DBSCAN for Hand Tracking and Gesture Recognition

Wisnu Aditya¹, Herman Tolle², Timothy K. Shih³

¹Faculty of Computer Science, University of Brawijaya, Malang, Indonesia,
²Department of Computer Science and Information Engineering, National Central University, Taiwan

¹wisnuadity@gmail.com, ²emang@ub.ac.id, ³tshih@g.ncu.edu.tw

Received 30 December 2019; accepted 08 May 2020

Abstract. Hand segmentation and tracking are important issues for hand-gesture recognition. Using depth data, it can speed up the segmentation process because we can delete unnecessary data like the background of the image easily. In this research, we modify DBSCAN clustering algorithm to make it faster and suitable for our system. This method is used in both hand tracking and hand gesture recognition. The results show that our method performs well in this system. The proposed method can outperform the original DBSCAN and the other clustering method in terms of computational time.

Keywords: Object Tracking, Gesture Recognition, DBSCAN

1 Introduction

An accurate object tracking depends on the condition of the input data. The good data can be obtained through a preprocessing or segmentation method. A lot of research indicates that these methods can provide a good result[1], but this method will meet difficulties when detecting the subjects when the background is cluttered or with articulated poses. That condition can decrease the accuracy, moreover using an unnecessary information will lead to an increasing of computational time and affect the accuracy[2]. Some of the algorithms needs high computation time when it is implemented[3][4], so it is difficult to be used on real-time object tracking systems.

More recently, depth data is found to be very beneficial. Compared to the color data, depth data has some advantages providing additional 3D information that offers more detail and important hints[5]. Depth camera can work in dark conditions, reduce ambiguity in scale, is largely color and texture invariant, and resolves some silhouette ambiguities. They also perform as a pre-processing by simplifying the background subtraction[6]. All of the advantages help to process the data and make it faster.

Developing hand gesture recognition system system should pay attention to the hand tracking as an important part. Hand detection or tracking must successful segmenting hand as an input to provide the best result of the recognition algorithm[7]. Considering the importance of clustering in image segmentation then it should choose the right method to be used[8]. The depth data produced by a depth camera is not always clear without noise[9], moreover most of these cameras do have high noise levels on a raw depth map[10]. Unexpected and large amounts noise can be a problem...
and may limit the information[11][12], so we need to eliminate it to obtain a good results. Many applications need a fast method to solve the clustering problems. Currently many methods have been used as a clustering method such as DBSCAN, K-Means[13], Mean Shift [14], and OPTICS[15]. The DBSCAN method is believed to be the fastest method of all those methods and it able to clustering data while reducing the existing noise[16]. In this research, we propose a modified Density-based spatial clustering of applications with noise (DBSCAN) as a data clustering algorithm to do a fast hand tracking and hand gesture recognition.

2 Related Work

Image segmentation is an important step in image processing, moreover it is related with the image recognition system. A meaningful part of the image is obtained using image segmentation process. The most problem that frequently occur is the background is similar to the object[17]. Currently some images also have depth data, with this data we can make segmentation more easily as in some existing research[18][19].

Generally speaking, hand tracking and hand gesture recognition is challenging problem to solved. There are so many methods used for clustering like k-means and hierarchical, and expectation-maximization(EM)[20]. There are also broadly use classification method such as Nearest Neighbor, Support Vector Machines and Linear Discriminant Analysis(LDA) for recognition problem[21]. Here we proposed a different approach to solve this problem. We use DBSCAN that originally used for clustering method to handle the object tracking and recognition problem. Using clustering method, make us more flexible against the additional of class while the system running, and it is no need to use a data training to solve the problems. Instead of only use a vanilla DBSCAN[22], we do some modification to make it faster and fit for real-time system.

3 Proposed Method

This system has three main process. First, is a segmentation process that use a depth threshold for the depth data as an input. This step resulting a depth data that only consist the hand object. Second, is a tracking process that use our proposed method which is the modified version from DBSCAN to separate the hand into left and right. To maintain the tracking result, do the calculation by find a closest distance between the center point of current and previous label. Third, recognition process that also use our proposed method to calculate the number of the raised up fingers. The overall process is shown in Figure 1.

Fig. 1. System Architecture

The proposed method has a similar with the original DBSCAN do the iteration until all data is clustered. DBSCAN that we proposed, finish the clustering process for all data without a repetition for the data that not clustered yet. This method runs in
one turn of iteration for all data and it able to clusters the data in the same time. The detail explanation is shown by algorithm 1. Here we set a minimum neighbor and epsilon as one.

Algorithm 1: Modified DBSCAN

```
data input = Matrix of x and y position
list= list of position of non-zero data
Set label of the first position data as cluster 1

Repeat all data in list

    If surrounding of the current data is not empty
        Set the label according to the smallest label among these data
        If the cluster of surrounding data is different
            Set all the cluster member into the same label
        Else
            Set the cluster as a new cluster

end.
```

3.1. Segmentation

Segmentation process is an important process for object tracking and hand gesture recognition. It is performed as a preprocessing step to make an input data more stable and easier to use. In this system, we use simple depth threshold as a segmentation to remove the background from the original depth data $D_t$. Fixed threshold value $T$ defined, any object that have value less than the threshold will be considered as a target. We assuming the target as a person that want to perform a gesture. The average depth value $A_{depth}$ of the target used as a depth reference and anything in front of it will be considered as a hand. The depth value $A_{depth}$ can be calculated as equation 1.

$$A_{depth} = \frac{\sum_{j=1}^{n} D_t(j)}{n}$$  \hspace{1cm} (1)

![Fig. 2. Data Segmentation Process](image)

(a) Original depth image, (b) After background removal, and (c) Hand segmentation

Data segmentation process is illustrated in Figure 2 which (a) as original depth image where the person (including his hand) and the area behind the target are
detected. (b) is the image after background removal using depth threshold, and (c) is a hand detection result as the last result of segmentation process. In this figure we can see the different color that indicate the different value of the depth.

3.2. Object Tracking

DBSCAN is an unsupervised clustering which means they do not create the fix labels or representations for the cluster [23]. This method only separates the object into clusters and if the data is refreshed in next frame the labels maybe change. In this step we have two labels for each frame as a hand area $H$ for both hands. For each area we find a center point $C$ by calculating the average of the point inside the hand area, like in equation 2. This center point is represented by a red dot as in Figure 3.

$$C = \frac{1}{K} \sum_{i} H_i$$  \hfill (2)

$$D = \sqrt{(C3 - C1)^2}$$  \hfill (3)

Fig. 3. Hand Center Point $C$ (Red Circle)

Fig. 4. Hand Tracking

This center point is used to determine the label by comparing the distance of current frame center points with the previous frame. Figure 4 shows the hand in white color as a previous frame and hand in blue color is a current frame. Each hand has a center point $C_1$, $C_2$ are in previous frame and $C_3$, $C_4$ are in current frame. We
calculate the distance of one center point in current frame with both of center points in
previous frame. We use Euclidian distance to calculate $D$ as in equation 3. Let we
choose a center point $C_3$ of current frame then we calculate its distance by $D_1$ from
$(C_3, C_1)$ and $D_2$ from $(C_3, C_2)$. If $D_1$ is less than $D_2$, then $C_3$ has the same label as $C_1$
and vice versa.

3.3. Hand Gesture Recognition

Hand gesture recognition is a way to know or differentiate between various hand
poses. Here we can recognize a gestures according to the fingers which raised and can
count up to 10 fingers raised for the total number from both hands. The recognition,
only need a hand region as an input. To keep the focus only on the hand, we crop the
input according to the width ($X$) and height ($Y$) of the both hands that refer to center
point $C$, by the equation 4 below to get the crop area $Cr$ for all hands like conceived
in Figure 5.

\[ Cr = \text{crop}(H, X, Y) \quad (4) \]

The hand center point $C$ is used to get the hand palm area $P$. We assume that this
center point is the center of the hand palm and use it to remove the area inside the
radius distance $R$ from the center point of the hand like illustrated by Figure 6. The
value of $R$ is important for the recognition, if the value is too small then several
fingers will unite. In the other hand if it is to large some of the fingers will be
removed. Both of condition can lead to a wrong recognition result, so it is very
important to set the correct $R$ value.

The fingers area $F$ obtained by using equation 5. The calculation of the fingers as a
gesture is using our proposed DBSCAN algorithm. However, we know from the
image that there is some part of hand $N$ which cannot be removed, so we consider this
as a noise and we do not count it as a finger.

\[ F = H \not\in P \quad (5) \]

The gesture is not affected by which fingers is rising up. Figure 7 shows the hand
gestures in a different fingers position, however some of them is the same gesture as
long as the number of fingers is the same. In that figure we can see that the gesture 2,
3 and 4 have a different finger that rising up.
4 Experimental and Result

Computational time is one of the issues that need to be solved. Here we provide some improvement in our proposed method to solve the issue. The comparison of computational time shown in Table 1, it is proving that our method is outperforming the other method in computational time and compared to the original DBSCAN by 192.5% times speedup on average.

| Resize | Total Point | DBSCAN | K-Means | OPTICS | Mean Shift | Proposed Method |
|--------|-------------|--------|---------|--------|------------|-----------------|
| 1      | 41560       | 0.26742| 0.26706 | 133.0831| 23.38776   | 0.13565         |
| 0.5    | 11890       | 0.08637| 0.13333 | 12.05357| 6.497829   | 0.03889         |
| 0.25   | 3538        | 0.01992| 0.03611 | 1.679686| 2.053788   | 0.01195         |
| 0.125  | 1156        | 0.01098| 0.02830 | 0.3948571| 0.584656  | 0.00598         |

However, to make it more reliable for real-time system it needs a faster result. So, we decide to reduce the input size by resize it. The smaller input size produce faster computational time. The fastest computational time while using our method was found to be about 5ms, but the data is too small, then it makes some important part of the data will be missing and it is affects the recognition result. We try various input size as shown by Figure 8, and choose the input with 0.25 resize that can produce a stable result and the smallest size input.
Table 2 shows the confusion matrix from testing result. The testing conducted by performs fifty times test per gesture. This matrix can be used to measure how accurate the method by calculate the Precision, Recall and Accuracy. The Precision shows an average of around 80% among those predicted positive for all gestures. The Recall represents the actual positives or true positive of the result, with an average value of 81%. The Accuracy observe correctly predicted against the total testing result and it shows 90.3% in average. Based on these result we can see that our method can achieve a good result in a recognition task. The recognition is also robust due to the rotation. It is because we can handle the hand rotation problem by preprocessing the hand data before it feeds on our proposed method. Figure 9 shows this method can handle the rotation.

5 Conclusion
This paper describes the possibilities of using DBSCAN as a method to handle an object tracking and gesture recognition problems. The result show it is robust against various hand poses and hand rotation for both problems. The proposed method outperforms another clustering method, even compared to the original DBSCAN by 192.5% times speedup on average and the recognition has a high accuracy up to 90%.

Acknowledgments. This research was conducted by join degree program, which is a collaboration between the Department of Computer Science at Brawijaya University and Multimedia Information Network Laboratory (MINELAB), Department of Computer Science and Information Engineering (CSIE) at National Central University, Taiwan. We really appreciate to the Bureau of Planning for International
Cooperation (BPKLN), that gave an opportunity and support.

Fig. 9. Remove Hand Palm Area

References

1. X. Zheng, Q. Lei, R. Yao, Y. Gong, and Q. Yin, “Image segmentation based on adaptive K-means algorithm,” *Eur. J. Image Video Process.*, vol. 2018, no. 1,(2018).
2. A. Jati, G. Singh, S. Koley, A. Konar, A. K. Ray, and C. Chakraborty, “A novel segmentation approach for noisy medical images using Intuitionistic fuzzy divergence with neighbourhood-based membership function,” *J. Microsc.*., vol. 257, no. 3, pp. 187–200, (2015).
3. C. H. Lin, C. C. Chen, H. L. Lee, and J. R. Liao, “Fast K-means algorithm based on a level histogram for image retrieval,” *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3276–3283, (2014).
4. Z. Khan, J. Ni, X. Fan, and P. Shi, “An improved K-means clustering algorithm based on an adaptive initial parameter estimation procedure for image segmentation,” *Int. J. Innov. Comput. Inf. Control*, vol. 13, no. 5, pp. 1509–1526, (2017).
5. M. H. Khan, K. Shirahama, M. S. Farid, and M. Grzegorzek, “Multiple human detection in depth images,” *2016 IEEE 18th Int. Work. Multimed. Signal Process. MMSP 2016*, pp. 1–6, (2017).
6. J. Shotton *et al.*, “Efficient human pose estimation from single depth images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, (2013).
7. U. Soni, A. Trivedi, and N. Roberts, “Real-Time hand tracking using integrated optical flow and CAMshift algorithm,” *Proc. - 2016 2nd IEEE Int. Conf. Res. Comput. Intell. Commun. Networks, ICRCICN 2016*, pp. 135–140, (2017).
8. J. Hou, C. Sha, L. Chi, Q. Xia, and N. M. Qi, “Merging dominant sets and DBSCAN for robust clustering and image segmentation,” *2014 IEEE Int. Conf. Image Process. ICIP 2014*, pp. 4422–4426, (2014).
9. K. Essmaeel, L. Gallo, E. Damiani, G. De Pietro, and A. Dipandà, “Temporal denoising of Kinect depth data,” *8th Int. Conf. Signal Image Technol. Internet*
Based Syst. SITIS 2012r, pp. 47–52, (2012).

10. A. Chatterjee and V. M. Govindu, “Noise in Structured-Light Stereo Depth Cameras: Modeling and its Applications,” arXiv:1505.01936, (2015).

11. R. Chaudhary and H. Dasgupta, “An Approach for Noise Removal on Depth Images,” arxiv.org/1602.05168, no. 1, pp. 1–2, (2016).

12. G. Halmetschlager-Funek, M. Suchi, M. Kampel, and M. Vincze, “An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments,” IEEE Robot. Autom. Mag., vol. 26, no. 1, pp. 67–77, (2019).

13. C. D. Megawati, E. M. Yuniarno, and S. M. S. Nugroho, “Clustering of Female Avatar Face Features Consumers Choice using KMeans and SOM Algorithm,” Proc. - 2019 Int. Semin. Intell. Technol. Its Appl. ISITIA 2019, pp. 366–370, (2019).

14. N. Fatemi, H. Sajedi, and M. E. Shiri, “Salient object detection with segment features using mean shift algorithm,” 2018 8th Int. Conf. Comput. Knowl. Eng. ICCKE 2018, no. Iccke, pp. 20–26, (2018).

15. A. Omrani, K. Santhisree, and Damodaram, “Clustering sequential data with OPTICS,” 2011 IEEE 3rd Int. Conf. Commun. Softw. Networks, ICCSN 2011, pp. 591–594, (2011).

16. H. K. Kanagala and V. V. Jaya Rama Krishnaiah, “A comparative study of K-Means, DBSCAN and OPTICS,” 2016 Int. Conf. Comput. Commun. Informatics, ICCCI 2016, pp. 1–6, (2016).

17. Y. Li, J. Zhang, P. Gao, L. Jiang, and M. Chen, “Grab Cut Image Segmentation Based on Image Region,” 2018 3rd IEEE Int. Conf. Image, Vis. Comput. ICIVC 2018, pp. 311–315, (2018).

18. R. Augustauskas and A. Lipnickas, “Robust hand detection using arm segmentation from depth data and static palm gesture recognition,” Proc. 2017 IEEE 9th Int. Conf. Intell. Data Acquis. Adv. Comput. Syst. Technol. Appl. IDAACS 2017, vol. 2, pp. 664–667, (2017).

19. N. L. Hakim, T. K. Shih, S. P. K. Arachchi, W. Aditya, Y. C. Chen, and C. Y. Lin, “Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model,” Sensors (Switzerland), vol. 19, no. 24, (2019).

20. M. Z. Rodriguez et al., “Clustering algorithms: A comparative approach,” PLoS One, vol. 14, no. 1, pp. 1–31, (2019).

21. R. M. Ştefan, “A Comparison of Data Classification Methods,” Procedia Econ. Financ., vol. 3, no. 12, pp. 420–425, (2012).

22. C. Bodenstein, M. Gotz, A. Jansen, H. Scholz, and M. Riedel, “Automatic Object Detection Using DBSCAN for Counting Intoxicated Flies in the FLORIDA Assay,” 2016 15th IEEE Int. Conf. Mach. Learn. Appl., pp. 746–751, (2017).