# Unsupervised Learning Based Static Hand Gesture Recognition from RGB-D Sensor

Bindu Verma and Ayesha Choudhary(✉)

School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi, India
binduverma67@gmail.com, ayeshac@mail.jnu.ac.in

**Abstract.** In this paper, we propose a novel, real-time static hand gesture recognition framework based on unsupervised learning. We use unlabelled training data from an RGB-D (RGB, Depth) sensor and use the depth data to detect the hand(s) in the image. We believe that the gesture is dependent on the shape of the hand, which is defined by number of extended/open fingers and which fingers are extended/open. We use unsupervised learning techniques to detect and label the extended fingers in the training data. We also use the same algorithm to detect and label the extended fingers in the test images on-the-fly. Our gesture recognition system is independent of the orientation of the gesturing hand and arm. Moreover, it does not require any markers and is completely unobtrusive, making it suitable for assistive living paradigms.

**Keywords:** Hand gesture recognition (HGR) · Depth camera based recognition · Unsupervised learning

## 1 Introduction

In this paper, we propose a novel hand gesture recognition algorithm based on unlabelled data from RGB-Depth sensors and unsupervised learning. Our framework uses depth data for detecting and extracting the gesturing hand. Our gesture recognition technique is independent of the orientation of the hand, and does not require the gesturing person to wear any markers or wrist bands. We believe that the hand shape describing the gesture is purely dependent on the number of extended fingers and which fingers are extended. Therefore, the training images have to be processed to detect the hand, then the number of extended fingers and then find which fingers are extended. During the test phase, as a user gestures in front of an RGBD sensor, the same steps are followed to detect and label the fingers correctly. Cascaded recognition is then performed for fast and accurate recognition of the gesture in real-time. Our hand gesture recognition framework is specially suitable for assistive living paradigms since it assumes minimal hand movement, basically only extending and retracting fingers and the gesture can be made while the arm of the gesturing hand is in

A. Abraham et al. (eds.), *Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016)*, Advances in Intelligent Systems and Computing 614, DOI 10.1007/978-3-319-60618-7_30

any orientation. A person needs to learn only a set of gestures for effective non-verbal communication. Hand gesture recognition based on an RGBD sensor is more suitable than a wearable sensor because it is completely unobtrusive and therefore, does not disturb the user in any way.

Hand gestures are a natural non-verbal form of human-human and human-machine communication. However, there are various forms of hand-gestures, such as gestures made when two or more people talk, dynamic gestures that require movement of the arms and hands and static gestures where only the shape of hands and fingers are enough to communicate the requirement. We focus on the static gestures, requiring only movement of fingers thus making our gesture recognition system suitable for people who have special needs and cannot verbally communicate or make large hand movements. Our system uses a depth sensor and does not require any wearable sensor. However, it does require the depth sensor to be placed optimally so that the gesturing hand is closest to the RGBD sensor. In the next section, we discuss the related work in the area of hand gesture recognition. For evaluating our framework, we use the data set from [1], where the hand is nearest to the Mirosoft Kinect sensor. We consider all the gestures in that data set to evaluate our framework.

## 2 Related Work

Hand gestures can be classified as static or dynamic. Various techniques are used for detecting the hands and finding the shape and trajectory of the hand movements in case of dynamic hand gestures. Hand gesture recognition is also carried out by capturing the gesture data using various sensors such as data glove, depth sensors and cameras. In this section, we focus on the static hand gesture recognition methods based on the data captured using RGB or RGBD cameras such as the Microsoft Kinect. Authors in [1] use depth data from a Kinect sensor to detect the user's hand robustly and then represent it as a time series curve, which captures the topological properties of the hands such as finger parts. They then apply template matching using Finger-Earth Mover's distance to recognize the hand gesture. However, this method requires the user to wear a black belt on the gesturing hand's wrist to accurately segment the hand shape. We use their data set for evaluating our automated hand recognition framework, however, we do not use the black belt as a delimiter to segment the hand region.

In [2], Kinect data along with the skeletal information is used for detecting the hand. To classify and recognize the hand gestures, super-pixel earth mover's distance is used. Authors in [3], use the depth data to divide the hand portion into two parts, the palm and the fingers. They then use a multi-class support vector machine classifier to recognize the hand gesture. Very recently, authors in [4] proposed $3D$ hand gesture recognition based on finger and hand pose estimation techniques where, they modeled the finger as a cylindrical object using the parallel edge feature. Using finger, wrist position and palm center they extract geometrical features of the hand shape. High level feature and finger localization are extracted using weighted radial projection. Gesture recognition is carried out using Hidden Markov Model (HMM) and Dynamic Time Warping (DTW) algorithms. Similarly, [5] use Dynamic time warping (DTW) to recognize static hand gesture and motion trajectories while authors in [6] uses deformable hand shape basis model to fit on their depth data. They use the concept of shape fitting energy to infer the hand shape from deformed depth data.

In [7] a saliency based feature construction algorithm is proposed and histogram intersection kernel function is used to map the original feature space to kernel feature space. Then, sparse representation classification is carried out in the kernel feature space. Authors in [14] develop a Kinect based hand gesture recognition technique specially to recognize the hand shapes useful for arithmetic operations and for playing the Rock-Paper-Scissors game. Authors in [8] use hand gesture recognition for the 10 American sign language digits by creating a realistic $3D$ hand model representing 21 different parts of the hand and using support vector machine for classification.

Authors in [9] have proposed an artificial neural network for static hand gesture recognition while authors in [10] have proposed a framework for music and sound control through hand gesture recognition, using time delay neural networks (TDNN). Authors in [11] have proposed self growing and self organizing neural gas (SGONG) network where the hand region is segmented using the color of the skin. Recently, authors in [12], proposed a RGBD descriptor based hand gesture recognition system with application for human-machine interface application in a car. A detailed review of hand gesture recognition methods is also given in [12]. In [13] uses histogram of 3D facets (H3DF) to define the 3D shape of the hand with the help of depth map. H3DF works as a characteristics descriptor to classify the hand gesture with Support Vector Machine (SVM). In our proposed work, we use incremental clustering to find the hand shape and the number of extended fingers in the static hand gesture. Hand gesture recognition is then based on the cascaded recognition, where first we consider both the number of extended fingers as well as which fingers are extended.

## 3 Hand Detection Using Depth Data

We assume that the RGBD sensor is placed such that the gesturing hand(s) is closest to the depth sensor compared to the rest of the body of the user. We assume that the gesturing hand and arm is neither too close nor too far from the depth sensor, so that the gesturing hand is completely visible in the hand. Therefore, the hand(s) is detected as the foreground object present at a fixed depth range. The pixels corresponding to the depth range $\theta_1$ to $\theta_2$, are segmented as the hand region using Eq. 1 where $\theta_1$ and $\theta_2$ are the thresholds on the depth for hand region segmentation.

$$f(i,j) = \begin{cases} 1, & \text{if}\, \theta_1 \le d(i,j) \le \theta_2 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where, d(i, j) is depth value at pixel coordinate $(i, j)$ and $f(i, j)$ represents the pixel value of the newly formed binary image. Thus, we get a binary image where white represents the pixels corresponding to the hand region and black represents the background. In Fig. 1, we show the RGB images, the corresponding depth images and the segmented hand region for some of the gestures in the dataset. Then, we find the connected components in the foreground image. If there is more than one component, we form separate foreground images of each of the connected components. Then, each of these foreground objects is treated as a separate image with one gesturing hand.

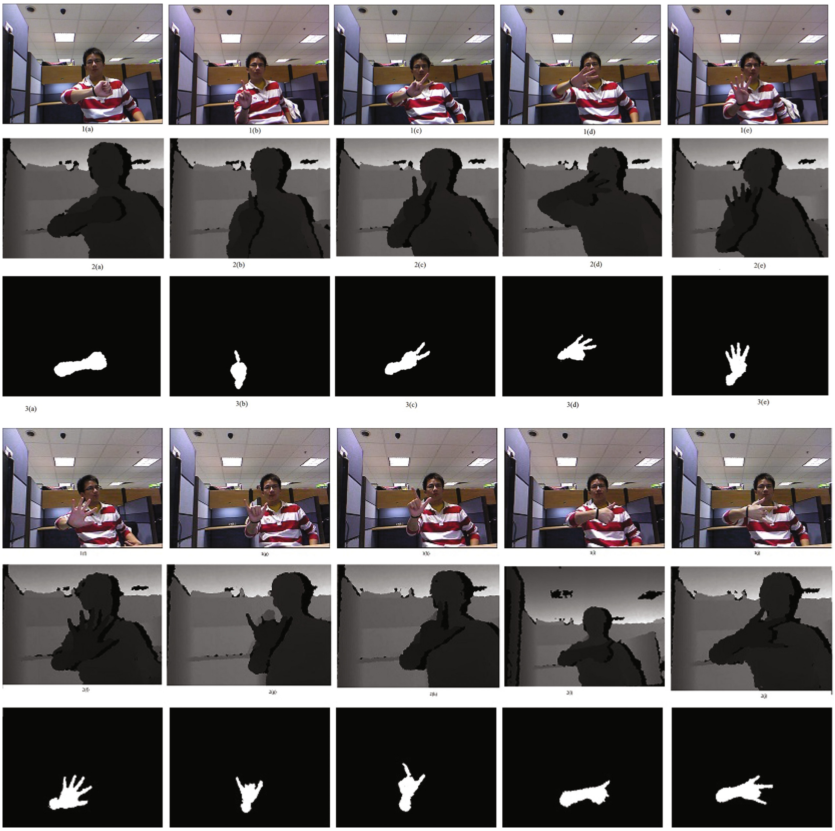

**Fig. 1.** The images 1(a) to 1(j) show the RGB images, 2(a) to 2(j) show the corresponding depth images and 3(a) to 3(j) show the detected hand region using the depth data.

## 4 Orientation Invariance

Our hand gesture recognition system is designed to provide flexibility to the user to gesture in any orientation. To enable this feature, we first find the angle of orientation of the detected region using principal component analysis (PCA)

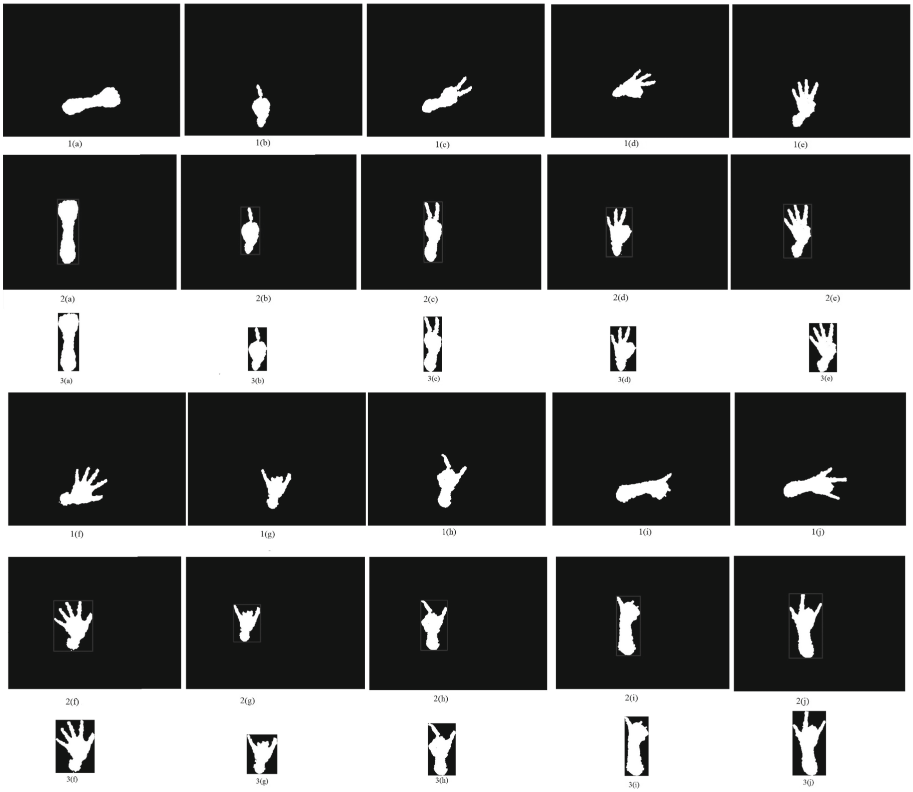

**Fig. 2.** Images 1(a) to 1(j) show the detected hand region; 2(a) to 2(j) show the corresponding rotated images and; 3(a) to 3(j) show the cropped images of the hand region.

and find the angle of the major axis. Then, we rotate the segmented gesture by this angle with respect to the vertical axis. The foreground detected region is cropped to reduce the size of the image and make the system efficient as shown in Fig. 2.

## 5 Detecting Shape of the Hand

Our focus is on the shape of the gesturing hand, which is dependent on the number of extended fingers and the fingers that are extended. We perform pixel level incremental clustering to discover the extended fingers in each gesture. We start with the top left corner of the image and the first foreground pixel becomes the element of the first cluster. Then, for row $j$, for all foreground pixels, $f_ij$ are put in a cluster if and only if $f_{i-1,j}$ is also in that cluster. If $f_{i,j}$ is a background pixel, then we keep incrementing to the next pixel in that row till we reach a foreground pixel. This foreground pixel is the first element of a new cluster. Therefore, for each row we get one or more clusters of foreground pixels. As we move across rows, we cluster within the row as well as across rows. This leads to each finger getting segmented as a separate cluster. The foreground pixels get partitioned into various clusters, and it is not necessary that the same gesture has the same partitioning. However, the fingers are correctly detected, as required by our algorithm. Once the fingers are detected, we detect the forearm and crop it. The forearm gets segmented as the region opposite to the fingers and consisting of one or more clusters that are much larger in size compared to the finger clusters. We discard these clusters since they do not provide any useful information in our hand gesture recognition algorithm. For visualization, we depict each cluster in a separate color. Figure 3 shows results of our clustering algorithm. As can be seen the hand gets partitioned into various clusters, however, each extended finger is represented by a separate cluster. For further processing, we randomly label the clusters $C_i$ for $i = 1, 2, \ldots, N$ where $N$ is the total number of clusters in an image and is different for each image. In the next section, we discuss the various features we use to identify the clusters representing the extended (open) fingers and the features that we extract to be able to uniformly label the clusters corresponding to each finger in all the gesture images.

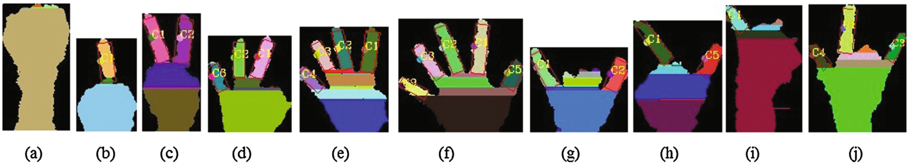

**Fig. 3.** The images (a) to (j) show the various clusters formed in each image using our proposed clustering algorithm. As can be noticed, each finger is a separate cluster.

## 6 Finger Identification and Labelling

A large number of dependencies among the finger joints and the finger and palm joint exist that constrains the search space for labelling fingers in any hand pose. Moreover, the joints of the fingers with the palm constrain the maximum distance and angle between any two fingers. We use these constraints to accurately label the fingers in each gesture. In our labelling, $F1$ represents the thumb, $F2$ represents the index finger, $F3$ represents the middle finger, $F4$ represents the ring finger and $F5$ represents the little finger.

To correctly extract the features, we first identify clusters that represent fingers based on the thickness and length of the clusters. We then collect information such as number of extended fingers, Euclidean distance between these clusters and their orientation. We assume that the clusters that represent extended fingers have a certain ratio of length to thickness (breadth), more specifically the length is much larger than the breadth. We measure the thickness as the average of the length of segments of white pixels in each row of the image belonging to that cluster and the length as the number of rows that belong to the cluster.

We then count the number of such clusters to find the number of open fingers in the gesturing hand. Once the number of open fingers is determined, we find the Euclidean distance between all the extended fingers. The first gesture we consider corresponds to the abduction of fingers and thumb, that is, the spread open hand. We use this data set as the base set to calculate the distance between all the pairs of finger clusters. We label the mid-point of the bounding box of each cluster as the cluster representative and use it for calculating the distance between two clusters representing the fingers. We use PCA on each cluster representing an open finger to find its orientation with respect to the x-axis. Since the images contain only the hand image, the finger clusters closest to the boundary of the image are the little finger and the thumb in case of a gesture with all fingers spread open. We use the premise that the distance between the clusters representing adjacent fingers will be less than the distance between the index finger and the thumb when all the fingers are spread open. Using this premise, we are able to identify the thumb from the two boundary clusters and label the other boundary cluster as the little finger. This enables us to identify the index, middle and ring fingers also. Figure 4 shows the results of detected fingers with their labels.

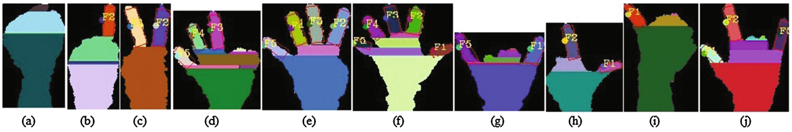

**Fig. 4.** The clusters representing the fingers are consistently labelled to represent fingers across all the gestures in the training set.

## 7 Hand Gesture Recognition

Let $G_1, G_2, \ldots, G_N$ be the $N$ gestures in the training set and $G_T$ be the gesture in the test gesture image. For recognizing a test gesture image, $G_T$, the steps of hand detection and segmentation, clustering and finger labelling is performed as explained in Sects. 3, 4, 5 and 6. These are the same steps that were applied to the training data. We represent each gesture as given in Eq. 2.

$$G_i = \{n_i, \{F_j^i = 1 \text{ or } 0\}_{j=1}^{5}\} \tag{2}$$

where, $n_i$ is the number of extended fingers and $F_j^i$ is 1 if the finger is extended and 0 if the finger is not present in the gesture. We perform a cascaded recognition to reduce the search space and search time. Let $G = \{G_1, G_2, \ldots, G_N\}$ be the set of all gestures in the training set. Initially, all gestures in $G$ are taken into consideration and we check whether the number of extended fingers in $G_T$ are same as in $\{G_i\} \in G \forall_i = 1, 2, \ldots, N$ using Eq. 3.

$$d(G_i, G_T) = \begin{cases} 1 & \text{if } n_i = n_T \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In the next phase, we consider the gesture as a subset $G' \in G$ given in Eq. 4 such that

$$G' = \{G_j | d(G_j, G_T) = 1, 1 \leq j \leq N\} \tag{4}$$

and check whether the finger labels of $G_T$ and $G_j \in G'$ are same. We compute the similarity score as given in Eq. 5 to find if the same fingers are open in the test and training gesture.

$$s(G_j, G_T) = \begin{cases} \frac{1}{n_T} \sum_{k=1}^{n} 1, & \text{if } F_k^j = F_k^T \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where, $n_T$ is the number of open fingers in $G_T$. We assign $G_T$ to the gesture $G_i \in G$ iff $s(G_i, G_T) = 1$.

## 8 Experimental Result

We use the NTU dataset collected in [1] for gesture recognition from RGBD data as well as we captured our own dataset from a Kinect sensor. The NTU dataset is a challenging dataset, with cluttered background involving various subjects gesturing with one hand. There is large variation in orientation and articulation of each gesture by different people. It contains 10 different gestures collected by 10 different subjects. Each subject performs the same gesture 10 times. Therefore, the dataset consists of 1000 images. For each gesture, we use 80 images as training data and rest 20 images as test data. We run the experiment many times, with various sets of 80 images as training data and 20 as test data and report the average recognition rate in 100 runs. Table 1 shows the confusion matrix of each hand gesture. Here in gesture $G_3$ confuses with gesture $G_7$ similarly gesture $G_6$ confuses with gesture $G_5$. We find that errors at the level of detection and segmentation propagate to errors in clustering and labelling of fingers. This leads to cases where the recognition is incorrect. Overall, our recognition rate is 98% and the average running time for hand gesture recognition of one test image is 0.03 s. As explained in Sect. 7, our recognition system first checks if the number of open fingers in the test image and training template are same. If the number of open fingers are not same then matching process returns that both gestures are not same. If number of open fingers are same then matching process further matches the labels on the fingers. If the labels are same then similarity of the gestures computed by Eq. 5 is 1 and our system returns a match and correctly classifies the gesture; otherwise it returns that the gestures are not same. We find that the recognition is carried out accurately using our method. However, errors at the level of detection and segmentation propagate to errors in clustering and labelling of fingers. This leads to cases where the recognition is incorrect. Diagonal entries in table represents the correctly recognized gesture out of 20 hand gesture. We also capture our own data set by a Kinect sensor. Since we do not use RGB image in our proposed work so we capture only depth data. Our captured data are same as NTU hand digit dataset. For each gesture we captured 20 similar gestures and 10 different hand gestures.

**Table 1.** Confusion matrix of our hand gesture recognition system on NTU data set.

| | | Training Gesture | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **CRR** | $(G_1)$ | $(G_2)$ | $(G_3)$ | $(G_4)$ | $(G_5)$ | $(G_6)$ | $(G_7)$ | $(G_8)$ | $(G_9)$ | $(G_{10})$ |
| **Test Gesture** | $(G_1)$ | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $(G_2)$ | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $(G_3)$ | 0 | 0 | 19 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | $(G_4)$ | 0 | 0 | 0 | 19 | 0 | 0 | 0 | 0 | 0 | 1 |
| | $(G_5)$ | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 |
| | $(G_6)$ | 0 | 0 | 0 | 0 | 1 | 19 | 0 | 0 | 0 | 0 |
| | $(G_7)$ | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 |
| | $(G_8)$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 |
| | $(G_9)$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 0 |
| | $(G_{10})$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 |

**Table 2.** Confusion matrix our hand gesture recognition system on own data set validated with the training data of NTU dataset.

| | | Training Gesture | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **CRR** | $(G_1)$ | $(G_2)$ | $(G_3)$ | $(G_4)$ | $(G_5)$ | $(G_6)$ | $(G_7)$ | $(G_8)$ | $(G_9)$ | $(G_{10})$ |
| **Test Gesture** | $(G_1)$ | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $(G_2)$ | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $(G_3)$ | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $(G_4)$ | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $(G_5)$ | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 |
| | $(G_6)$ | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 |
| | $(G_7)$ | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 |
| | $(G_8)$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 |
| | $(G_9)$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 |
| | $(G_{10})$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 |

Thus, we performed 200 total hand gestures. We also used our captured hand gesture for gesture recognition in testing. We validate our data set with the NTU training set. The recognition accuracy of our captured data set is 100% and is shown in confusion Table 2. We have shown the comparison of our proposed work with other methods in state of the art in Table 3. As compare to other methods our system achieves best recognition accuracy. Since we do not use any markers or wrist bands to delimit the hand, a part of the arm can also be detected as the gesturing arm.

**Table 3.** Comparison of the proposed framework with others state of the art method on NTU data set

| Paper | Algorithm | Recognition rate | Recognition time |
|---|---|---|---|
| Part-based hand gesture recognition [1] | Thresholding Decomposition + FEMD | 93.2% | 0.0750 s |
| | Near convex decomposition + FEMD | 93.9% | 4.0012 s |
| An image-to-class dynamic time warping approach for both 3D static and trajectory hand gesture recognition [5] | Image to class dynamic time warping approach (I2CDTW) | 90.5% | NA |
| Histogram of 3D facets: a characteristic descriptor for hand gesture recognition [13] | HOG | 93.1% | NA |
| | H3DF | 95.5% | |
| **Our proposed approach** | **Matching with Finger label + number of open finger** | **98%** | 0.03 s |
| **Own captured dataset** | | **100%** | 0.03 s |

## 9 Conclusion

We propose an unsupervised learning based hand gesture recognition framework using depth data from a RGBD senor, such as Microsoft Kinect. The gestures that we have considered are simple and our algorithm is independent of the orientation of the arm, making it easy for an assistive living application. It is completely unobtrusive and does not require the user to wear any markers or wrist band. Moreover, the gestures are formed by extension or opening of different fingers and are classified using the knowledge of which fingers are open and the number of fingers that are open, making it applicable for people who have limited hand and arm movement. Our system is completely unsupervised and does not require labeled data. Experimental result shows that our system is able to recognize these gestures correctly. Moreover our system works in real time since the test data is matched based on matching only the finger labels. Since the gestures are differentiated on the basis of the extended fingers, it assists people in communicating through movement of the fingers only making it suitable for the assistive living paradigm.

## References

1. Ren, Z., Yuan, J., Meng, J., Zhang, Z.: Robust part-based hand gesture recognition using kinect sensor. IEEE Trans. Multimedia **15**(5), 1110–1120 (2013)
2. Wang, C., Liu, Z., Chan, S.C.: Superpixel-based hand gesture recognition with Kinect depth camera. IEEE Trans. Multimedia **17**(1), 29–39 (2015)
3. Dominio, F., Donadeo, M., Zanuttigh, P.: Combining multiple depth-based descriptors for hand gesture recognition. Pattern Recogn. Lett. **50**, 101–111 (2014)
4. Zhou, Y., Jiang, G., Lin, Y.: A novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recogn. **49**, 102–114 (2016)
5. Cheng, H., Dai, Z., Liu, Z., Zhao, Y.: An image-to-class DTW approach for both 3d static and trajectory hand gesture recognition. Pattern Recogn. **55**, 137–147 (2016)
6. Tan, D.J., Cashman, T., Taylor, J., Fitzgibbon, A., Tarlow, D., Khamis, S., Izadi, S., Shotton, J.: Fits like a glove: rapid and reliable hand shape personalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5610–5619 (2016)
7. Yang, W., Kong, L., Wang, M.: Hand gesture recognition using saliency and histogram intersection kernel based sparse representation. Multimedia Tools Appl. **75**, 1–14 (2016)
8. Keskin, C., Kıraç, F., Kara, Y.E., Akarun, L.: Real Time Hand Pose Estimation Using Depth Sensors, pp. 119–137. Springer, London (2013)
9. Oniga, S., Tisan, A., Mic, D., Buchman, A., Vida-Ratiu, A.: Hand postures recognition system using artificial neural networks implemented in FPGA. In: 30th International Spring Seminar on Electronics Technology, pp. 507–512 (2007)
10. Modler, P., Myatt, T.: Recognition of separate hand gestures by TDNN based on multi-state spectral image patterns from cyclic hand movements. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, pp. 1539–1544 (2008)
11. Stergiopoulou, E., Papamarkos, N.: Hand gesture recognition using a neural network shape fitting technique. Eng. Appl. AI **22**(8), 1141–1158 (2009)
12. Ohn-Bar, E., Trivedi, M.M.: HGR in real-time for automotive interfaces: a multimodal vision-based approach and evaluations. IEEE Trans. on ITS **15**(6) (2014)
13. Zhang, C., Yang, X., Tian, Y.: Histogram of 3D facets: a characteristic descriptor for hand gesture recognition. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 1–8 (2013)
14. Ren, Z., Meng, J., Yuan, J., Zhang, Z.: Robust hand gesture recognition with Kinect sensor. In: Proceedings of the 19th ACM International Conference on Multimedia (2011)