Robust Hand Gesture Recognition from 3D Data

Vinod K Kurmi
Indian Institute of Technology
Dept. of Electrical Engineering
Kanpur
India (208016) , Kanpur
vinodkk@iitk.ac.in

Garima Jain
Indian Institute of Technology
Dept. of Electrical Engineering
Kanpur
India (208016) , Kanpur
gari1217@gmail.com

KS Venkatesh
Indian Institute of Technology
Dept. of Electrical Engineering
Kanpur
India (208016) , Kanpur
venkats@iitk.ac.in

Abstract

In this paper, we use the output of a 3D sensor (ex. Kinect from Microsoft) to capture depth images of humans making a set of predefined hand gestures in various body poses. Conventional approaches using Kinect data have been constrained by the limitation of the human detector middleware that requires close conformity to a standard near erect, legs apart, hands apart pose for the subject. Our approach also permits clutter and possible motion in the scene background, and to a limited extent, in the foreground as well. We make an important point in this work to emphasize that the recognition performance is considerably improved by a choice of hand gestures that accommodate the sensor's specific limitations. These sensor limitations include low resolution in *x* and *y* as well as *z*. Hand gestures have been chosen(designed) for easy detection by seeking to detect a fingers apart, fingertip constellation with minimum computation. without, however compromising on issues of utility or ergonomy. It is shown that these gestures can be recognised in real time irrespective of visible band illumination levels, background motion, foreground clutter, user body pose, gesturing speeds and user distance. The last is of course limited by the sensor's own range limitations. Our main contributions are the selection and design of gestures suitable for limited range, limited resolution 3D sensors and the novel method of depth slicing used to extract hand features from the background. This obviates the need for preliminary human detection and enables easy detection and highly reliable and fast (30 fps) gesture classification.

Keywords

Hand gesture recognition; Kinect; depth map; histogram; depth slicing; fingertip constellation;

1 INTRODUCTION

Human computer interaction (HCI) is not limited by physical contact with the devices. It has the potential to change the way users interact with computers and appliances, by eliminating input devices such as joysticks, mice, remote control units and keyboards, and allowing the unencumbered body to give signals to the computer through gestures. In gesture based devices, the right choice of gestures is very important for user comfort. Thus, all the gestures should be ensured to be comfortable and simple.

There are different techniques available for hand gesture recognition such as hand modeling, pattern recognition, image processing, etc. Modeling of the hand for gesture recognition has been based on Hidden Markov Model (HMM) in [eickeler1998, elmezain2008, wilson1999, fujii2014]. [suryanarayan2010] presented a 2D and 3D descriptor-based recognition system. For training the hand model, a support vector machine (SVM) was used. A state-based technique for recognition of hand gesture was also used in [bobick1997] by defining the gesture as a sequence of states. Recently, depth based gesture recognition has become quite popular in HCI. Kinect based gesture recognition systems are an example of this. [biswas2011, asad2013, raheja2011] are based on depth data provided by Kinect. Biswas [biswas2011] uses background subtraction and auto thresholding to segment the hand from an image. This allows the system to work only if hand is the first object in front of the camera. A multiclass SVM is required to train the system and individual analysis of each pixel in each frame makes it computation intensive.

Asad [asad2013], Ren [ren2013], Heickal [heickal2013] and Raheja [raheja2011] use OpenNI SDK which is dependent on predefined and available libraries like Nite middleware from Primesense, which can not be used as an open source and besides, works only when the subject is in a standing pose and facing the sensor. We thus observe many limitations in the present day literature for Kinect based hand gesture recognition.

Liu [liu2004] captures depth data using time-of-flight camera and assumes the hand to be the closest to the camera with no objects at the same depth or at a lesser depth. For their algorithm to work, the presence of the face in the frame is compulsory. Penne [penne2008] makes a similar assumption for the user to sit directly in front of the camera (distance 70-100 cm) with hand being the nearest object to the sensor.

Prisacariu [prisacariu2011] uses a combination of visual tracking and an off-the-shelf accelerometer. The tracking requires intricate 3D modeling of the hand and an accelerometer needs to be mounted on the hand, making the process cumbersome. Bigdelou [bigdelou2012] and Jaemin [jaemin2013] use built in Random Forest classifiers provided by Microsoft Kinect SDK, a closed source software, which provides skeleton modelling of human body.

In this paper, we present low complexity algorithms specifically for hand gesture recognition from the data obtained by the 3D sensor. The main goal is to detect hand gestures in varying light conditions, irrespective of neighbouring clutter, both in front of and behind the subject. In the first part, we proceed with hand segmentation with the assistance of depth histograms. Next, the fingertips of the hand are used to form a human hand specific, distinctive constellation of salient points, which yields the position and orientation of the hand and fingertips. The system is able to reject invalid gestures and works even in complete darkness, on account of exclusively depending on the depth information. Separately, we design gestures that are planned for easy detection and ergonomy, and take into consideration the sensor's specific limitations. Gesture recognition involves tracking the change in position, orientation , shape(open/close) of the hand. Gestures have also been designed for two hands used simultaneously, which keep a track of their positions with respect to each other. We have worked with a commercially available depth sensor called Kinect sold by Microsoft. The system works in real time at 30 fps. The software used for the project is Open CV in Microsoft Visual Studio using C and C++, on Windows platform with 4GB RAM. We present a brief summary of present technology and compare with our own in Table 1.

The paper is organized as follows: Section 2 explains our approach for hand segmentation from 3D sensor data and subsequent hand identification. Section 3 describes the algorithms devised for gesture recognition. The methodology adopted for distinguishing single-hand gestures and two-hand gesture is also discussed. Section 4 explains the results and experiments of the proposed algorithm. Section 5 concludes the paper with the discussion on future scope.

2 PROPOSED ALGORITHM

In our work, we have extracted depth data using Kinect sensor. The depth data sent by the Kinect is of 16 bits, with the least significant 3 bits containing index of user detected at pixel and 1 bit for error. The remaining 12 bits carry the depth information. So, the actual depth at any pixel is related to the pixel value as follows:

$$\text{Actual depth in mm} = \text{pixel value} \times \text{depth scale factor} \tag{1}$$

where depth scale factor= $\frac{2^{12}}{255}$.

2.1 Depth slicing

The essence of our approach lies in exploiting the z information of the image. A sample depth image captured from Kinect is shown in Fig. 1. We create a depth (z) histogram of the image and apply a sequence of thresholds that slice the RGBD scene into depth segments. Since we take care to design gestures that maintain a minimum hand-torso distance and a clear hand sensor distance, the hand and torso are represented as peaks in the depth histogram separated by valleys. An appropriate choice of thresholds will isolate the slice containing the hand from the rest of the scene. Fig. 2 shows the histogram and the related depth image, where the green peak in the histogram plot corresponds to the human hand and white peak to the human body. It is quite evident from the figure that the peak corresponding to the body has higher amplitude on account of the torso's much larger size.

There is a valley in the depth histogram between the hand depth peak and the human body depth peak. This valley is best suited for the depth threshold required for segmenting the hand. The first and second peaks are at d' and d'' if they satisfy (2) and (3) respectively. Subsequently, the valley is selected at depth d_v if it satisfies (4).

$$\begin{cases} h(d') > h(d) & 0 < d < d' \\ h(d') \geq h(d' - |k|) \end{cases} \tag{2}$$

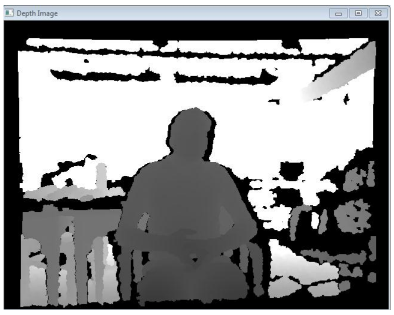


Figure 1: *Depth image*

Method	Camera Used	Foreground objects	Training Required	Real time	Additional Constraints
Biswas [biswas2011]	Kinect	Not allowed	Yes	No	No
Asad [asad2013]	Kinect	Not allowed	No	Yes	NITE Middleware Required
Raheja [raheja2011]	Kinect	Not allowed	No	Yes	NITE Middleware Required
Liu [liu2004]	TOF	Not allowed	No	Yes	Face needed in the frame
Penne [penne2008]	TOF	Not allowed	No	No	No
Ren [ren2013]	Kinect	Not allowed	No	(13.5fps)	Kinect SDK Required
Bigdelou [bigdelou2012]	Kinect	Not allowed	Yes	Yes	Kinect SDK Required
Heickal [heickal2013]	Kinect	Allowed	Yes	Yes	NITE Middleware Required
Jaemin [jaemin2013]	Kinect	Allowed	No	Yes	Kinect SDK Required
Proposed	Kinect	Allowed	No	Yes (30fps)	No

Table 1: Comparison with other reported approaches

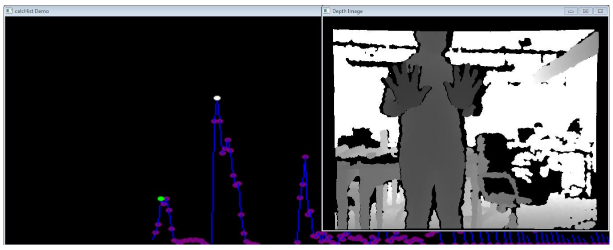


Figure 2: *Histogram and corresponding depth image*

$$\begin{cases} d'' > d' \\ h(d'') > h(d) \qquad d' < d < d'' \\ h(d'') \geq h(d'' - |k|) \end{cases} \tag{3}$$

$$\begin{cases} d'' > d_v \\ d' < d_v \\ h(d_v) < h(d') \\ h(d_v) < h(d'') \\ h(d_v) \leq h(d" - |k|) \end{cases} \tag{4}$$

where $|k|$ <10 cm is any small integer and $h(d)$ is the histogram function at any depth value d.

Next, we segment the hand using any threshold $t \leq v - d$ into the binary hand indicator image $h(x,y)$ using (5).

$$h(x,y) = 1, \text{iff } d - t < z < d + t \tag{5}$$

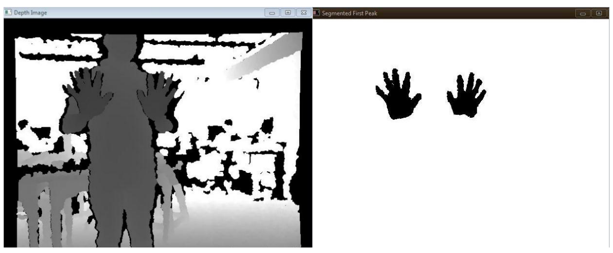


Figure 3: *Hand segmentation*

In Fig. 3, the segmented region of hand is shown in black and the background pixels are shown in white. This segmentation output is a result of depth thresholding to isolate the first peak.

In Fig. 6, both hands present the gesture at two different depths, and the above method is extended to isolate each hand through a sequence of depth slices. Also, if some object precedes the slices containing the hands, then the foremost slice is rejected and detection is attempted in the subsequent slices.

2.2 Fingertip constellation and hand orientation

Fingertip constellation of the hand is the basis for identification of hand, since, in the open hand, fingertips are always arranged in their unique configuration in space. There may be some other objects which slightly resemble this configuration, but the human hand yet has many distinctive properties that distinguish it from other objects.

In the segmented hand binary image, the contour is detected, and is enclosed in its convex hull. High curvature points on the hull represent fingertips. Fingertip positions as well as the maximum convexity defect points between consecutive fingertips are both noted as feature points. They together constitute the constellation of points representing the hand.

2.2.1 Geometrical Properties of Hand

There is a possibility of objects being present at the same depth as the hand, and these objects would not be eliminated by depth slicing alone. These are filtered out, first, with a size filter (hand area is expected to be 1000 to 8000 pixels). A second shape filter examines the feature point constellation of the object.

Suppose that array **P** contains convex hull points

$$\mathbf{P} = [p_1, p_2, ...]$$

and corresponding convexity defects points are stored in the array **Q**

$$\mathbf{Q} = [q_1, q_2, ...]$$

The points p_i and q_i form a vector $\bar{r}_i$, and p_{i+1} and q_i form a vector $\bar{s}_i$. We focus on the angle between vector $\bar{r}_i$ and vector $\bar{s}_i$.

$$\cos(\theta) = \frac{\bar{r}_i.\bar{s}_i}{\|\bar{r}_i\|.|\bar{s}_i\|} \qquad (6)$$

This angle is always less than 60° (only between thumb and index finger it goes to 90°), for the real fingertip. This criterion removes all the false fingertip points.

2.2.2 *Orientation of Hand*

To detect hand orientation, we determine s_m, the centre of the palm as the centre of the largest circle inscribable within the palm and s_c, the centre of the convex hull of the hand. With our choice of gestures, we ensure that the center point of convex hull s_c (red dot inside the maximally inscribed circle in Fig. 8) is always different from the palm center point s_m (blue dot inside the palm in Fig. 8), point s_c always lies towards the fingers as compared to the point s_m. The direction vector $\vec{p}$ from s_m to s_c measures the hand orientation.

2.2.3 *Left and Right Hand*

For hand disambiguation, one uses the differences between the two hand constellations in relation to the current hand orientation. When the orientation is vertical, left and right hands can be disambiguated by the difference in the thumb position. When the hand is horizontal, then we look at the location of the hand constellation with respect to s_c to disambiguate the hands.

2.3 Two hand gestures

In the two-hand case, the first peak of the depth histogram corresponds to both the hands if both hands are at the same depth. After segmenting the hand with the help of depth data, we analyse all the contours in the depth slice. If two contours satisfy the hand identification criteria, we conclude that two hands are present. But, if both the hands are at different depths, they occur in different depth slices. We avoid this condition for two hand gesture case and treat this as a single hand gesture. The hand nearer to the sensor is considered for gesture analysis.

3 PROPOSED GESTURE IDENTIFICATION ALGORITHM

In the previous section, we have explained the isolation of hand from a scene. This section deals with the gesture identification algorithms. The first step is to decipher if the gesture is single or two handed. Then the hand position is tracked from the reference, and the system identifies the gesture performed by the user, if valid. If an invalid gesture is performed, the system rejects it.

3.1 Detection of the number of hands

Fig. 4 outlines the steps used to detect if the gesture is performed by one hand or both the hands. This algorithm also eliminates the foreground objects.

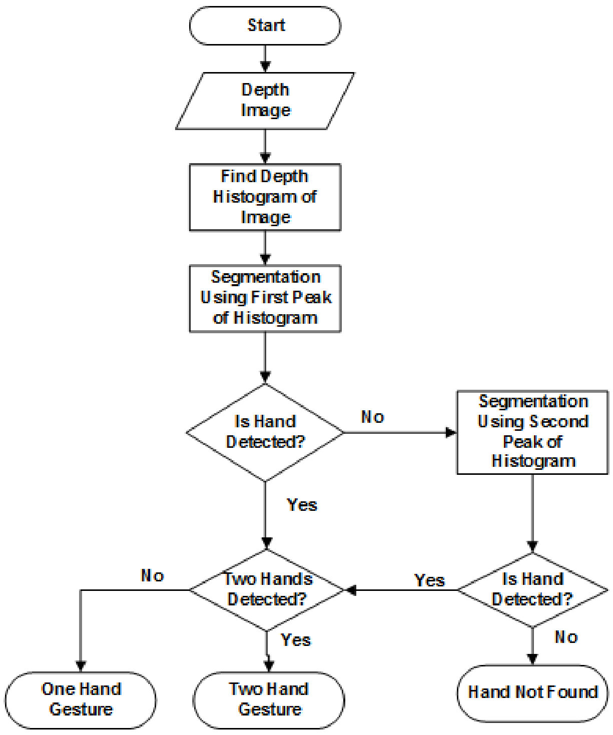


Figure 4: *Flowchart for the detection of number of hands*

3.2 Identification of the gesture

In a gesture recognition system, only valid gestures, i.e, gestures predefined in our library, should be identified and invalid gestures should be ignored. In our algorithm, before performing a gesture, the user has to set a reference point for that gesture. This allows for device initialization, the duration of which maybe set as an operational parameter. The hand center point, if maintained steady for the chosen initialization time, is accepted as the reference point for the rest of the gesture. All the gestures described by us are defined by the relative position of the hand reference point with respect to the palm center point s_m.

All gestures also end with a predefined closing pose called the “gesture over" state after which the reference point is cleared. A reference point has to be created again for performing the next gesture. The presence of the initialisation step ensures that inadvertent hand movements are not considered for processing.

Different gestures for single hand and two hands are described below with the algorithms listed for their identification. This gesture vocabulary is scalable in terms of the number of gestures, using various combinations of both the hands in different orientations.

3.2.1 *Single hand gestures*

There are six gestures designed for single hand implementation and are listed in the Table 2.

The algorithm to identify a single hand gesture, i.e., Algorithm 1, starts when it has been identified that a single hand is going to perform the gestures.

Algorithm 1 Identification of single hand gestures

```
1: start:
2: sm ← center of palm
3: i ← threshold for reference point creation
4: j ← threshold for change in area
5: k ← threshold for decrease in depth
6: l ← threshold for rate of decrease in depth
7: first:
8: if reference point exists then
9:     goto top2
10: else
11:     goto top1
12: top1:
13: if sm constant for time > i then
14:     create reference point
15: else
16:     goto start
17: top2:
18: if change in area > j then
19:     if count==1 then
20:         gesture ← mute
21:     else
22:         gesture ← unmute
23:     count ← !count
24:     goto last
25: top3:
26: if decrease in depth > k then
27:     if rate of decrease in depth > l then
28:         gesture ← OFF
29:     else
30:         gesture ← Invalid Gesture
31:     goto last
32: top4:
33: if hand is moving then
34:     if hand orientation==left then
35:         goto top5
36:     else
37:         goto top6
38: else
39:     goto last
40: top5:
41: if hand is moving up then
42:     gesture ← Channel Up
43: else
44:     gesture ← Channel Down
45: goto last
46: top6:
47: if hand is moving up then
48:     gesture ← Volume Up
49: else
50:     gesture ← Volume Down
51: goto last
52: last:
53: Gesture over
54: goto start
```

S.N.	Hand Gesture	Meaning
1	Horizontal right hand moved upward	Channel Up
2	Horizontal right hand moved downward	Channel Down
3	Horizontal left hand moved upward	Volume Up
4	Horizontal left hand moved downward	Volume Down
5	Vertical hand (L/R) closed	Mute/Unmute
6	Vertical hand (L/R) moved towards the sensor	OFF

Table 2: Single hand gestures

A few of the gestures are specific to whether the gesture is performed by left or right hand. But some gestures are independent to hand identification, like Mute/Unmute and OFF. These gestures can be done with either of the hands.

3.2.2 Two hand gestures

We have defined ten two-hand gestures for this work, with four being static and six being dynamic gestures, as in Fig. 5. In the dynamic gestures, at least one of the hands is in motion. It should however be kept in mind that the two hands do not overlap while performing the gesture. They should be kept at a sufficient distance from each other. These gestures are using Algorithm 2.

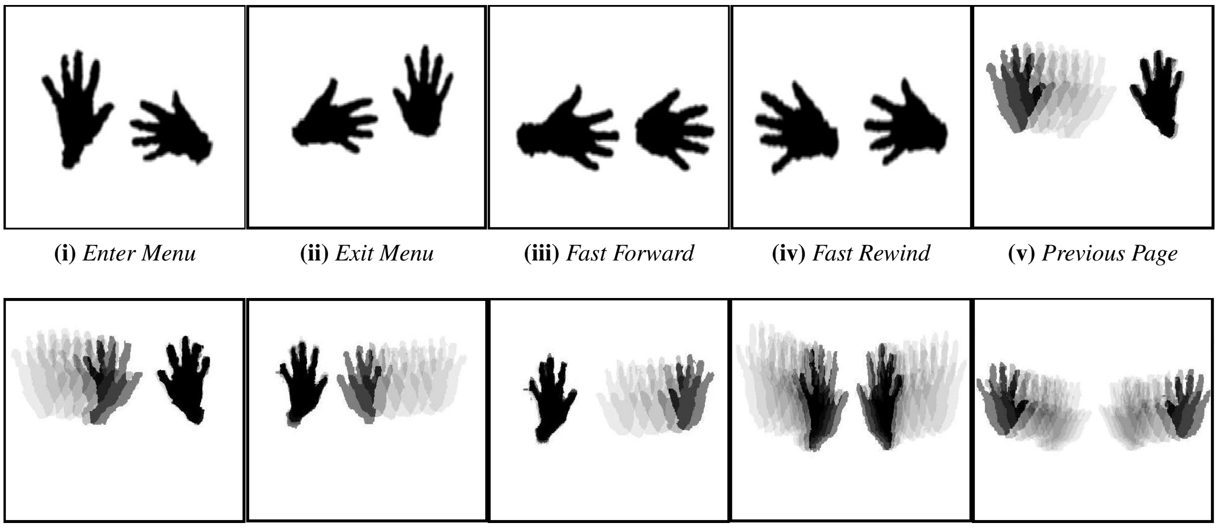
4 RESULTS AND EXPERIMENTS

Initially, the hand is segmented from the scene with the aid of depth histogram. The contour detection of the segmented depth slice is shown in Fig. 7. Fig. 8 shows the convex hull points and convexity defect points of the contour. The purple dots are the convexity defect maxima and red dots are the fingertips. A maximal inscribed circle is drawn inside the hand.

Depth based hand segmentation depends upon the peak value of depth histogram. It is possible that this peak does not correspond to hand or there might be other objects present at the same depth as hand. The fingertip constellation detection used by us can deal with these situations comfortably. Some of the cases are presented here.

4.1 Spurious object present at same depth

The objects which are present at the same depth level of the hand, do not get eliminated after depth slicing. The convex hull points and convexity defects points configuration for them cannot be the same as of the hand, and thus allows us to eliminate such objects.



(i) *Enter Menu* **(ii)** *Exit Menu* **(iii)** *Fast Forward* **(iv)** *Fast Rewind* **(v)** *Previous Page*

(vi) *Next Page* **(vii)** *Scroll Left* **(viii)** *Scroll Right* **(ix)** *Zoom In* **(x)** *Zoom out*

Figure 5: *Two Hand Gestures*

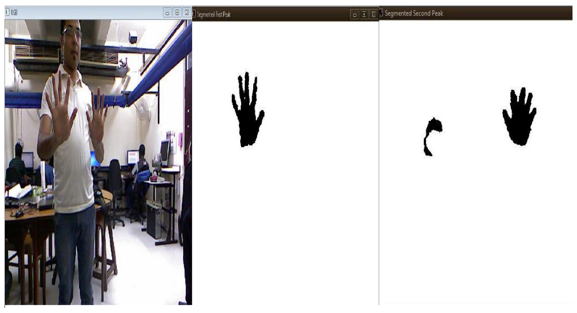


Figure 6: *Segmentation using first and second peak*

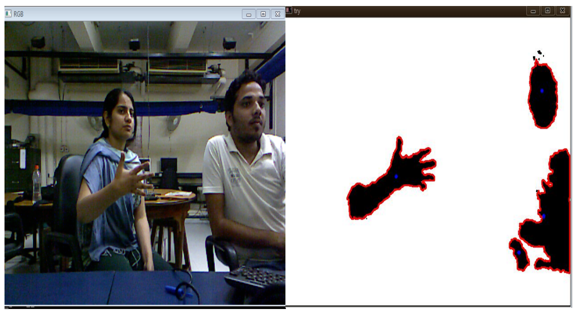


Figure 7: *Contour of the hand*

4.2 Spurious objects in foreground

In the gesture recognition environment, it is also possible that the first peak of histogram does not satisfy the properties of a human hand. This means the hand is not the nearest object from the depth sensor. In this case, we have to look at the next depth slice. This process ideally can be continued until the human hand is found. But there are also limitations of range as well as view angle of the camera. The foreground objects should not be too large, otherwise they will cover most of the image.

Fig. 9 shows the segmentation using the second peak of the depth histogram. In this figure, we have segmented the second object from the sensor or in other words it is the second depth slice. Based on the fingertip constellation, the human hand is identified and remaining objects are ignored. The figure shows that only the hand is detected, even in the presence of other objects in the foreground or at the same depth.

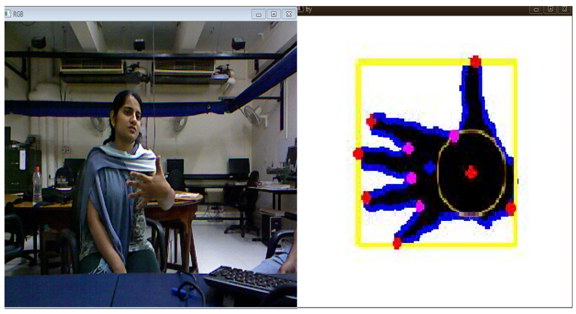


Figure 8: *Convex hull points and convexity defects points with palm center*

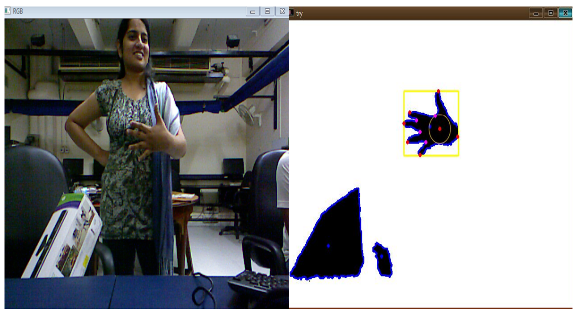


Figure 9: *Identification of the hand in the presence of other objects*

4.3 Recognition Performance Analysis

The performance of the recognition system is only slightly affected to an extent by careless gesturing. This

Algorithm 2 Identification of two hand gestures

```
start:
sm ← center of palm
i ← threshold for reference point creation
first:
if reference point exists then goto top2
else goto top1
top1:
if sm constant for time > i then
    create reference point
else goto start
top2:
if both hands vertical then goto top3
else
    if both hands horizontal then goto top7
    else goto top8
top3:
if both hands moving then
    goto top4
else
    if only left hand moving then
        goto top5
    else goto top6
top4:
if both hands moving outward then
    gesture ← Zoom Out
else
    gesture ← Zoom In
goto last
top5:
if hand moving outward then
    gesture ← Next Page
else
    gesture ← Previous Page
goto last
top6:
if hand moving outward then
    gesture ← Scroll Right
else
    gesture ← Scroll Left
goto last
top7:
if both hands oriented towards left then
    gesture ← Fast Forward
else
    gesture ← Fast Rewind
goto last
top8:
if left hand horizontal then
    gesture ← Enter Menu
else
    gesture ← Exit Menu
goto last
last:
Gesture over
goto start
```

is not to deny that the recognizer has considerable, and tunable recognition acceptance tolerance incorporated in its design. Yet, a gesture performed in an extremely wrong way, will result in lesser accuracy as compared to our sample set results. All the gestures work fine in real time scenarios. Our gesture vocabulary is intuitive and comfortable. Sensitivity to body pose variation and background and foreground clutter is minimal. Experiments to evaluate the performance of the gesture recognition system have been conducted, with each gesture performed 3 times by 15 individuals. The demonstration videos clearly reflect the robustness and accuracy of the system, irrespective of neighbouring clutter, user body pose, and its completely invariance to visible band illumination in indoor conditions.

Gesture	Correct Recognition	Unsuccessful Recognition
Channel up	45/45	0/45
Channel down	45/45	0/45
Volume up	45/45	0/45
Volume down	45/45	0/45
Mute/Unmute	45/45	0/45
OFF	45/45	0/45

Table 3: Single hand gestures

Gesture	Correct Recognition	Unsuccessful Recognition
Enter Menu	45/45	0/45
Exit Menu	45/45	0/45
Fast Forward	44/45	1/45
Fast Rewind	44/45	1/45

Table 4: Two hand static gestures

Gesture	Correct Recognition	Unsuccessful Recognition
Zoom In	45/45	0/45
Zoom Out	45/45	0/45
Next Page	45/45	0/45
Previous Page	45/45	0/45
Scroll Left	45/45	0/45
Scroll Right	45/45	0/45

Table 5: Two hand dynamic gestures

5 CONCLUSIONS

The ability of our algorithm to work without any intricate modelling of the hand or the use of heavy machine learning techniques, enables it to work straight away without any sort of user dependent training whatsoever. We develop a novel approach for segmenting the hand, based on the constellation of fingertip points in space, while ensuring that the design of gestures (all five fingertips of the hand are visible to the sensor with a minimal hand-torso distance) accommodates the limitations of the sensor.

This approach is robust and accurate irrespective of neighbouring clutter, varying lighting conditions, user body pose, and is completely invariant to visible band illumination in indoor conditions. To our knowledge, no currently available system possesses all these features.

The system working range is from 0.6 m to 2.5 m. The working range of the system can be improved when sensors that operate over a wider range are available. The prototype can be further extended for multi-user control, finger gesture recognition and to distinguish between palm and the back of the hand using a higher resolution depth camera.

6 ACKNOWLEDGMENTS

This work was supported by Samsung India Electronics Private Limited as a part of a research project.

7 REFERENCES

[asad2013] Asad, M.; Abhayaratne, C., "Kinect depth stream pre-processing for hand gesture recognition," Image Processing (ICIP), 2013 20th IEEE International Conference on , vol., no., pp.3735,3739, 15-18 Sept. 2013. doi: 10.1109/ICIP.2013.6738770.

[bigdelou2012] Bigdelou, Ali, Tobias Benz, Loren Schwarz, and Nassir Navab. "Simultaneous categorical and spatio-temporal 3d gestures using kinect." In 3D User Interfaces (3DUI), 2012 IEEE Symposium on, pp. 53-60. IEEE, 2012.

[biswas2011] Biswas, K. K.; Basu, S.K., "Gesture recognition using Microsoft KinectÂ®," Automation, Robotics and Applications (ICARA), 2011 5th International Conference on , vol., no., pp.100,103, 6-8 Dec. 2011 doi: 10.1109/ICARA.2011.6144864.

[bobick1997] Bobick, Aaron F and Wilson, Andrew D, "A state-based approach to the representation and recognition of gesture, " Pattern Analysis and Machine Intelligence, IEEE Transactions no 12, 1997 vol. no. 19, pp. 1325-1337.

[eickeler1998] Eickeler, S.; Kosmala, A; Rigoll, G., "Hidden Markov model based continuous online gesture recognition," Pattern Recognition, 1998. Proceedings. Fourteenth International Conf. on , pp.1206,1208 vol.2, 16-20 Aug 1998.

[elmezain2008] Elmezain, M.; Al-Hamadi, A; Appenrodt, J.; Michaelis, B., "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," Pattern Recognition, 2008. ICPR 2008. 19th International Conference on , vol., no., pp.1,4, 8-11 Dec. 2008.

[fujii2014] Fujii, Tatsuya, Jae Hoon Lee, and Shingo Okamoto. "Gesture Recognition System for Human-Robot Interaction and Its Application to Robotic Service Task." In Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists. 2014.

[heickal2013] Heickal, Hasnain, Tao Zhang, and Md Hasanuzzaman. "Real-time 3D full body motion gesture recognition." In Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on, pp. 798-803. IEEE, 2013.

[jaemin2013] Jaemin, Lee, H. Takimoto, H. Yamauchi, A. Kanazawa, and Y. Mitsukura. "A robust gesture recognition based on depth data." In Frontiers of Computer Vision, 2013 19th Korea-Japan Joint Workshop on, pp. 127-132. IEEE, 2013.

[penne2008] Penne, Jochen, Stefan Soutschek, Lukas Fedorowicz, and Joachim Hornegger. "Robust real-time 3D time-of-flight based gesture navigation." In Automatic Face and Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on, pp. 1-2. IEEE, 2008.

[raheja2011] Raheja, J.L.; Chaudhary, A; Singal, K., "Tracking of Fingertips and Centers of Palm Using KINECT," Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on , vol., no., pp.248, 252, 20-22 Sept. 2011.

[ren2013] Ren, Zhou, J. Yuan, J. Meng, and Z. Zhang. "Robust part-based hand gesture recognition using kinect sensor." Multimedia, IEEE Transactions on 15, no. 5 (2013): 1110-1120.

[suryanarayan2010] Suryanarayan, P.; Subramanian, A; Mandalapu, D., "Dynamic Hand Pose Recognition Using Depth Data," Pattern Recognition (ICPR), 2010 20th International Conference on , vol., no., pp.3105,3108, 23-26 Aug. 2010.

[wilson1999] Wilson, Andrew D and Bobick, Aaron F, "Parametric hidden markov models for gesture recognition," Pattern Analysis and Machine Intelligence, IEEE Transactions 1999, vol.no 21, pp.884-900.

[liu2004] Liu, Xia, and Kikuo Fujimura. "Hand gesture recognition using depth data." In Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, pp. 529-534. IEEE, 2004.

[prisacariu2011] Prisacariu, Victor Adrian, and Ian Reid. "Robust 3D hand tracking for human computer interaction." In Automatic Face and Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp. 368-375. IEEE, 2011.