2016 IEEE 6th International Conference on Advanced Computing

Real Time Hand Gesture Recognition for Human Computer Interaction

Rishabh Agrawal, Nikita Gupta
Department of Computer Engineering, Army Institute of Technology
Savitribai Phule Pune University, Maharashtra, India
rishabhagarwal_12207@aitpune.edu.in, ngupta@aitpune.edu.in

Abstract- **Most of the human computer interaction interfaces that are designed today require explicit instructions from the user in the form of keyboard taps or mouse clicks. As the complexity of these devices increase, the sheer amount of such instructions can easily disrupt, distract and overwhelm users. A novel method to recognize hand gestures for human computer interaction, using computer vision and image processing techniques, is proposed in this paper. The proposed method can successfully replace such devices (e.g. keyboard or mouse) needed for interacting with a personal computer. The method uses a commercial depth + rgb camera called Senz3D, which is cheap and easy to buy as compared to other depth cameras. The proposed method works by analyzing 3D data in real time and uses a set of classification rules to classify the number of convexity defects into gesture classes. This results in real time performance and negates the requirement of any training data. The proposed method achieves commendable performance with very low processor utilization.**

Keywords- **Human Computer Interaction (HCI), gesture recognition, hand gestures, convex hull, convexity defects, computer vision and image processing.**

I. INTRODUCTION

Human Computer Interaction (HCI) is the art of designing interfaces that can be used for communication between humans and computers. HCI with a personal computer today is not just limited to keyboard and mouse interaction. The invention of smartphones has single handedly disrupted the idea of what a computer should be. To put it simply, the modern computer should be 'smart', i.e., it should understand the user naturally without requiring explicit instructions for every action. Gesture based interface on a touch screen and swipe keyboards are the perfect example of a smart interface. They are intuitive and fast and still give the user the full control of the interface without being too confusing or complicated. Speech recognition services like "Google Now" further cement the smartness of these devices.

Desktops sadly have been the subject of keyboard and mouse based interfaces for generations now. The now omnipresent direct manipulation interface that consists of a pointing device and a keyboard was first demonstrated by Ivan Sutherland in Sketchpad, which was his 1963 MIT PhD thesis [11]. Adding touchscreens to laptops is also not solving any problems and they in fact make the interface feel more awkward to use.

Thus there is a need for designing more intuitive interface for interacting with personal computers. Hand gestures are one of the most natural mode of communication among human being after verbal communication. Designing an HCI interface that uses hand gestures can be much more intuitive than traditional methods. This is evident by the commercial success of the Kinect sensor present in Xbox 360 and Xbox One gaming system, which is more focused on body parts recognition and their pose estimation.

A hand gesture recognition system to be able to successfully replace a mouse or keyboard needs to be able to precisely detect each finger and hand orientation in real time and should be robust to various changes in hand measurements, rotation, color and lighting. This is a very complex problem and requires advanced image processing and computer vision concepts. In this paper, a novel method is proposed to recognize hand gestures in real time with high accuracy and precision.

II. RELATED WORK

Karam et al. in his work [21] reported that hand has been widely used in comparison to other body parts for gesturing as it is a natural form of medium for communication between human to human hence can best suited for human computer interaction. Kanniche et al. [21] classifies contact based devices for hand gesture recognition into mechanical, haptic, ultrasonic, inertial and magnetic. Chaudhary et al. [21] recognized the need of different algorithms depending on the size of the dataset and the gesture performed. He also notes that the developed system should be both flexible and expandable which maximize efficiency, accuracy and intuitiveness. Segmenting a hand from a cluttered background and tracking it steadily and robustly are challenging tasks. Wachs et al. [3] discusses soft computing based methods like artificial neural network, fuzzy logic and

978-1-4673-8286-1/16 $31.00 © 2016 IEEE
DOI 10.1109/IACC.2016.93

Fig. 1. Experimental Setup

genetic algorithms in designing the hand gesture recognition. It is effective in getting results where knowing the exact positions of hand or fingers are not possible.

Jacob et al. [3] performs context based gesture recognition for use in operating rooms. Dominio et al. [4] combines multiple depth-based descriptors for hand gesture recognition. Liang et al. [5] detects gestures using depth based features and tracks fingertips using a particle filter. Manresa et al. [6] uses a finite state classifier to identify the hand configuration which is then classified into a gesture class to play video games. Hong et al. [1] performs gesture recognition using a convexity defect histogram. Yao et al. [7] performs contour based hand gesture recognition. Liu et al. [8] performs a fusion of inertial and sensor depth data to recognize hand gestures. Wang et al. [10] performs a super-pixel based hand gesture recognition using a depth camera.

A lot of work on hand gesture recognition via computer vision approaches is being researched using different approaches. However, many of them are unnecessarily complex approaches which either require a supercomputer for training or are very taxing on the processor that they cannot be used in a real-world/real-time scenario. Many of them are also not able to accurately and precisely localize the hand position and contour in an image in presence of a complex background. Many of them also restrict the motion of hand and may also be marker based making them semi-automatic, inconvenient, uncomfortable and difficult to use. Even the algorithms used for the analysis of the contour have limited performance both in terms of the number of gestures recognized and response time. In most of these papers, a Kinect sensor is used to capture the depth information which is very bulky and costly as compared to the Senz3D camera.

In this paper, a novel method based on computer vision is proposed for the automatic and precisely detection of a hand, accurately detect its contours and give a complete analysis for it using a set of algorithms to detect fingers, arm, and gestures without using any kind of markers or training data. It is also invariant to rotation and lighting. All this is done in real time at 30 frames per second. The proposed method gives commendable results in both experimental and real word hand scenarios.

III. THE PROPOSED METHOD

This section describes the framework to detect and track palm region and recognize gestures. The proposed method uses a Creative Senz3D camera to capture both color and depth information, and is implemented using OpenCV API in C++ language.

A. Hand Detection and Tracking

The Senz3d camera captures a RGB video frame along with the associated depth data. Depth based thresholding is performed to remove the background. Then segmentation based on depth data is performed for the object closest to the camera.

$$P = \{(x, y) : d_i \leq D(x, y) \leq d_i + c\} \quad (1)$$

Here, P is the set of pixels (x, y) that may represent the hand region. D(x, y) represents the depth of pixel (x, y) from the camera. This may also include pixels that belong to the arm region. A color based filtering is performed on these pixels to check if these actually represent the hand pixels based on a predefined color model. If these are not recognized to belong to the hand region, then the algorithm waits for the next frame.

$$P' = \{(x, y) : I(x, y) \in hand\ color\ model\} \quad (2)$$
$$S = P \cap P' \quad (3)$$

Where, $P^{'}$ is the set of color filtered pixels and I(x, y) represents the intensity of pixel (x, y.), S are the pixels recognized as hand pixels, then the back projection of the detected region is passed on to mean-shift algorithm to track the hand region. This provides us with the hand mask.

B. Hand Region Analysis

This consists of first segmenting arm from the palm region of the hand. To achieve this, first the fingers are eroded away from the hand mask and is the palm mask is created. This is

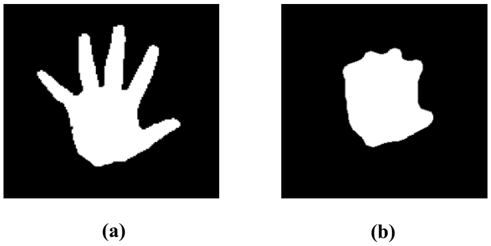

Fig. 2. (a) Hand Mask, (b) Palm Mask

done by successive erosion and dilation morphological operations. Once the fingers are eroded away, we need to detect the palm next. If we generalize the shape of the palm, it can be approximated to a square. So the maximum width of this region is found using a rotated rectangle of minimum enclosing area that can entirely fit the contour of palm mask. This contour is found using simple chain approximation.

$$d = \left(\frac{h}{2}\right) - \left(\frac{w}{2}\right) \qquad (4)$$

$$x' = x - (d * cos\theta) \qquad (5)$$

$$y' = y - (d * sin\theta) \qquad (6)$$

Where, d is the offset by which the center by which the center of the rotated rectangle R_1 (green, Fig.3. (a)) needs to be moved, h and w represent its height and width respectively, (x, y) represents its center, θ is the angle which the height of the rectangle R_1 makes with the positive x axis and (x, y) is the new center for square R_2 (red, Fig.3. (a)) with side of length w. This center can be seen as the small green dot in Fig.3. (b).

C. *Gesture Recognition*

Gesture recognition module is the core part of the proposed method. This starts by first creating a closed contour C_1 for the hand mask. This is done using simple chain approximation. A vector V_1 representing the orientation of the hand in the image plane is created.

$$V_1 = (x_2 - x_1)\hat{\imath} + (y_2 - y_1)\hat{\jmath} \qquad (7)$$

Where, (x_1, y_1) is the center of rectangle R_1 and (x_2, y_2) is center of rectangle R_2. A vector V_2 perpendicular to V_1 is also created. A line L_1 passing through the center of the points (x_1, y_1) and (x_2, y_2) with direction of V_2 is created, and is used to create a contour C_2 which consists of points that lie above L_1.

$$C_2 = \{(x, y): m * x - y + c \geq 0\} \qquad (8)$$

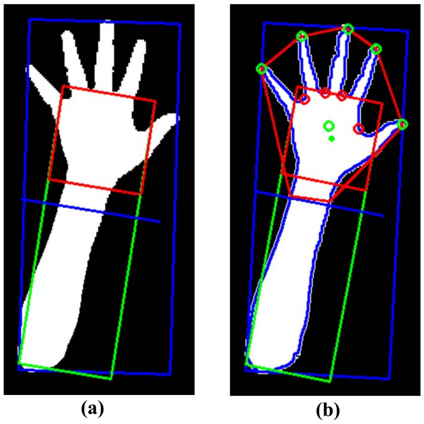
(a)
(b)

Fig. 3. (a & b) Hand Region Analysis

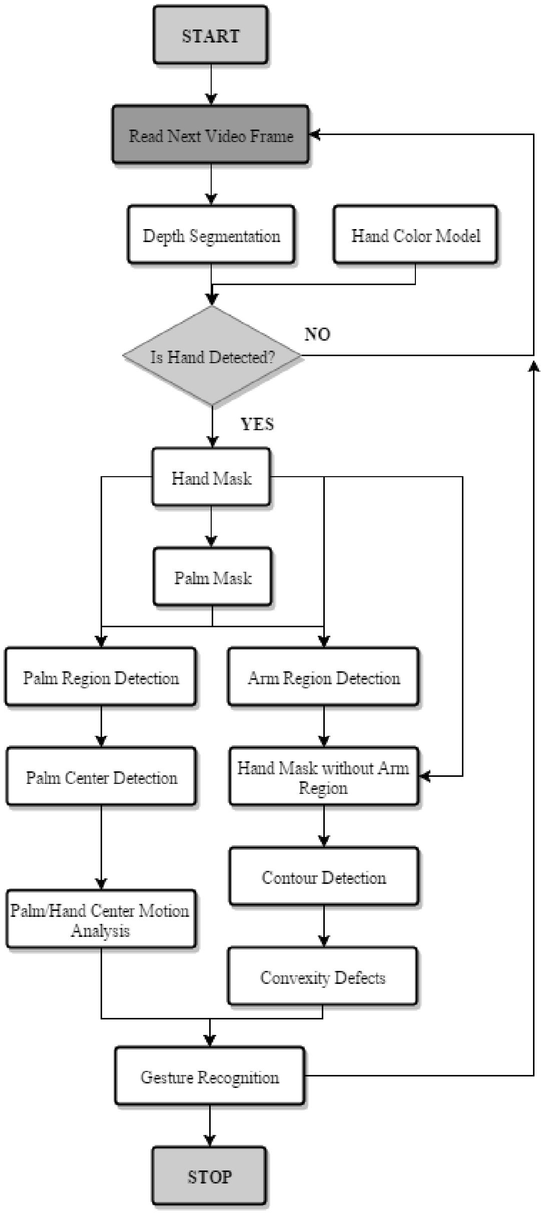

Fig. 4. Flowchart for the Proposed Method

Where, $y = m * x + c$ is the equation of line L_1. The next step is to find the convexity defects. For this, first a convex hull H_1 is created for the hand contour C_2. Using this convex hull, convexity defects D_1 are calculated. Each convexity defect consists of a start and end point, its location on the contour C_2 and its size. Using linear regression, a function is found between the distance of the center of the hand from the camera and the maximum size of the convexity defects to detect all open fingers when the hand is at different distances from the camera. Using the predicted maximum size as a threshold (T_1) for the convexity defects, all convexity defects in D_1 which are greater than T_1 are considered as valid convexity defects D_2.

To track and count the number of fingertips, we need to analyze the convexity defects D_2. The convexity defects are taken in an anticlockwise direction starting from the bottom leftmost point. Both the starting and ending points are recorded for this point in a list L_1, and for subsequent points, only the starting points are recorded in the list. Now, we need to check whether these points are actually fingertips or not. For this, the distance D' of the fingertip from the center (x', y') of the hand should be greater than twice the distance of the distance of the corresponding convexity (x_d, y_d) defect from the center (x', y'). If this condition holds, the valid fingertips are added to the list L_2, and its size is counted, which gives us the number of open fingers. The combination of this number along with the motion of the mass center of the contour C_2 is then translated into a desired gesture.

$$D'(x,y) > 2 * D(x_d, y_d) \tag{9}$$

$$D'(x,y) = \sqrt{(x - x')^2 + (y - y')^2} \tag{10}$$

In Fig.5, the green points on the convex hull represent the detected fingertips, the red points represent their respective convexity defects. These points are detected in real time and are analyzed to recognize the gestures which can be programmed to carry out an action for a HCI interface.

IV. EXPERIMENTAL SETUP AND RESULTS

To evaluate the results we simulate an environment in which a user uses hand gestures to control all the mouse functions, i.e. the mouse is completely emulated using hand gestures. This enables us to measure the precision and accuracy of tracking hand and recognizing the hand gestures in real time. All the experiments are performed on a standard laptop with four gigabytes of memory and a third generation Intel Core i5 processor. The setup requires no special calibrations or controlled lighting and can work in most environments without any problems.

The recognition rate is used as a parameter to quantitatively measure the performance of gesture recognition using the proposed method. Recognition rate (RR) for a gesture is defined as the ratio of the number of video frames in which the gesture was correctly recognized to the total no of video frames in which the gesture was tested. We calculate Static Recognition Rate (RR_s) when there is no or small relative motion between the camera and the hand and Dynamic Recognition Rate (RR_d) when there is significant motion between the camera and hand.

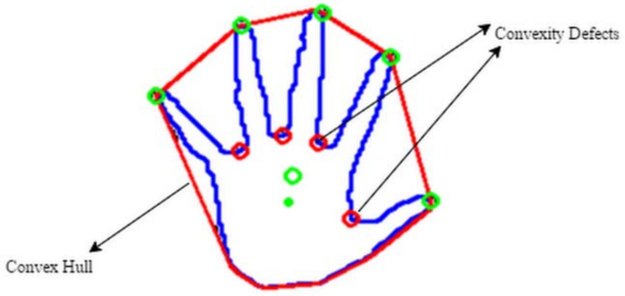

Fig. 5. Convex Hull and Convexity Defects

$$RR = \frac{Frames\ correctly\ identified}{Total\ Frames\ for\ gesture} \tag{11}$$

TABLE I. GESTURES AND THEIR MEANINGS

Gesture Class	No of Fingers	Meaning
1	0	Grab/Move Mouse
2	1	Single Click
3	2	Double Click
4	3	Right Click
5	4	Mouse Hold Toggle
6	5	Release Mouse

TABLE II. GESTURE CLASSES AND THEIR RESPECTIVE NUMBER OF CONVEXITY DEFECTS

Gesture Class	No of Convexity Defects
1	0
2	0,1
3	1,2
4	2,3
5	3
6	4

Table 1 and Table 2 establish the relations between the gesture classes, the number of convexity defects the gestures have, the number of open fingers they represent and the actions they perform. The test frames were taken from 5 different individuals with varying hand sizes, in different lighting conditions and at different distances from the camera. These people used the program to interact with the computer instead of a regular mouse.

Table 3 and Table 4 give the gesture class specific and the overall recognition rate for both static and dynamic gesture recognition scenarios. As we can see, we get commendable performance in real time with average of only 8% CPU utilization.

TABLE III. STATIC RECOGNITION RATE OF DIFFERENT GESTURE CLASSES

Gesture Class	No of Test Frames	Successful Recognition	Recognition Rate*100(%)
1	300	300	100.0
2	300	293	98.33
3	300	300	100.0
4	300	300	100.0
5	300	300	100.0
6	300	286	95.33
Total	1800	1779	98.83

TABLE IV. DYNAMIC RECOGNITION RATE OF DIFFERENT GESTURE CLASSES

Gesture Class	No of Test Frames	Successful Recognition	Recognition Rate*100(%)
1	300	296	98.66
2	300	289	96.33
3	300	299	99.66
4	300	273	91.0
5	300	294	98.0
6	300	251	83.66
Total	1800	1702	94.55

V. RESULT ANALYSIS

The proposed method is able to perform in real time at 30 frames per second, achieving high recognition rate while utilizing minimal processing power. The proposed method is also invariant of rotation, lighting conditions and can also work in absolute darkness. It achieves a very high recognition rate. The performance can be made even smoother by using a camera that records video at a higher resolution and frame rate combined with a latest generation processor. The proposed algorithm also has lower latency in recognizing gestures as compared to the gesture recognition demo of Intel Perceptual SDK.

In most cases, the proposed method consistently produces better results than the existing vision based hand gesture recognition in terms of recognition rate and response time. It also has the advantage that it is rotation invariant, performs automatic arm and palm region segmentation and does not require any kind of markers and is also invariant to lighting conditions to such a degree that it can also work in total darkness.

In Fig.6. and Fig.7., we have compared the classification accuracy of the proposed method with the methods proposed in [1] and [2] respectively. As we can infer from the figure, the proposed method always gives better results for the tested gestures.

VI. CONCLUSIONS AND FUTURE WORK

The proposed method for real time hand gesture recognition produces commendable results with high accuracy and precision and can be used in a real world scenario for interaction with a computer. It is much more intuitive to use than a mouse and can detect the fingertips accurately and has great potential for extension to other HCI applications.

Future work will be based on increasing the accuracy of hand detection, increasing its range and performing real time motion analysis and three dimensional pose estimation of hand with robust fingertip tracking.

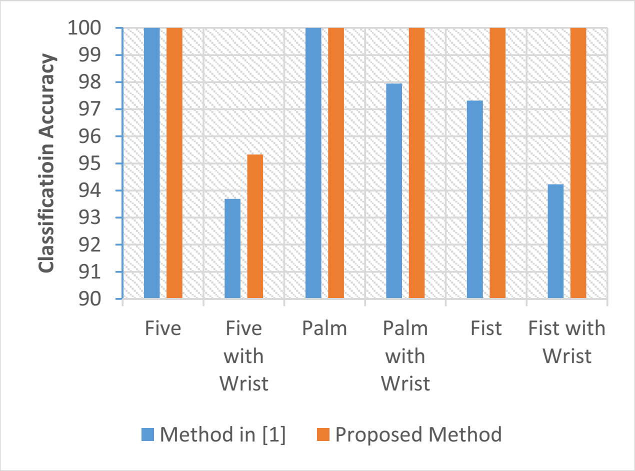

Fig. 6. Classification Accuracy for the proposed method wand method proposed in [1] for static gestures

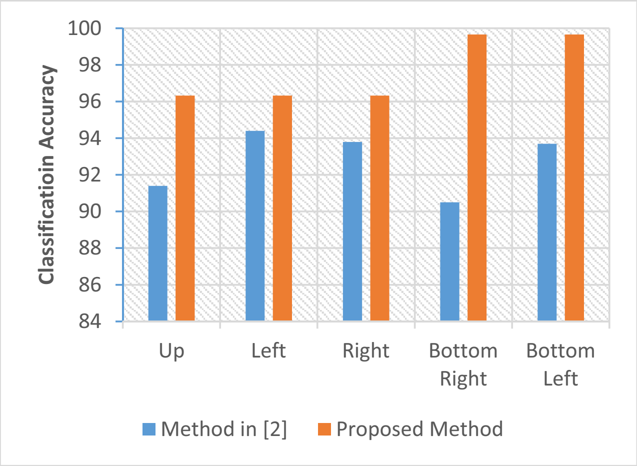

Fig. 7. Classification Accuracy for the proposed method wand method proposed in [2] for dynamic gestures

REFERENCES

[1] Juhyeon Hong; Eung Sup Kim; Hyuk-Jae Lee, "Rotation-invariant hand posture classification with a convexity defect histogram," in Circuits and Systems (ISCAS), 2012 IEEE International Symposium on , vol., no., pp.774-777, 20-23 May 2012, doi: 10.1109/ISCAS.2012.6272153

[2] Shuxin Qin, Xiaoyang Zhu, Yiping Yang, Yongshi Jiang, "Real-time Hand Gesture Recognition from Depth Images Using Convex Shape Decomposition Method," in Journal of Signal Processing Systems January 2014, Volume 74, Issue 1, pp 47-58, DOI: 10.1007/s11265-013-0778-7

[3] Mithun George Jacob, Juan Pablo Wachs, "Context-based hand gesture recognition for the operating room," in Pattern Recognition Letters, Volume 36, 15 January 2014, Pages 196-203, ISSN 0167-8655

[4] Fabio Dominio, Mauro Donadeo, Pietro Zanuttigh, "Combining multiple depth-based descriptors for hand gesture Recognition," in Pattern Recognition Letters, Volume 50, 1 December 2014, Pages 101-111, ISSN 0167-8655

[5] Hui Liang, Junsong Yuan, and 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, 785-788. DOI=10.1145/2393347.2396312

[6] Manresa-Yee, Cristina, Javier Varona, Ramon Mas, and Francisco J. Perales, "Hand tracking and gesture recognition for human-computer interaction," in Electronic letters on computer vision and image analysis 5, no. 3 (2005): 96-104.

[7] Yuan Yao; Yun Fu, "Contour Model-Based Hand-Gesture Recognition Using the Kinect Sensor," in Circuits and Systems for Video Technology, IEEE Transactions on , vol.24, no.11, pp.1935-1944, Nov. 2014, doi: 10.1109/TCSVT.2014.2302538

[8] Kui Liu; Chen Chen; Jafari, R.; Kehtarnavaz, N., "Fusion of Inertial and Depth Sensor Data for Robust Hand Gesture Recognition," in Sensors Journal, IEEE , vol.14, no.6, pp.1898-1903, June 2014, doi: 10.1109/JSEN.2014.2306094

[9] Ohn-Bar, E.; Trivedi, M.M., "Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations," in Intelligent Transportation Systems, IEEE Transactions on , vol.15, no.6, pp.2368-2377, Dec. 2014

[10] Chong Wang; Zhong Liu; Shing-Chow Chan, "Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera," in Multimedia, IEEE Transactions on , vol.17, no.1, pp.29-39, Jan. 2015, doi: 10.1109/TMM.2014.2374357

[11] Myers, Brad A, "A brief history of human-computer interaction technology," in interactions 5, no. 2 (1998): 44-54.

[12] Jeong, Jinwoo, and Yoonhee Jang, "Max–min hand cropping method for robust hand region extraction in the image-based hand gesture recognition," in Soft Computing (2015) 19:815–818, DOI 10.1007/s00500-014-1391-9, Springer-Verlag Berlin Heidelberg 2014

[13] Hasan, Haitham, and Sameem Abdul-Kareem, "Human–computer interaction using vision-based hand gesture recognition systems: a survey," in Neural Computing and Applications (2014) 25:251–261, DOI 10.1007/s00521-013-1481-0, Springer-Verlag London 2013

[14] Hasan, Haitham, and S. Abdul-Kareem. "Static hand gesture recognition using neural networks." Artificial Intelligence Review 41, no. 2 (2014): 147-181, DOI 10.1007/s10462-011-9303-1, Springer Science+Business Media B.V. 2012

[15] Rautaray, Siddharth S., and Anupam Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," Artificial Intelligence Review 43, no. 1 (2015): 1-54, DOI 10.1007/s10462-012-9356-9, Springer Science + Business Media Dordrecht 2012

[16] Yin, Liang, Mingzhi Dong, Ying Duan, Weihong Deng, Kaili Zhao, and Jun Guo, "A high-performance training-free approach for hand gesture recognition with accelerometer," Multimedia tools and applications 72, no. 1 (2014): 843-864, DOI 10.1007/s11042-013-1368-1, Springer Science+Business Media New York 2013

[17] Molchanov, Pavlo, Shalini Gupta, Kihwan Kim, and Kari Pulli, "Multi-sensor system for driver’s hand-gesture recognition," in IEEE Conference on Automatic Face and Gesture Recognition, pp. 1-8. 2015.

[18] Frati, V.; Prattichizzo, D., "Using Kinect for hand tracking and rendering in wearable haptics," in World Haptics Conference (WHC), 2011 IEEE , vol., no., pp.317-321, 21-24 June 2011, doi: 10.1109/WHC.2011.5945505

[19] Fritz, Daniel, Annette Mossel, and Hannes Kaufmann, "Evaluating RGB+ D hand posture detection methods for mobile 3D interaction," in Proceedings of the 2014 Virtual Reality International Conference, p. 27. ACM, 2014.

[20] Yadav, Kapil, and Jhilik Bhattacharya, "Real-Time Hand Gesture Detection and Recognition for Human Computer Interaction," in Intelligent Systems Technologies and Applications, pp. 559-567. Springer International Publishing, 2016.

[21] Rautaray, Siddharth S., and Anupam Agrawal. "Vision based hand gesture recognition for human computer interaction: a survey." Artificial Intelligence Review 43, no. 1 (2015): 1-54., DOI 10.1007/s10462-012-9356-9 , Springer Science + Business Media Dordrecht 2012