Poster: Real-time Markerless Kinect based Finger Tracking and Hand Gesture Recognition for HCI

Arun Kulshreshth* Chris Zorn† Joseph J. LaViola Jr.‡

Department of EECS
University of Central Florida
4000 Central Florida Blvd
Orlando, FL 32816

ABSTRACT

Hand gestures are intuitive ways to interact with a variety of user interfaces. We developed a real-time finger tracking technique using the Microsoft Kinect as an input device and compared its results with an existing technique that uses the K-curvature algorithm. Our technique calculates feature vectors based on Fourier descriptors of equidistant points chosen on the silhouette of the detected hand and uses template matching to find the best match. Our preliminary results show that our technique performed as well as an existing k-curvature algorithm based finger detection technique.

Keywords: Finger tracking, hand gestures, human computer interaction, Microsoft Kinect

Index Terms: H.5.2 [Information Interfaces and Presentation]: User interface—Graphical user interfaces; input devices; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking;

1 INTRODUCTION

Developing natural and intuitive interaction techniques is an important goal in HCI. Typically, humans interact with computers using mouse and keyboard. Outside of computer interaction, we are used to interacting with the world using our hands, body, and voice. Interfaces based on interaction with hands are a natural and intuitive way to interact with computers. Such an interface could be used for robot and human collaboration, virtual reality, scientific visualization, geographic information systems (GIS), or games. With the widespread use of the Microsoft Kinect, increasingly more people are creating interfaces based on full body and voice recognition.

Finger and hand gesture recognition with the Kinect is still an open problem due to its low resolution (640×480), especially considering how hands occupy a much smaller portion of the full body image. Traditional vision-based hand gesture recognition methods [1, 2, 3] are far from satisfactory due to the limitations of the optical sensors used and dependency on lighting conditions and backgrounds. Data gloves [4] can be used for precise accuracy, but require the user to wear a special glove; this may hinder the naturalness of the hand gesture. The Microsoft Kinect is a commodity hardware device that can be used for designing natural gesture based interfaces.

In this paper, we describe a framework for a real-time markerless finger tracking technique using Microsoft Kinect as an input device and apply the technique for hand gesture recognition. We compare the results of our technique with another technique based on work of Trigo et al. [5].

2 RELATED WORK

Gesture recognition has been an ongoing research problem for many years. One of the first tracking systems to analyze articulated hand motion was presented in [6]. In their system, a 27 degree-of-freedom hand could be tracked at 10Hz by extracting point and line features from grayscale images. However, their system could not track in the presence of occlusion and complicated backgrounds; it also required a manual initialization step before tracking could begin. Jennings [7] used multiple cameras for finger tracking, but their system was too slow to be used in real time. Oikonomidis et al. [8] proposed a markerless model based algorithm for hand gesture recognition and achieved accurate and robust 3D hand tracking at 15Hz. Frati et al. [9] developed a hand tracker based on the Kinect sensor for wearable haptics. Finger tracking is also explored in the area of augmented reality [10, 11]. Ren et al. [12] proposed a robust hand gesture recognition framework based on finger-earth mover's distance. Our proposed technique is real-time and does not require a simple background.

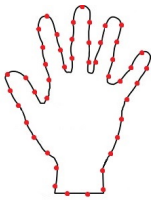

Figure 1: Hand contour with equidistant boundary points. Our feature vector is based on the discrete Fourier transform of the centroid distance of these points.

3 FINGER TRACKING AND HAND GESTURE RECOGNITION

We used the Microsoft Kinect, which captures both RGB and depth images at 640×480 resolution.

3.1 Hand Segmentation

We used Kinect skeleton tracking to detect the approximate $2D$ position (P_x, P_y) of the palm of a hand in the depth image. We require the user to make sure that the hand is the front-most object facing the Kinect sensor. The Kinect sensor is adjusted by rotating up or down so that hands are approximately in the middle of the detected depth image. Median filtering [13] is applied to remove noise from the detected hand region. A bounding box, with parameters top, left, width, and height, of the hand are found heuristically by scaling an initial box (of size 10×10) based on the distance of the hand from the sensor. These equations take into account how the hand has more area above the center of the palm (fingers) then below it (wrist) and are described below.

$$\text{Scale} = \frac{7000}{\text{Depth}(P_x, P_y)}$$

$$(\text{top,left}) = \big(P_x - 10 * \text{Scale} * 0.5, P_y - 10 * \text{Scale} * 0.6\big)$$

$$(\text{width,height}) = (10 * \text{Scale}, 10 * \text{Scale})$$

3.2 Finger Tracking and Recognition

The segmented hand image obtained from the previous step is a grayscale image and may have some missing pixels due to noise from the Kinect. We fix those missing pixels by finding the contours in the segmented image and filling areas smaller than a threshold value. The contour of the hand was found by converting the image to binary. Our classification technique is based on work by Zang et al. [14]. A predefined number, N, of equidistant points (see Figure 1) were selected on the contour, and a centroid distance function was calculated based on these selected points as:

*e-mail: arunkul@knights.ucf.edu
†e-mail: czorn@knights.ucf.edu
‡e-mail: jjl@eecs.ucf.edu

IEEE Symposium on 3D User Interfaces 2013
16-17 March, Orlando, FL, USA
978-1-4673-6098-2/13/$31.00 ©2013 IEEE

$$r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{1/2}$$

where (x_c, y_c) is the centroid of the selected boundary points and $t = 0, 1, 2, \ldots, N$. We then calculate the Discrete Fourier transform of $r(t)$ as

$$u_n = \frac{1}{N} \sum_{t=0}^{N-1} r(t) exp\left(\frac{-j2\pi nt}{N}\right), n = 0, 1, 2, \ldots, N-1.$$

The coefficients of $u_n, n = 0, 1, 2, \ldots, N-1$ are called Fourier descriptors (FD) of the shape denoted as $FD_n, n = 0, 1, 2, \ldots, N-1$. Once we calculate Fourier Descriptors (FD) of the centroid distance function, we can get the feature vector, f as

$$f = \frac{|FD|_1}{|FD|_0}, \frac{|FD|_2}{|FD|_0}, \ldots, \frac{|FD|_{N/2}}{|FD|_0}.$$

This feature vector is invariant to translation, rotation and scale. During the training phase, a model feature vector, $f_m = \left[f_m^1, f_m^2, \ldots, f_m^{N/2}\right]$, is calculated for each gesture in the gesture set by taking the average of feature vectors for that same gesture. For classifying a given vector, $f_d = \left[f_d^1, f_d^2, \ldots, f_d^{N/2}\right]$, Euclidean distance, D, between the f_d vector and model vectors is used as a similarity measurement and is calculated as

$$D = \left(\sum_{i=0}^{N/2} \left|f_m^i - f_d^i\right|^2\right)^{\frac{1}{2}}.$$

The nearest match with a given rejection threshold distance is used to find if a match exists in the training database.

3.3 Implementation details

For our implementation we used Microsoft's WPF library, the Kinect SDK v1.5, and the emgu [15] library with Nvidia CUDA support.

3.4 Existing Technique

The second method for identifying fingers was an implementation that analyzed the outline of the depth image of the hands [5]. First, we used the Kinect SDK to determine the location of the user's hands. Next, we created a bounding box around the hands. Within the box, we identify the depth pixels that lie on the contour of the hand. Finally, we apply the K-curvature algorithm as described, to locate the points of inflection on the contour. These points are the tips of the extended fingers.

4 User Study

To determine the accuracy of our finger tracking system, we compared it to an implementation of the existing technique by Trigo et al. in a user study. We recruited 10 participants (4 female, 6 male, ages 18-30) from the University of Central Florida and asked them to put their hands into several poses by raising different numbers of fingers (see Figure 2) in front of the Kinect. Participants made 6 poses, expressing the numbers 0 - 5 with their fingers, with each hand for each technique, for a total of 24 poses. They were not told which technique was analyzing their hands; the order of the techniques was randomized between subjects. We recorded the recognition output of both the techniques. The mean accuracy of our technique, 90.9%, was statistically similar to that of the existing technique, 88.3%.

5 Discussion and Conclusion

Our technique performed similarly to the existing technique for counting the number of fingers held up by a user. Under both techniques, the accuracy was most affected by the limited resolution of the Kinect. Although the Kinect provides images with a resolution of 640×480, the hands are typically a small fraction of the image. The accuracy of the counting improved as the user's hands moved closer to the Kinect, which provided a crisper image of the hands; beyond a distance of about 4-5 feet, neither technique could accurately determine the number of fingers held up.

The existing technique worked best when the user's hands were parallel to the camera plane and when the fingers were sufficiently spread apart, providing a more optimal contour. Because our technique uses training data, it was less affected by sub-optimal contours. However. it was less able to recognize poses that didn't match the training data, such as poses where the hand was rotated by 90°. This could be improved by increasing the training data to include less common poses.

Figure 2: Gesture set used in user study. Participants created these poses with both hands for each technique.

6 Acknowledgments

This work is supported in part by NSF CAREER award IIS-0845921 and NSF awards IIS-0856045 and CCF-1012056.

References

[1] C.S. Chua, H. Guan, and Y.K. Ho. Model-based 3d hand posture estimation from a single 2d image. *Image and Vision computing*, 20(3):191–202, 2002.

[2] Ho-Sub Yoon, Jung Soh, Younglae J. Bae, and Hyun Seung Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 34(7):1491 – 1501, 2001.

[3] Feng-Sheng Chen, Chih-Ming Fu, and Chung-Lin Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745 – 758, 2003.

[4] Z. Ren, J. Yuan, C. Li, and W. Liu. Minimum near-convex decomposition for robust shape representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 303–310. IEEE, 2011.

[5] T.R. Trigo and S.R.M. Pellegrino. An analysis of features for hand-gesture classification. In *17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010)*, pages 412–415, 2010.

[6] J.M. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 16–22. IEEE, 1994.

[7] C. Jennings. Robust finger tracking with multiple cameras. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*, pages 152 –160, 1999.

[8] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. *BMVC, Aug*, 2, 2011.

[9] V. Frati and D. Prattichizzo. Using kinect for hand tracking and rendering in wearable haptics. In *World Haptics Conference (WHC), 2011 IEEE*, pages 317 –321, june 2011.

[10] K. Dorfmuller-Ulhaas and D. Schmalstieg. Finger tracking for interaction in augmented environments. In *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pages 55 –64, 2001.

[11] J. Crowley, F. Berard, J. Coutaz, et al. Finger tracking as an input device for augmented reality. In *International Workshop on Gesture and Face Recognition, Zurich*, pages 195–200. Citeseer, 1995.

[12] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 1093–1096, New York, NY, USA, 2011. ACM.

[13] A. Marion. *An Introduction to Image Processing*. Chapman and Hall, 1991.

[14] D. Zhang and G. Lu. A comparative study on shape retrieval using fourier descriptors with different shape signatures. In *Proc. of international conference on intelligent multimedia and distance education (ICIMADE01)*, pages 1–9, Fargo, ND, USA, 2001.

[15] Emgu library. `http://www.emgu.com/wiki/index.php/Main_Page`.