Real-time efficient detection in Vision Based Static Hand Gesture Recognition

Amrutnarayan Panigrahi, Jaganath Prasad Mohanty, Ayaskanta Swain, Kamalakanta Mahapatra
Electronics and Communication Engineering Department
National Institute of Technology, Rourkela, India
vlsi.nitrkl@gmail.com, kmaha2@gmail.com

***Abstract*—**

The focus on Human-Computer Interaction (HCI) research is increasing day by day, due to the increasing requirement of intelligent input devices. Hand Gesture Recognition is a small sub-field but presents a significant number of applications and consumer products. Most researches target on the feasibility of recognition systems but give less weight to the device resources, so the cost and time. The time-consuming complicated algorithms' use is limited to special purpose devices such as expensive gaming consoles. The use of such systems in low cost embedded hardware in real-time circumstances is required, with the comfortability to use it. In this paper, we design an efficient real-time keyboard-like HCI using Static HGR. We have proposed and implemented new methods to reduce the time consumption while maintaining the high accuracy of 90% with scale and rotation invariance. Also, to maintain the comfort of use, we have eliminated complicated gestures and used only 11 gestures as input gesture set.

Keywords—gesture recognition, top hat transform, reference direction, wrist identification, static HGR, Kinect sensor

I. INTRODUCTION

Human Computer Interaction (HCI) is an emerging topic of research. With the invention of new technologies, the demand of effortless interaction systems is increasing more than ever. Hand Gesture Recognition (HGR) has a major importance in this field, because hand is the primary mode for voiceless interaction. The applications of HGR are vast; from sign language to virtual reality and touch less control.

There are various approaches for Hand Gesture Recognition as shown in Fig. 1. Vision Based HGR is dominant in applications because of its less resistance to user. Any hand gesture is usually a combination of static and dynamic gestures. To use HGR as command interface, we use static gestures to convey the information. It is a good option for user-comfort in case of frequent use. The methods for Static HGR categorized into two groups: *region and contour based*. Region based methods extract features from hand region. But contour-based methods extract feature from contour of the hand. It is noted that contour-based methods require lesser computations than region-based ones [1].

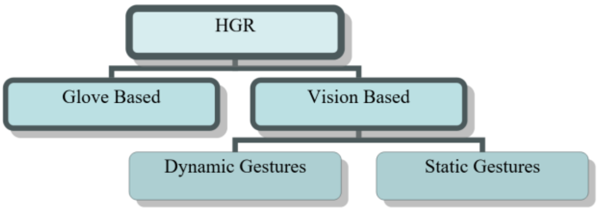


Figure 1 Various approaches for HGR

Various researches have used either RGB image or depth images for gesture acquisition. The major disadvantage of using RGB image is that the results are sensitive to lighting conditions [8]. Also, detection of other objects such as face, in skin color detection is a problem. To prevent this problem, wearing of sleeves or face subtraction is used in some cases [8] [5]. So, depth image is widely considered for HGR, for to its robustness in lighting conditions [18] [19].

After acquisition of gesture, feature extraction can be done by various methods, such as Shape Context (SC) [13], Time Series Curve [3] [4], Top-Hat Transform [16], Convex Hull [2] [5] [14], HOG [7] [11], SIFT [10], SURF-BOF [9], Discrete Curve Evolution method [15], finger emphasizing multi-scale description [12] etc. Then the classification of features can be done using manual classification like if-else, FEMD [3] [4], SVM [5] [1], MLP [7] or BPNN [12] etc. For simple features, simpler algorithms such as FEMD, can be used. However, features obtained from HOG, SIFT or SURF are highly complex to be directly interpreted. In this case, machine learning is implemented. To reduce a large feature set into a smaller one, methods such as Lasso algorithm [4], PCA [7] etc. are used. The larger the feature set or the complexity of feature, the more calculation it requires. Proportionally time consumption also increases. Contour-based methods such as Convex Hull give less number of features and consume less time to execute with reliable accuracy [2]. However, the major problem with Convex Hull is that sometimes it mistakenly takes the hand for another finger, which results in classification error [2]. Without taking any predefined setup such as wearing sleeves, the need to eliminate the hand-part arises. In addition, Top-Hat transform, although region based, converts the input into smaller feature set.

In both cases, the orientation (and direction) of fingers is taken as features, which is very sensitive to rotation. Most researches define the gesture to be in certain direction [18], restricting the user to perform with constraints. Although most methods are scale invariant, the need for consistency of HGR with rotation is required.

Shin et al. combined static and dynamic gesture recognition to develop a complex gesture-keyboard [6]. By adding dynamic gesture into static gesture recognition, the reliability of the system is improved because it would eliminate random noisy inputs from background or from hand in idle state. Moreover, a portable and real-time system is preferred in design of such systems [1] [16].

While using Kinect Camera, a hand easily affected by segmentation errors as compared to entire human body, because of its more complex articulations in a smaller area. Most methods use many complex background subtractions to isolate the hand gesture [8] [17] [18]. We propose a new method to isolate the gesture by isolation of space in a 3D image.

In this work, we have implemented HGR application by using depth data from Kinect sensor. The algorithm used is Top-Hat transform combined with Convex Hull to increase the accuracy of HGR and for a smaller feature set (contour-based method) to reduce the number of calculations and make it real time. Here, three novel approaches are used. The confusion of hand with finger is eliminated by extracting data from local curvature, the importance of which is emphasized in [15]. In addition, the rotation invariance of gesture and spatial background subtraction is proposed and implemented.

II. Methodology

A real time scenario of few gesture recognition techniques observed and reported using proposed algorithms below. The block diagram of the HGR system given in Fig. 2.

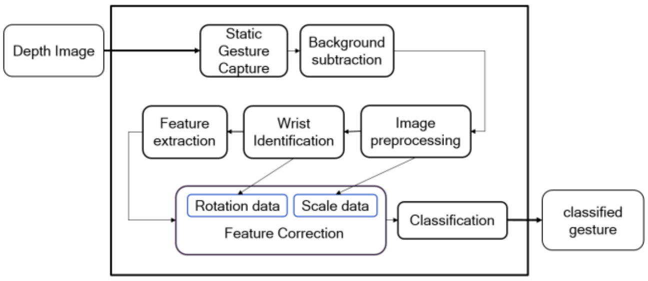


Figure 2 Block Diagram of the HGR system

A. Static gesture acquisition and spatial BG subtraction

The gesture we perform in front of any camera for Gesture Recognition is dynamic in nature. In this work, the static gesture is defined to be performed, only when performed in a specific way, such as a 'click'. By tracking the hand using OpenNI library, we get the position of the hand as (x, y, z), which are the respective Cartesian coordinates keeping camera at origin faced upwards.

For background subtraction, we just isolate the hand in 3D space, by taking a spatial cube of side almost 20 cm with hand position at its center. This also results in the assumption that hand must not be close to other objects while performing the gesture (Fig. 3).

Another interesting result from the thresholding, described above is that it reduces the bit length of pixel values. Originally, the depth values are in mm ranging from 0 to 65535, i.e. 16 bits. The image having 16-bit values is unsuitable for various image-processing functions such as edge detection and distance transform and their use implies double the RAM usage. By thresholding, we get a useful range of 200mm values 0-200, which then converted to 8-bit values, resulting in same sensitivity in depth.

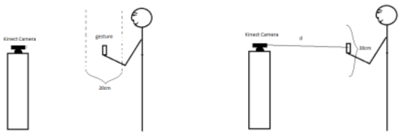


Figure 3 Background Subtraction by depth thresholding and cropping

B. Image Preprocessing

To convert the gesture image ranging from 0-255, to a binary image, we apply threshold operation. However, the output of Kinect is very noisy, so there are holes inside the hand BLOB and small BLOBs outside it as in fig 4(ii), in most images. In morphological operations, these imperfections present quite a problem.

To avoid these kinds of problems, we propose the following method. First, the biggest contour in the image is extracted, which usually represents the hand. In a blank image the contour is drawn. Then, image-filling process done from a suitable point such as the palm center (Fig 4) or left-top corner. Then we get a binary image of only the hand gesture.

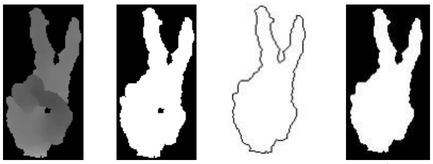


Figure 4 Input image with hole, the hole inside BLOB after binary image conversion, Contour drawn in blank image and Noise free Silhouette

To obtain the center of the palm, which taken as reference point, we take the center of maximum inscribed circle inside the gesture (Fig 5). Alternatively obtained by distance transform using Euclidian distance as distance metric and taking the location of maximum value.

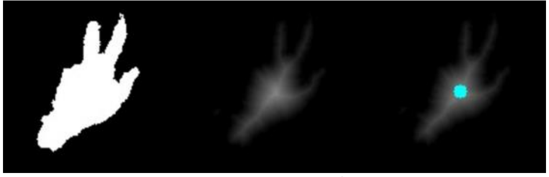


Figure 5 Input binary image, distance transform and maxima highlighted as center of palm

C. Wrist Identification or Hand Elimination

To make the HGR invariant to rotation, we need a reference direction to which all directions adjusted. The wrist is a universal reference in all gestures. In addition, the presence of hand part in gesture can sometimes be mistaken as one of the fingers. In some gestures, the presence of more fingers does not require the reference direction, but it is crucial when there is only one finger. It is better to eliminate the hand part. We propose an algorithm to extract the wrist points as given below.

Algorithm 1 (Obtain wrist points)

1. Obtain the radius of all points on the hand contour and smoothen it by a low pass filter.
2. Initialize P1 to farthest point on hand-contour and then move in one direction (by next or previous point on contour) and set counter as zero.
3. If the radius decreases, increment the counter.
4. If the monotonic increase in radius becomes enough (by comparing to a value), check if P1 is on the boarder of image or counter value.
5. If P1 is on the boarder or the counter value isn't enough then reset the monotonic increase record of radius and the counter.
6. If P1 is not on boarder and counter value is high, then break the loop. P1 is one of the wrist points.
7. Move to next point and repeat step 3
8. Repeat the steps 2-7 for wrist point P2.

The above algorithm visualized by looking at the radius plot in fig 6.

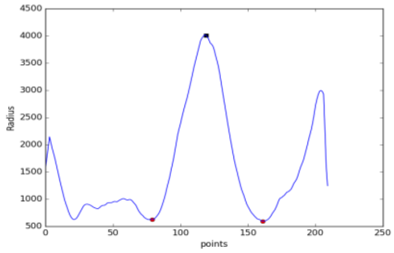


Figure 6 Radius Plot of contour points for wrist identification

Our goal is to obtain the local minima points already highlighted by blue dots in fig 7.

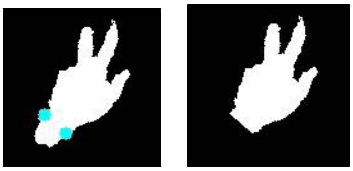


Figure 7 Highlighted Wrist Points and gesture without hand part

D. Feature Detection and Extraction

The features in the gesture, fingers in this case, obtained by two ways: Top Hat Transform and Convex Hull both.

Top Hat Transform is a region-based method. It is a morphological operation, which extracts the small-sized details and elements by eliminating the bigger homogeneous areas. Then the contour extracted, and their enclosing areas are compared with a threshold to determine whether they represent a finger or not.

The finger BLOBs obtained from Top-Hat transform converted to contour and the point on each finger BLOB closest to the palm center taken as feature points. Since this feature point assumed to connect to palm, only the angles of feature points and the number of such contours taken as features (fig 8). Thus, for Top-Hat Transform, the feature vector has a length of 6.

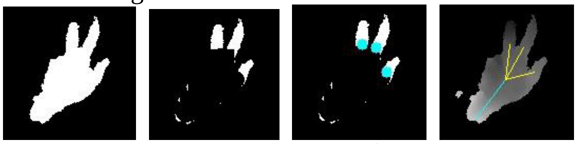


Figure 8 Input binary image, Top Hat Transform, Feature Points and Features as angle between yellow and blue lines

Convex Hull method is a contour-based method. The Convex Hull directly extracted from the contour and then the convexity defects. For each convexity, we get starting points and ending points as fingertips, farthest points or defect point as region between finger regions and the depth of defect; all in the form of Cartesian 2D coordinates. The parameters converted to polar coordinates normalized by scale. The major steps illustrated in fig 9.

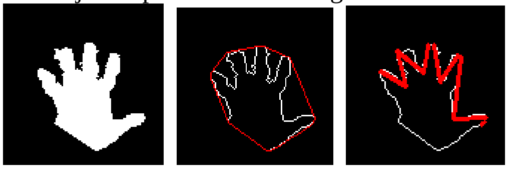


Figure 9 Input Binary Image, Convex Hull (red) and Convexity Defects illustrated in image

E. Scale and Rotation Correction

The default scale of a hand is the radius of maximum inscribed circle inside the palm. In code, it obtained by the peak value of distance transform. For scale dependent processes such as erosion, the kernel or the parameter taken proportional to the palm radius.

Since angle of fingertips also taken as a feature, the angles corrected with respect to rotation. The correction proposed done by subtracting the angle of the center of wrist. The centroid obtained from P1 and P2. The illustration of the method shown in fig 10.

F. Classification

In many research works, machine learning used to classify data from feature vectors [1] [5] [7] [12]. The popular ones are SVM, BPNN, ANN, RBFNN etc. They use supervised learning for training. A feature vector of constant length is taken, by adding trivial values in case of no fingers.

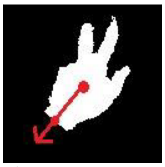


Figure 10 Reference Direction in a gesture

Supervised learning require a large sample set to perform accurately. Also, if the number of gesture commands are less, there is no need for going to high complexity algorithms, which would require more computations. In this aspect, If-else classification is a better alternative. But this method, requires the logic to be designed manually and the parameters to be hardcoded into the module.

III. IMPLEMENTATIONS AND RESULTS

A. Software and hardware

Kinect is a depth-sensing camera. There are three software interfaces for Kinect sensor: Kinect SDK, OpenNI and libfreenect [19]. For image processing and neural network implementation, OpenCV and MATLAB are viable options [17]. For convenience of use in python programming and considering real-time usage, we have used OpenCV and OpenNI v1.5.

In this work, 11 gestures were chosen for comfortability of user while performing. The samples were obtained from 7 people, each performing each gesture at least 10 times. At last, a dataset of 1730 images created. The programs were written in python and was implemented as a real time interface in Linux PC having Intel i3 1.7GHz as well as Raspberry Pi 3B+ 1.2GHz.

The classification of features done by two ways. I. If-else classification: output features of Top-Hat Transform used and the logic manually designed. II. BPNN (Back Propagation neural network): features of convex hull, top hat transform considered. A few scenarios of real time gestures captured, processed and detailed in fig 11 and 12.

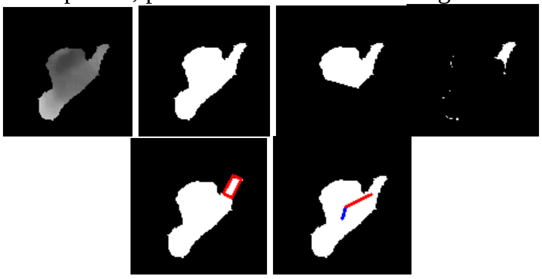


Figure 11 Scenario 1: Input image containing gesture, Binary image, image with hand eliminated, Top-Hat Transform, Detected Finger highlighted in image and the blue & red lines represent the feature angle

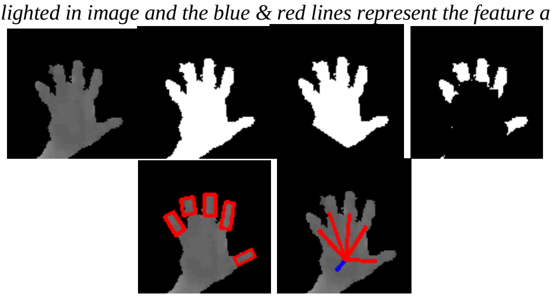
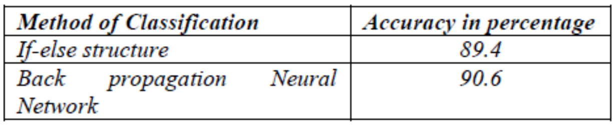


Figure 12 Scenario 2: Input image containing gesture, Binary image, image with hand eliminated, Top-Hat Transform, Detected Finger highlighted in image and the blue & red lines represent the feature angle

B. *Results and Discussion*

For each gesture, out of 1730 gestures, the input label and output of classification plotted in a Confusion Matrix. In table 1, we show the accuracy results of detecting the gestures out of the performed captures using the mentioned setting.

Table 1 Accuracy of methods used



Method of Classification	*Accuracy in percentage*
If-else structure	*89.4*
Back propagation Neural Network	*90.6*

A careful observation of gesture images shows that a lot of feature detection error was because of the noisy output of Kinect sensor and the assumptions laid down upon the algorithm. The Kinect sensor has slow response for depth data than RGB data. The shadowing of objects in depth data, also gave rise to erroneous images, which ultimately contribute to misclassification.

The main assumption of the gesture was that the palm surface in the gesture is normal to the axis of camera. Many gestures, which performed slightly, bent to the camera axis, were misclassified.

The classification accuracy in both cases was almost 90%. This implies that for the given set of gestures the features the cause of error is the same, this might be because of the feature detection algorithm or the noises in the output of Kinect.

The dynamic gesture to static gesture conversion was indeed successful. The number of erroneous captures were only 2 out of 1730, making the probability almost zero.

In table 2, it is observed that our method achieves real time criteria with an execution time of 5ms and a nominal accuracy because of the reasons explained above. The method used in [12] requires sleeves and clear background, but in this work this restriction is completely removed. The complete removal of background and hand with minimal time consumption validates the efficiency of the algorithm.

Table 2 Accuracy and Efficiency Comparison

Methods of HGR	Accuracy	Time consumption
TSC [3]	90.6 %	500 ms
Convex Hull [5]	96.8 %	41.6 ms
Top hat + RGB [16]	95.5 %	143 ms
DCE [15]	97.5 %	56.8 ms
MSWHCD [1]	97.1 %	32.86 ms
FMD+BPNN [12]	99 %	0.94 ms
Top-Hat + Convex (proposed)	90 %	5 ms

IV. Conclusion

In this work, new and efficient methods for gesture recognition through background subtraction, hand elimination and rotation invariance were proposed to remove constraints on gesture input. The Static HGR Algorithm was implemented on a large dataset and an accuracy of 90% observed. In capturing the correct gesture from video stream, the accuracy was 99.88%. The algorithm can run with a frame rate of 30 fps, with a comparative real-time performance to be used as lightweight applications or embedded systems.

ACKNOWLEDGMENT

This work is supported by Ministry of Electronics and Information Technology (MeitY), Govt. of India under SMDP-C2SD (Special Manpower Development Programme for Chips to System Design) Project.

References

[1] Yiyi, R. E. N., et al. "Hand Gesture Recognition with Multi-Scale Weighted Histogram of Contour Direction (MSWHCD) Normalization for Wearable Applications." IEEE Transactions on Circuits and Systems for Video Technology (2016).

[2] Agrawal, Rishabh, and Nikita Gupta. "Real time hand gesture recognition for human computer interaction." Advanced Computing (IACC), 2016 IEEE 6th International Conference on. IEEE, 2016.

[3] Ren, Zhou, Junsong Yuan, and Zhengyou Zhang. "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera." Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011.

[4] Liu, Honghai, et al. "A novel approach to extract hand gesture feature in depth images." Human Motion Sensing and Recognition. Springer, Berlin, Heidelberg, 2017. 193-205.

[5] Elleuch, Hanene, et al. "A static hand gesture recognition system for real time mobile device monitoring." Intelligent Systems Design and Applications(ISDA), 2015 15th International Conference IEEE, 2015.

[6] Shin, Jungpil, and Cheol Min Kim. "Non-Touch Character Input System Based on Hand Tapping Gestures Using Kinect Sensor." IEEE Access 5,2017, 10496-10505.

[7] Bobić, Vladislava, Predrag Tadić, and Goran Kvaščev. "Hand gesture recognition using neural network based techniques." Neural Networks and Applications (NEUREL), 2016 13th Symposium on. IEEE, 2016.

[8] Randive, A. A., H. B. Mali, and S. D. Lokhande. "Hand gesture segmentation." International Journal of Computer Technology and Electronics Engineering 2.3 (2012).

[9] Zhang, Runqing, Yue Ming, and Juanjuan Sun. "Hand gesture recognition with SURF-BOF based on gray threshold segmentation." Signal Processing (ICSP), 2016 IEEE 13th International Conference on. IEEE, 2016.

[10] Y. Fang, K. Wang, J. Cheng and H. Lu. "A real-time hand gesture recognition method," in Proc. of IEEE Int. Conf. Multimedia and Expo, Beijing, China, July 2007, pp. 995-998.

[11] N. H. Dardas and D. G. Nicolas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," IEEE Trans. Instrumentation and Measurement, vol. 60, no. 11, pp. 3592-3607, Nov. 2011.

[12] Yang, Jianyu, Chen Zhu, and Junsong Yuan. "Real tie hand gesture recognition via finger-emphasized multi-scale description." Multimedia and Expo (ICME), 2017 IEEE International Conference on. IEEE, 2017.

[13] S. Belongie, J. Malik and J. Puzicha. "Shape matching and object recognition using shape contexts," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pp. 509-522, Apr. 2002.

[14] Li, Yi. "Hand gesture recognition using Kinect." Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on. IEEE, 2012.

[15] Lai, Zhongyuan, et al. "Fingertips detection and hand gesture recognition based on discrete curve evolution with a kinect sensor." Visual Communications and Image Processing (VCIP), IEEE, 2016.

[16] Dulayatrakul, Jakkrit, et al. "Robust implementation of hand gesture recognition for remote human-machine interaction." Information Technology and Electrical Engineering (ICITEE), 2015 7th International Conference on. IEEE, 2015.

[17] Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image based hand gesture recognition techniques." Green Engineering and Technologies (IC-GET), 2016 Online International Conference on. IEEE, 2016.

[18] Cheng, Hong, Lu Yang, and Zicheng Liu. "Survey on 3D hand gesture recognition." IEEE Transactions on Circuits and Systems for Video Technology 26.9 (2016): 1659-1673.

[19] Andersen, Michael Riis, et al. "Kinect depth sensor evaluation for computer vision applications." Technical Report Electronics and Computer Engineering 1.6, 2012.