach for human computer interaction. Our system is different from previous systems in two aspects: First, we use a depth camera instead of traditional visible light cameras. Second, we detect hand motion gestures instead of appearance features. Because there is no need to compute complex visual features (e.g. HOG and SIFT) or decision function (e.g. cascade classifier), our method can run at real-time. An overview of our proposed system is present in Fig 1. Firstly, frame difference is employed to detect moving regions from two consecutive depth images. Then, the rough regions are refined by erode and small region removal. Next, the foremost region is selected as candidate moving hand. Finally, the trajectory of the foremost moving region is classified to verify the focus gesture.

II. MOVING HAND DETECTION

We employ a background subtraction framework to detect moving hands. For efficiency reasons, a simple frame difference algorithm is used to detect rough moving regions.

$$\mathrm{M} = \mathrm{abs}(D_1 - D_2) < \Delta d \qquad (1)$$

Where D_1 and D_2 are two consecutive depth frames. Δd is a threshold for frame difference mask computation.

Since depth images captured by most real-time 3D sensors (e.g. Microsoft Kinect) are noisy, the moving regions detected by frame difference also contain lots of small regions caused by noise. Thus we remove small region from the resultant mask map by a erode operation and an area open operation. After that, the foremost region is selected as the moving hand. Figure 2 gives an illustration of moving hand detection procedure. In Figure 2, (a) and (b) are two consecutive depth frames. (c) is the resultant mask map of frame difference operation. (e) is the result of erode operation. (f) is the result of small region removal. (g) is the mask map of the foremost region.

Note that the final foremost moving regions detected by the approach described in this section are just candidate positions of human hand. Since false alarms are also contained in the candidate positions, we further verify hand positions by detecting focus gestures (e.g. wave) using temporal features of the trajectory.

III. TRAJECTORY CLASSIFICATION

To detect focus gestures, we propose an automatic state machine (ASM) based approach to classify the trajectory of moving hand (i.e. foremost moving region). Figure 3 gives an illustration of the automatic state machine for specified focus gesture “wave” detection. To detect “wave”, four states of the 3D trajectory are used: “Not Ready”, “Horizontal Moving”, “Steady” and “Wave Detected”. “Not Ready” indicates that the waving gesture of the point (i.e. centroid of the foremost region) does not begin at all. “Horizontal Moving” indicates that the point is in a path of moving left or right. During this state, if the direction of moving is changed, a flip is counted. The number of flip is checked and the state will transit to “Wave Detected” if the number of flips satisfies a threshold. To make the classifier more robust, we employ another state called “Steady”, which means the point is not moving. To be precise, we list the details of the state transition conditions as follow:

Published by Atlantis Press, Paris, France.

(a). Current displacement of the point is not horizontal (i.e. the angle between the displacement vector and x-axis is above a given threshold) or it is the first point of the trajectory. Stay at "Not Ready" state.

(b). Current displacement of the point is horizontal (i.e. the angle between the displacement and x-axis is below a given threshold). Transit from "Not Ready" to "Horizontal Moving".

(c). C

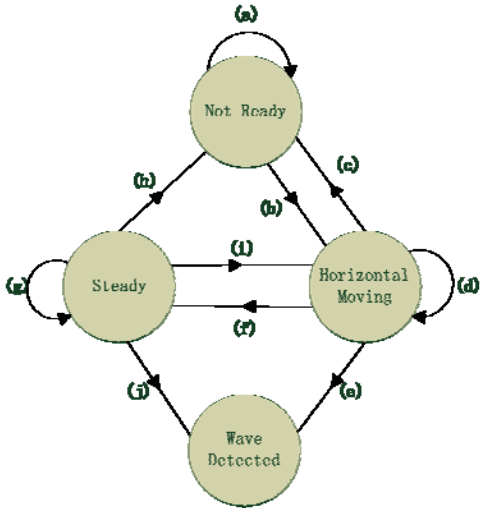

Figure 3: Automatic state machine for "Wave" detection. See the main text for more detail explanation about the states and transition conditions.

nd previous displacement is not in the same direction (i.e. the angle between current displacement and previous displacement is below a given threshold) or a flip occurred (i.e. the angle between current displacement and previous displacement is larger than a threshold) but the accumulated displacement of a continuous horizontal moving is too short. Transit from "Horizontal Moving" to "Not Ready".

(d). Current displacement is in the same direction of the previous accumulated horizontal displacement or a flip occurred and the previous accumulated horizontal moving is long enough. Stay at "Horizontal Moving" state.

(e). A flip occurred and the previous accumulated horizontal moving is long enough and the number of flips is above a given threshold. Transit form "Horizontal Moving" to "Wave Detected".

(f). Length of current displacement is below a given threshold. Transit from "Horizontal Moving" to steady and mark the current point as the center point of "Steady".

(g). Length of the displacement from center point of "Steady" to current point is smaller than a given threshold. Stay at "Steady" state.

(h). Length of the displacement from center point of "Steady" to current point is larger than a given threshold and condition (d) is not satisfied if we replace all the points in the "steady" state with a single point. Transit from "Steady" to "Not Ready".

(i). Length of the displacement from center point of "Steady" to current point is above a given threshold and condition (d) is satisfied if we replace all the points in the "steady" state with a single point. Transit from "Steady" to "Horizontal Moving".

(j). Length of the displacement from center point of "Steady" to current point is above a given threshold and condition (e) is satisfied if we replace all the points in the "steady" state with a single point. Transit from "Steady" to "Horizontal Moving".

Note that the "Wave Detected" state is just a temporary state. Every time a "Wave Detected" is trigged, the state will transit to "Not Ready" automatically in order to detect the next waving gesture.

In some frames, there is no moving region detected. We do not feed any 3D trajectory point into the ASM in such cases. That's to say, the ASM is paused for frames without any moving region. To further improve the robustness of the system, we reinitialize the ASM (i.e. set the state to "Not Ready" and number of flips to 0) if no moving regions are detected for a long time (e.g. number of consecutive frames without moving region is larger than a threshold).

Our proposed ASM framework is suitable for other kind of gestures (e.g. raise hand, push) also. Therefore, we can define a couple of focus gestures and detect these gestures instead of detecting human hands by computing complex appearance features. In a real world application, asking the user to perform a simple focus gesture is not difficult and acceptable. With such a simple assumption we can avoid complex feature extraction and save valuable computing resources. Therefore, our method could be implemented in hardware platforms with limited computing resources.

IV. Experimental Results

To test our algorithm, we collect a benchmark dataset using a Microsoft Kinect sensor for Xbox. Specifically, 25 sequences which contain wave gestures are recorded as positive samples. Another 25 sequences which contain wave-like but not wave (e.g. moving up and down) are recorded as negative samples. For positive samples, each sequence contains one wave gesture only.

We run our algorithm on the collected dataset. The average recognition rate of our algorithm is 94% (i.e. only three samples are misclassified), 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 memory. Note that the Kinect sensor captures 30 frames per second. We believe that a C++ version of the algorithm will achieve 30 fps. Figure 4 gives some example results of the experiments.

V. Conclusion

In this paper, we propose a novel real-time depth image based 3D hand gesture detection algorithm. We detect the moving region using frame difference first and classify the trajectory of moving regions using an automatic state

machine. Experimental results justify the effectiveness and efficiency of our algorithm.

There are several interesting directions which worth investigating in the future. Firstly, we will add more focus gesture ASM classifier into our system. Secondly, further refine the candidate moving region position using an end point locating algorithm will improve the robustness of our system much. Thirdly, we will investigating depth image based hand tracking problem. Finally, the ASM based trajectory classification method can be applied to trajectories generated by tracking also.

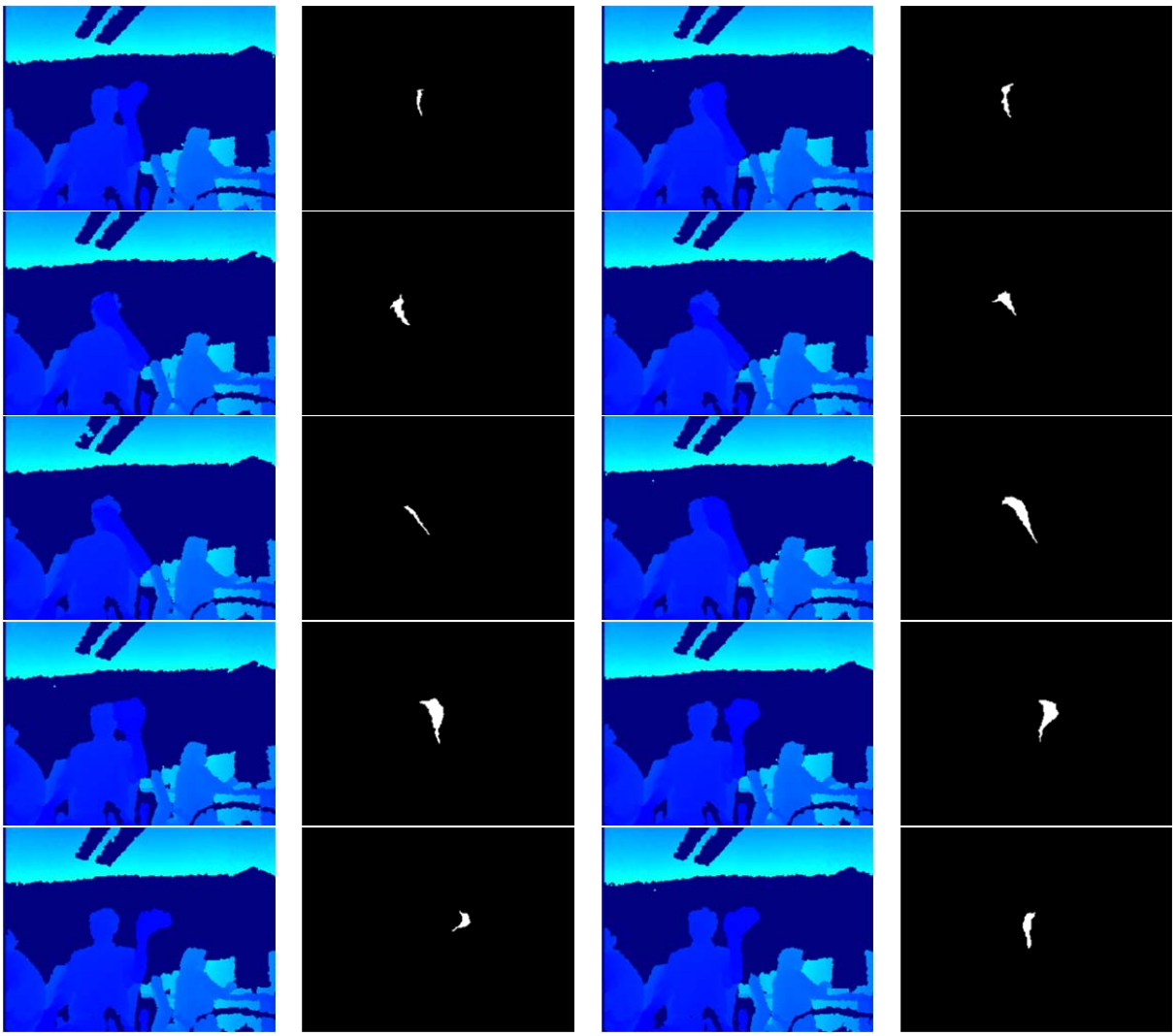

Figure 4. An example of detected wave gesture. From left to right, top to bottom are depth images and corresponding moving region masks of a sequence. The states of the automatic state machine are transit and finally a wave gesture is detected.

REFERENCES

[1] Y. Wu, Q. Liu, and T. S. Huang. An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization. In Proceedings of Asian Conference on Computer Vision, 2000.

[2] Y. Wu and T. S. Huang. View-independent recognition of hand postures. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[3] X. Zhu, J. Yang and A. Waibel. Segmentation hands of arbitrary color. In Proceedings of International Conference of Automatic Face and Gesture Recognition, 2000.

[4] M. Kolsch and M. Turk. Robust hand detection. In Proceedings of International Conference on Automatic Face and Gesture Recognition, 2004.

[5] E.J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In Proceedings of International Conference on Automatical Face and Gesture Recognition, 2004.

[6] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001.

[7] P. Buehler, M. Everingham, D. P. Huttenlocher and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In Proceedings of British Machine Vision Conference, 2008.