Finger Application Using K-Curvature Method and Kinect Sensor in Real-Time

M. Zabri Abu Bakar, Rosdiyana Samad, Dwi Pebrianti, Mahfuzah Mustafa, Nor Rul Hasma Abdullah
Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang (UMP)
26600 Pekan, Pahang, Malaysia
rosdiyana@ump.edu.my

Abstract—**Gesture is one of the important aspects of human interaction and also in the context of human computer interaction. Gesture recognition is the mathematical interpretation of a human motion by a computing device. It is often used hand gestures for input commands in personal computers. By recognizing the hand gesture as input, it allows the user to access the computer interactively and makes interaction more natural. This paper presents a finger detection application by using Kinect. Kinect is a depth sensor that is an effective device to capture the gesture in real-time. To detect and recognize the fingertips, it needs to extract the detail of the captured hand image using image processing methods. In this paper, the proposed method is to detect and recognize the fingertips by using the K-Curvature algorithm. Finally, the finger counting application is applied and the proposed method is discussed at the end of this paper. The results obtained from the experiment show that the acceptable average accuracy for the fingertips detection is 73.7% and the average processing time is 15.73 ms. By considering this result, the application of the proposed method can be extended to the hand rehabilitation system.**

***Index Terms*—hand tracking, finger detection, Kinect sensor, gesture recognition.**

I. INTRODUCTION

Real-time finger detection application is one of the natural interaction methods that can perform in human computer interaction (HCI). The conventional interaction to send an instruction is by using the keyboard, mouse, touchpad, joystick, and other mechanical devices. The real-time finger detection can be completed by using a camera or other interactive imaging device such as Kinect sensor, Asus Xtion and thermal camera. Kinect sensor is an imaging device that consists of RGB camera and infrared (IR) sensor, which is producing a depth image [1].

The motivation to develop the finger detection in real-time is to make the interaction between human and computer become more interactive. Normally, to access the computer interactively, the hand detection or recognition system is most favorable to replace mouse pointer. Some applications not only limited to hand recognition but also include finger detection to make the detection or recognition system more robust. In this work, the gesture recognition is started with the hand tracking and followed by the fingertips detection. This paper only focuses on how to detect fingertips using image processing methods. The image of the moving hand is captured by using Kinect sensor. Then, the application of fingertips detection is applied and tested. The main contribution of this paper is the proposed method. The proposed method that is used in this paper is the thresholding in a depth image which is able to manage with the various light condition and a *K* - Curvature algorithm to detect fingertip after hand segmentation with ROI of detection hand.

II. RESEARCH BACKGROUND

Nowadays, many applications in real-time can be created based on finger detection such as in game controller, automatic sign language recognition, public information display panel, smart television, and smart phone. With the advance of Human Computer Interaction (HCI) technology, it has increased the number of researchers in the signal detection and recognition research area to create a user-friendly application.

J. Farooq et al. developed a music player and virtual mouse application in the year 2014 by implementing gesture recognition. They implemented the application after finishing the template matching and finger tips detection of hand gesture recognition [2]. In their work, they used color segmentation in the initial stage to recognize hand and mention that using the color segmentation is a very difficult task. The color segmentation needs to apprehension about the skin color and the background color. Another researcher also used skin color segmentation to tracked hand. However they controlled the background lighting to minimize the variations of the skin color. N. Soontranon et al. is one of the researchers that used skin color segmentation in hand recognition for the sign language application [3].

To develop finger detection application, a hand tracking and detection need to be configured at first before performing a real-time finger detection. Some methods are created and updated for implementation in application time by time. Universal remote control system based on computer vision has been published by D. Lee et al. [4]. The application is a usable finger counting as a command for the information control. The method that they used to detect the hand wave was a *k-cosine* in measuring the curvature angle to detect finger. In their finding, the user couldn't wear long sleeve to prevent capability of detection. This happened because of the problem with color

978-1-4799-1723-5/15/$31.00 ©2015 IEEE

segmentation in the initial process. Another method to find the fingertips is the dominant point method that used by Z. Y. Ming et al. in their publish paper [5], which is almost similar with previous method. The dominant point was obtained by scanning the point on the contour of the finger, then the triangle was constructed and needed to find the threshold angle of the triangle. The author also made a comparison between an object with a different shape to test the dominant point. However, the result was not consistent and not precise and they suggested to add a fuzzy logic to make a classification.

The curvature algorithm is a famous method that is used to detect certain shape of the object and also for fingertips detection. Recently, this method has been expanded with the K-Curvature method. This method tries to find the angle between two points at the finger. If the threshold angle between two points is in range, the middle point will be considered as fingertips [6, 15, 16].

A finger counting application was developed to be used as a remote commander for the robot [7]. The command from the finger gestures instructs the robot to move forward, backward, right and left. The advantage of this application is that it can control the robot movement from the distance. This application was developed by using convex hull and convexity defect. However, this method has some limitations on finger counting, which are the accuracy of the detection and the finger counting only can be used to count not more than five fingers. Another different method to detect the fingertips had proposed in [8]. They created the real-time finger counting application by placing a color marker on the fingertips. Color detection was implemented to detect a color marker and to recognize the shape of the finger for a finger counting application. This method needs to manipulate the color marker to make it different with the background color and need to mark the finger even though it was a low cost method.

Recently, many imaging devices with the compact version and reasonable price are sold such as Kinect sensor. Kinect sensor becomes popular for three dimensions (3D) data tracking. The 3D data tracking is widely used to overcome skin color and cluttered background problem [9]. The 3D data are produced by the infrared (IR) emitter that transmit the IR signal to the IR depth sensor, which this sensor receives a depth distance data. Carolina, Barreto, and Urbano had published a method to calibrate Kinect sensor with high accuracy and fast [10]. It was shown that the performance of Kinect sensor was reliable. Fig. 1 shows the structure of the Kinect sensor from Microsoft XBOX 360.

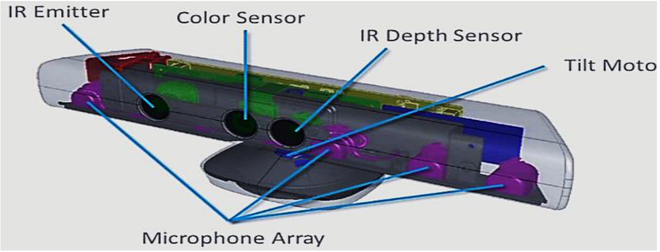


Figure. 1. Kinect sensor structure

This paper also demonstrates the usage of Kinect sensor to track hand and fingertips. An open source interaction which is OpenNI2 is used to communicate between Kinect sensor and computer. The advantage of Kinect sensor is, it can capture video data in 3D under acceptable low ambient light condition. It is also cheap and high resolution in short range 3D device [11].

III. Gesture Recognition

In this manuscript, the hand gesture recognition has been done in real-time. From the gesture recognition, the user can develop a computer vision application which can communicate human and computer. Before recognizing the gesture, the tracking must be conducted first. The hand tracking and gesture recognition method will be described in this section.

A. Hand Tracking and Detection

The process flow for hand tracking had been explained in detail in the earlier publication [13]. In this paper, the hand tracking algorithm will be summarized and the process flow is shown in Fig. 2. At the initial stage, the hand needs to be tracked before recognizing the fingertips. To develop this algorithm, the middleware in OpenNI2 and OpenCV library are used in this stage. The hand tracking algorithm will start with the hand detection by using gesture recognition algorithm from NiTE module. The gesture recognition is initialized by using three types of motion gesture. The motion gestures are *hand waving*, *hand raise* and *hand push*. Once the gesture is recognized, it will obtain the location (point) of the palm.

By using the hand point, a square is drawn around the hand's area and this is the region of interest (ROI) for this image. The ROI of hand is set to follow the hand movement. The ROI is resized automatically, which is the size is increased when the hand moves forward near to the Kinect sensor or decreased when the hands move away from the Kinect as shown in Fig. 3. In the previous publication, it also introduced the segmentation process using the depth threshold [13]. This method calculated a distance from the hand point to the Kinect and set a threshold to get a suitable depth of hand. Fig. 4 shows the illustration of this method. The image segmentation is processed just in the region of interest (ROI) area with the certain threshold depth. The range of the threshold is chosen from the distance of z-axis (depth) direction and the value that had been set was within -60 mm to +40 mm. The ROI is used to reject the unwanted object near to the detected hand and it is useful to focus on hand area.

The process diagram shows the initial process of hand gesture recognition and tracking. The procedure starts with hand recognition. The result of the segmentation process is shown in Fig. 3. The work is continuing to detect and recognize the fingertips. At the end, the application is created to prove the method. Numerous researchers performed a segmentation using the depth threshold in a whole image [12, 14, 15]. However, in a previous paper performed an ROI around the detected hand and made the segmentation process became more precise [13].

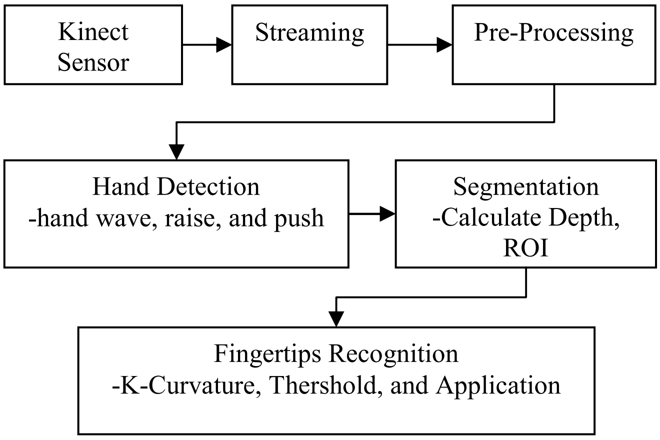
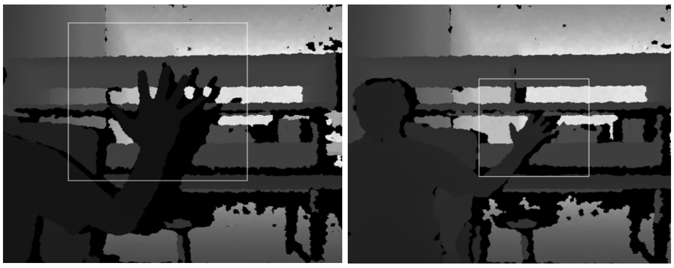


Figure. 2. Process diagram of hand tracking and fingertips recognition



(a) (b)

Figure. 3. Auto resize of ROI (a) Hand is positioned near to the sensor (b) Hand is far away from the sensor

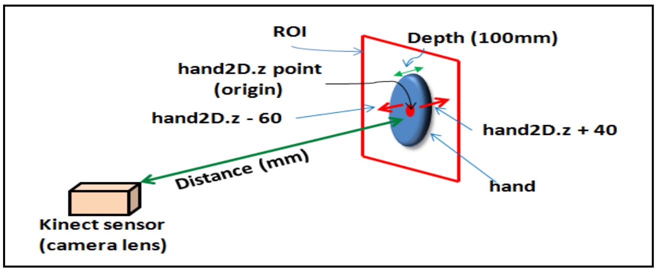


Figure. 4. Illustration of the depth threshold

B. Fingertips Detection

Fingertips detection can be expended to perform an interaction between human and computer. By recognizing a fingertip, user can make a finger as a commander for a further process. This paper presents a method to detect fingertips using the *K* - Curvature method after the hand shape detection that used the depth data. From the previous hand detection and recognition steps, the contour of the hand is executed. After extracting the hand contour, the K-Curvature algorithm is computed. This algorithm is familiar to find fingertips [15,16]. Refer to Fig. 5, this algorithm takes each vector point A (*i*) to its neighbor points *B* and *C* at distance of *K*. Calculation for point *B* and *C* are shown in (1) and (2). *i* is a contour point and *K* is a constant value that obtained from the experiment. This value is tested several times from 10 to 20 for different hand sizes. The final *K* value obtained from this experiment is 15, which is suitable for almost all situations.

$$B = (i + K) \tag{1}$$

$$C = (i - K) \tag{2}$$

The angle α between *AB* and *AC* is calculated from the experiment and it is used as a threshold to distinguish the fingertips and the valleys. In this calculation, the fingertips are defined as a positive curvature while the valleys are negative curvature. The particular point *A* will be recognized as a fingertip when the angle α is smaller or equal to the threshold value. Fig. 6 shows the fingertips that have been marked with small yellow circles and the valleys that marked with small blue circles.

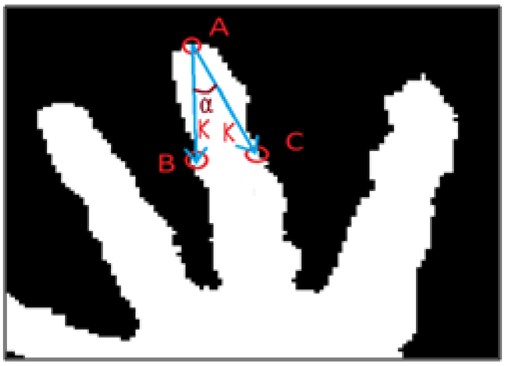


Figure 5. Illustration of K-curvature method

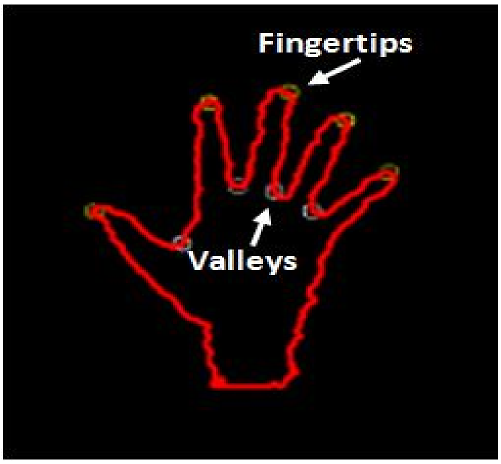


Figure 6. The fingertip (yellow) and the valleys (blue)

IV. Results & Discussion

This method used 3D data which were the depth images from the Kinect sensor. From the results showed that the moving hand can be tracked and the fingertips were successfully detected and recognized. To prove the successfulness of the detection and recognition, an experiment was done by creating an application for fingers.

The experiment was done on finger counting application, where this experiment was carried out to test the capability and efficiency of the proposed fingertips detection. The performance of fingertips detection was tested by applying various hand movements and rotations. The fingertips detection should detect the fingertips even though the hand was moving or rotating.

The obtained results show that the proposed method was successful to recognize the fingertips during the hand movement and rotation. Fig. 7 shows the finger counting application based on the proposed fingertips detection. The experiment is running in various light conditions, which is to verify that Kinect sensor also can capture in low light condition. From the experiment that used finger counting application, the accuracy of fingertips detection can be determined. 15 sets of image frame captured using Kinect were used in this experiment and the data of the experiment is captured after almost 10 second. The details are shown in Table 1. Each set was tested with the finger counting test for the accuracy of the detection and processing time. The results showed that the acceptable average accuracy obtained in this experiment was 73.7% and the average processing time was 15.73 ms.

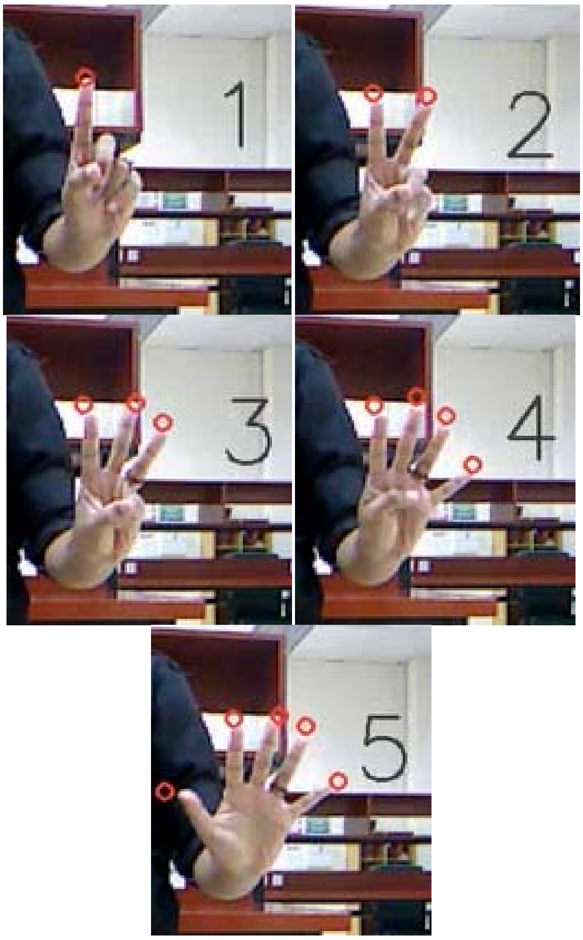


Figure 7. Finger count base on fingertips detection

TABLE I. RESULTS OF THE EXPERIMENT

Set of image frame	Total Frames	Correct Detection (Frames)	Processing Time (ms)	Accuracy (%)
Set1	305	216	16	70.8
Set2	325	256	17.2	78.8
Set3	288	199	15.9	69.1
Set4	312	221	16.7	70.8
Set5	300	210	15.6	70.0
Set6	340	221	17	65.0
Set7	302	226	15.3	74.8
Set8	296	234	14.3	79.1
Set9	322	241	15.3	74.8
Set10	277	221	14.4	79.8
Set11	294	226	14.7	76.9
Set12	321	231	15.1	72.0
Set13	278	208	17.2	74.8
Set14	300	219	15.9	73.0
Set15	311	236	15.3	75.9
Average			**15.73**	**73.7**

After testing the application for the capability and efficiency of the fingertips detection, then this detection program was used in the picture selection application. For this application, there were 4 different shapes (labeled as A, B, C, D) of binary images as shown in Fig. 8. The gestures used for this interaction were finger that pointed according to the number of count 1, 2, 3 and 4. Each detected finger count displayed the image such as shown in Fig. 9. Image A was displayed when the finger was counted as 1, image B for 2 fingers, image C for 3 fingers and image D for 4 fingers. Only one gesture was not used to display the binary image, which was the finger that count for 5. This gesture was used to give an exit command or close the display window and exit the program.

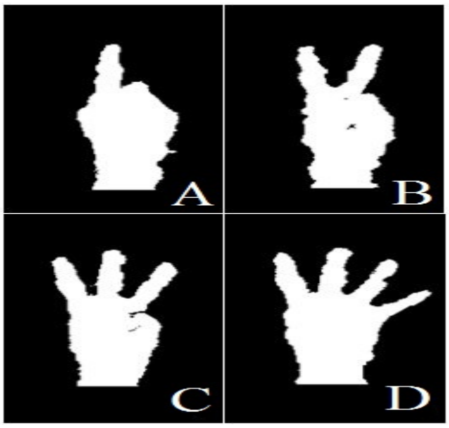


Figure 8. Binary image of the figure gesture

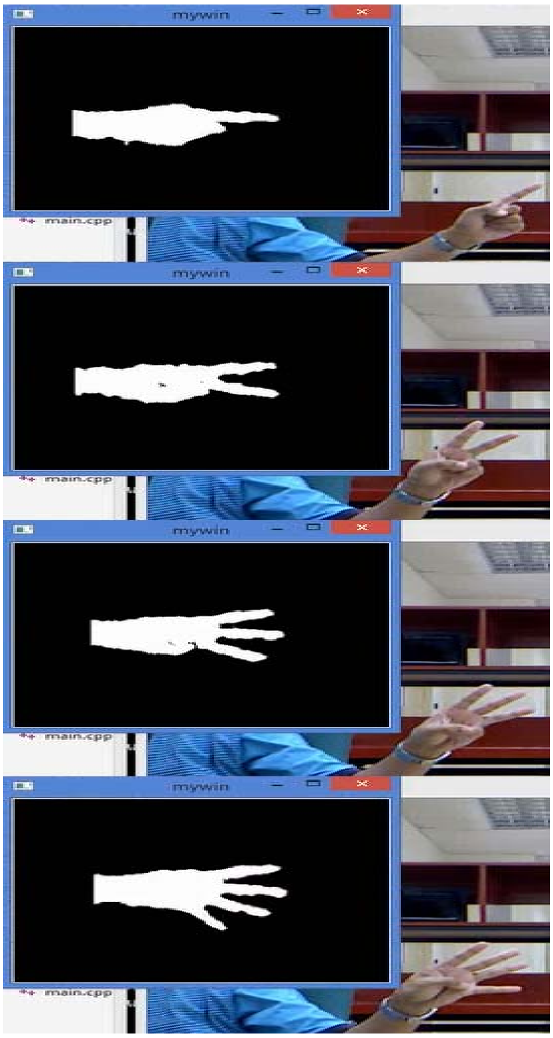


Figure 9. Picture change depends on instruction on the finger

V. Conclusion

This paper presents a finger detection method begin from Kinect sensor. The depth data from Kinect sensor is successfully applied. The hand tracking and fingertips recognition algorithm can be manipulated by the depth threshold and K-Curvature. This method is able to perform in real-time. Further system will be developed using this method, which is rehabilitation system that is able to assist the exercise of the hand. Another application that also can be applied is upper limb detection and motion tracking that suitable for controlling a computer, device or robot.

Acknowledgment

This work is supported by a Fundamental Research Grant Scheme (FRGS) and Faculty of Electrical and Electronic Engineering, Universiti Malaysia Pahang (UMP).

References

[1] T.B. Blair, and C.E. Davis, "Innovate Engineering Outreach: A special application of the Xbox 360 Kinect sensor." In *IEEE Frontiers in Education Conference*, pp.1279-1283, October, 2013.

[2] J. Farooq, M.B. Ali, "Real time hand gesture recognition for computer interaction" on *Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference*, pp. 73 - 77, 22-24 April 2014.

[3] N. Soontranon, S.Aramvith, T.H. Chalidabhongse, "Improved Face and Hand Tracking for Sign Language Recognition" in *International Conference on Information Technology: Coding and Computing (ITCC),* vol. 2, pp.141-146, 2005.

[4] D. Le, Y. Park, "Vision-Based Remote Control System by Motion Detection and Open Finger Counting", *IEEE Transactions on Consumer Electronics*, Vol. 55 , Issue: 4, pp2308 - 2313, Nov. 2009.

[5] Z. Y. Meng, J. S. Pan, K. K. Tseng, W. Zheng, "Dominant Points based Hand Finger Counting for Recognition under Skin Color Extraction in Hand Gesture Control System", *2012 Sixth International Conference on Genetic and Evolutionary Computing (ICGEC)*, pp.364 - 367, 25-28 Aug. 2012.

[6] M. Vanco, I. Minarik, G. Rozinaj,"Evaluation of Static Hand Gesture Algorithms", *2014 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp.83 - 86, 12-15 May 2014.

[7] S. Ganapathyraju, "Hand Gesture Recognition Using Convexity Hull Defects to Control an Industrial Robot", 2013 *3rd International Conference on Instrumentation Control and Automation (ICA)*, pp.63 - 67, 28-30 Aug. 2013.

[8] G. B. Narejo, S. Pervez Bharucha, "Real Time Finger Counting and Virtual Drawing Using Color Detection and Shape Recognition", *2013 5th International Conference on Information & Communication Technologies (ICICT)*, pp.1 - 6, 14-15 Dec. 2013.

[9] G. Feng He, S. Kyung Kang, W. Chang Song, and Sung-Tae Jung, "Real-time Gesture Recognition using 3D Depth Camera" in *IEEE 2nd International Conference on Software Engineering and Service Sciene (ICSESS)*, pp.187-190, July, 2011.

[10] C. Raposo, J.P. Barreto, U. Nunes,"Fast and Accurate Calibration of a Kinect Sensor", on *3D Vision - 3DV 2013, 2013 International Conference*, pp. 342 - 349, 29 June 1 July 2013.

[11] Kenneth David Mankoff, Tess Alethea Russo "The Kinect: a low-cost, high-resolution, short-range 3D camera, on *Earth Surface Processes and Landforms*, Vol 38, Issue 9, pp 926–936, July 2013

[12] K. Rimkus, A. Bukis, A. Lipnickas, and S. Sinkevicius, "3D human hand motion recognition system." in *The 6th International Conference on Human System Interaction (HSI)*, IEEE, pp. 180-183, June, 2013.

[13] R. Samad, M. Z. Abu Bakar, P. Dwi, N. L. Yong Aan, *Real-time Rotation Invariant Hand Tracking Using 3D Data.* on *4th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014),* 28- 30 November 2014.

[14] Z. Ren, J. Yuan, J. Meng, Z. Zhang, "Robust Part-Based Hand Gesture Recognition Using Kinect Sensor", on *Multimedia, IEEE Transactions*, Vol 15, Issue 5, pp. 1110 - 1120, 25 Feb 2013.

[15] Y. Wen, C. Hu, G. Yu, and C. Wang, " A robust method of detecting hand gestures using depth sensors" in *Workshop Haptic Audio Visual Environments and Games (HAVE)*, pp.72-77, October, 2012.

[16] J. Segen,S. Kumar,"Shadow Gestures: 3D Hand Pose Estimation using a Single Camera ", In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.1, pp. 1063-6919, 23 Jun 1999-25 Jun 1999.

[17] S. Malik,"Real-time hand tracking and finger tracking for interaction CSC2503F project report." *Department of Computer Science, University of Toronto, Tech. Rep* (2003).