MALAYSIAN JOURNAL OF SCIENCE
MJS
ISSN 1394-3065

Regular Issue

Malaysian Journal of Science 42 (3): 75-84 (Oct 2023)

https://mjs.um.edu.my

POINT CLOUD SIMPLIFICATION METHOD BASED ON IMPROVED FUZZY C-MEANS WITH AUTOMATIC NUMBER OF CLUSTERS

Siti Nur Athirah Adibah Maiddin[1a] and Normi Abdul Hadi[2a]*

Abstract: The essence of the point cloud is to express geometric information about objects by getting discrete coordinates on their surfaces. However, with a million points, this data may record redundant details in which it is needless to be kept for the model's analysis. In addition, there is also a limitation where the processors cannot process the large-size datasets. The point cloud simplification algorithm was developed to solve the stated obstacles in data processing. Numerous algorithms have been published to produce the best methods for data reduction process. Since the simplification process might eliminate essential features of the data, this study introduces the features preservation process to keep the important points before the simplification. This study employed the Fuzzy C-Means (FCM) algorithms for the simplification stage due to their simplicity and ability to generate an accurate result. Regardless, the FCM still suffers from drawbacks, where their initial cluster centres are prone to fall into local optima. This study improved the FCM by employing the Score and Minimum Distance (SMD) to determine the number of clusters and cluster centres. The SMD is enhanced by changing the Gaussian to Cosine kernel function to increase the accuracy. This new technique is named SMD(C)-IFCM. The method was then applied to the 3D point cloud of a box, cup, and Stanford bunny. The performance of the developed method was compared with the original SMD-FCM and SMD-IFCM for the percentage of the simplified data, error evaluation, and processing time. The result and analysis showed that the developed method had the best score, which was six (6) out of nine (9) measurements, compared to the other two methods with scores of one (1) and two (2) respectively. This score suggests that the developed algorithm successfully reduced the error evaluation and the processing time to generate the output.

Keywords: *Point cloud simplification, fuzzy C-means, automatic number of cluster, and preservation strong features*

1. Introduction

A point cloud can be described as a collection of data points arranged in space. A three-dimensional (3D) structure or object might be represented by points, each with its Cartesian coordinate, which illustrates each point position (Wang & Kim, 2019). The source of point clouds can be from a 3D scanner or a photogrammetry software, and other devices.

Point clouds are used for various applications, including creating 3D Computer-Aided Design (CAD) models for manufactured components, metrology and quality inspection, and a variety of visualisation, animation, rendering, and mass customisation. In order to ensure that the result is reliable for later applications, the process of point cloud simulation must be precise (Hadi & Alias, 2019). However, by creating thousands of point clouds for each data set, the procedure of point cloud analysis might face some interference. Moreover, storage space and the processing time in analysing the point clouds are likely to be vast and prolonged, and a standard processor such as a laptop will reach its memory bottleneck. Removing random data from the dataset may eliminate the essential features of the data and give misleading information. It is crucial to preserve the features of an object since it gives measurements that specify the characteristics of an object (S. A. Halim et al., 2021).

On that account, the technique of point cloud simplification was introduced to eliminate duplicate data points and reserve the essential and meaningful point position (Mahdaoui & Sbai, 2020). Based on the literature, simplification algorithm can be done using two approaches of creating polygonal networks on the point cloud data or using geometric details. However, several studies focused on the second approach as setting up polygonal networks can be extremely expensive in calculation and storage. The clustering process can be a handy approach in differentiating the necessary spots in the point cloud data sets.

Accordingly, several clustering approaches have been published to simplify the point clouds while guaranteeing that the results will be accurate. One of the methods in data clustering that many researchers utilised was the Fuzzy C Means (FCM) algorithm, because it was proven to give satisfactory results. However, FCM still encountered drawbacks as their initial cluster centre was unstable, making it easy to fall into local optimal (Alomoush et al., 2021; Krishna & Bhaskari, 2016; Yang et al.,

Authors information:
[a]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40170 Shah Alam, Selangor, MALAYSIA. E-mail: tyrahdybah@gmail.com[1]; normi@fskm.uitm.edu.my[2]
*Corresponding Author: normi@fskm.uitm.edu.my

Received: June 23, 2022
Accepted: January 25, 2023
Published: October 31, 2023

DOI:https//doi.org/10.22452/mjs.vol42no3.10

2020). Considering that, the clustering results can be inaccurate. Therefore, improvement steps need to be done in the FCM to make the clustering process more trustworthy.

Another stumbling block discovered in the FCM is that most of the existing methods used random cluster numbers and did not preserve the essential features, significantly influencing the simplification results. Thus, this can make the algorithm remove a wrong point cloud instead, and consequently, the output of the simplified point clouds could be unreliable.

Therefore, this study improves the simplification method by generating an automatic number of clusters by utilising the Score and Minimum Distance of the Point Clouds (SMD). The SMD model by He et al. (2019) became the chosen method since it is an established technique in scoring each point cloud based on the density function in getting an optimal number of clusters. Rating the point clouds is the best conceptual method in generating the number of clusters. In addition, the method is also considered the theory of minimum distance as the cluster centres should not be close to each other to avoid unnecessary cluster creation.

Moreover, the original SMD is improved by changing the Gaussian Kernel Function to Cosine Kernel Function, which has a higher accuracy, and the feature-preservation process is done before the simplification. This improvement can reduce errors from the wrong clustering process caused by a random number of clusters. Consequently, the output of a simplified algorithm can be more reliable as this process only retains an important point position. The formulation of the cluster number can make the whole process done automatically. The following are the study's main contributions:

1) The number of cluster centres can be generated automatically without needing a user to decide beforehand. The SMD model is embedded into the algorithm, which will give the number of clusters and initial cluster centres that can overcome the drawbacks of FCM.
2) The algorithm also includes the techniques of preserving strong feature information of the point cloud, before being divided through the FCM. Consequently, the output of the simplification is the reduced point cloud data with the features reserved.
3) The discussion analysis has shown that the algorithm is effective to some extent.

The flow of this paper is organized by section. Section 2 reviews some works on the point cloud simplification and the FCM. The steps taken in the developed algorithm are illustrated in Section 3. The following section is where the result and discussion of the three datasets occurred. Lastly, the summary of this study and the suggested further direction can be found in Section 5.

2. Literature Review

A considerable amount of literature has been published on point cloud simplification. Among these papers, Xuan et al. (2018) described that point cloud simplification is a process that needs to be done to remove the redundant points while retaining the main geometric features of the object. Scanning the surface of a physical object using a 3D technology could produce a massive number of point clouds (Halim et al., 2017; Leal et al., 2017).

Thus, to reduce the storage and computing problems, the point cloud simplification process plays an important role. In short, point cloud simplification can be defined as a process to eliminate duplicate data points while selecting important and meaningful 3D points (Mahdaoui & Sbai, 2020). In general, Wu et al. (2021) stated that there are two categories of point cloud simplification methods: (i) mesh-based, and (ii) mesh-free. The authors explained that mesh-based methods were constructed to set up polygonal networks on the point cloud data and cut down the point connectivity information. Meanwhile, the mesh-free approaches seek to assess the local neighbour information of cloud points and resample important spots among original point cloud data using geometric information. However, mesh-based methods are suffering from extreme complexity of computational issues since these approaches require geometric and topological computation (Ji et al., 2019; Wu et al., 2021). For that reason, the main focus of this research is to simplify the point cloud data using mesh-free methods or, more precisely, clustering algorithms.

Most previous studies defined clustering as a process of dividing a set of data into clusters so that images in the same cluster are as similar as possible, while images from other clusters are as distinct as possible (Askari, 2021; Zhang et al., 2020). Mahdaoui & Sbai, (2020) described clustering as statistical dissection techniques used to categorise raw data into homogeneous groups. According to Askari (2021), clustering methods are classified into many types such as cluster-centric algorithms, K-means, and FCM, which determine the centres of groupings of data points using distance to measure similarity and dissimilarity. Density-based clustering could also mark adjacent data points with a particular concentration into a single cluster based on pre-set density and neighbourhood criteria.

Nevertheless, Askari (2021) also stated that FCM is preferred by many researchers as the method used in data clustering. This view is supported by Zhang et al. (2020), who said that FCM developed from fuzzy logic is one of the most outstanding clustering methods due to its simplicity and efficacy. Despite that, the FCM still suffers from certain drawbacks that the researchers are still interested in solving. The literature review by Askari (2021) stated that the FCM is extremely sensitive to noise, outliers, and cluster size. The authors explained that there was no specific calculation to filter out the noise and outliers from necessary points since all data points equally contribute to the estimation of cluster centres which affects the actual structures of the point cloud objects.

The FCM could be bad in clustering due to the random initial cluster centres. Therefore, several modifications have been made to the FCM algorithm by combining it with other models, such as Gravitational Search Algorithm (GSA) (Yang et al., 2020), Particle Swarm Optimization (PSO) algorithm (Siringoringo & Jamaluddin, 2019), and Artificial Bee Colony algorithm (Alomoush et al., 2021).

As stated before, the FCM is the most well-known method among researchers. However, there is a lack of literature that tries to generate an optimal number of clusters for fuzzy clustering in the point cloud simplification process. Nearly all studies used an arbitrary number of clusters that had to be decided by the researchers beforehand (Leal et al., 2017).Therefore, this study has developed a method to estimate an automatic number of clusters by adopting the Score and Minimum Distance of the Point Clouds (SMD) technique by He et al. (2019).

3. Methodology

Figure 1 below illustrates the point simplification algorithm used in this study:

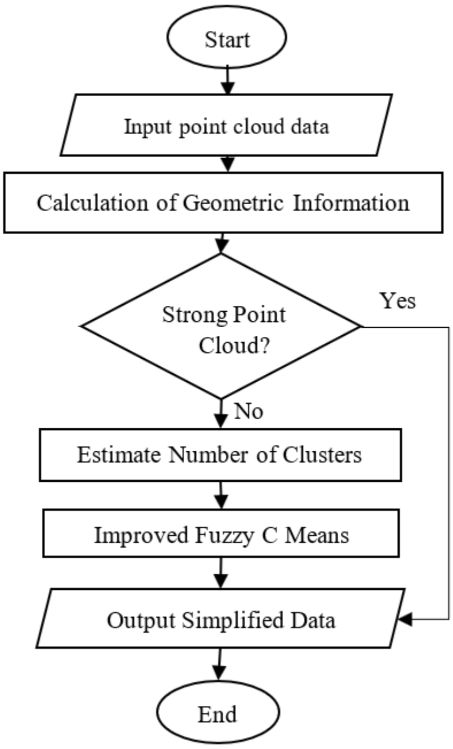


Figure 1. Flowchart of the developed point cloud simplification algorithm.

3.1 Input Point Cloud Datasets

This study uses three different datasets for experimental verification purposes: (i) box, (ii) cup, and (iii) Stanford bunny obtained from the PointCleanNet database.

3.2 Calculation of Geometric Information

Preserving point cloud features requires the spatial coordinate value of the point cloud and the geometric information, which are normal vector and point curvature. However, Yang et al. (2020) added angle entropy and point cloud density as an additional information.

3.2.1 Point Cloud Normal Vector and Curvature Estimation

The Normal vector is calculated using the Principal Component Analysis (PCA). PCA is built by the covariance matrix using the neighbourhood information.

Equations (1) and (2) below illustrate the formula used for kNN and PCA models for a given point $p(x, y, z)$, respectively.

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \qquad (1)$$

$$C^{3\times3} = \sum_{i=1}^{k} (p_i - \bar{p})(p_i - \bar{p})^T \qquad (2)$$

where k is the neighbourhood size, d is the Euclidean distance between point (x_a, y_a, z_a) and (x_b, y_b, z_b) which are the three-dimensional coordinates of point a and point b, and $\bar{p}$ is the mean value of the point p.

Then, it is suggested to adjust the normal consistency to equation (3) owing to the unpredictability of the point cloud's normal vector direction.

$$\bar{y}' = \begin{cases} \bar{y}\bar{y}(l - p_i) & if \quad \bar{y} \geq 0 \\ -\bar{y}\bar{y}(l - p_i) & if \quad \bar{y} < 0 \end{cases} \qquad (3)$$

where $\overrightarrow{y}$ and $\bar{y}'$ is the original and adjusted normal vector in the form of (h, m, n), respectively, and l is the location of the view.

The formula for the curvature σ_i of point p_i is given as,

$$\sigma_i = \frac{\tau_0}{\tau_0 + \tau_1 + \tau_2} \qquad (4)$$

where $\tau_0, \tau_1, and\ \tau_2$ are the eigenvalues from equation (2) that satisfy $\tau_0 \leq \tau_1 \leq \tau_2$.

3.2.2 Point Cloud Angle Entropy

Another method to describe the local geometric characteristics of the subject is to estimate the angles between the normal vectors of its neighboring points which known as angle entropy. According to Xuan et al. (2018), angle entropy can be used to decide which point needs to be reserved, and which does not. A high value indicates that the point is situated in an area where the concave and convex clearly differ, and the removal of the point would result in a change to the local geometry. As a result, the point is crucial in expressing the specific geometric feature in the area and must be kept. Contrarily, a low value indicates that the point is in a flat area, where it can be replaced by its neighbours and removed.

Thus, angle entropy En_i for each point cloud is estimated based on its neighbourhood as follows,

$$En_i = -\frac{\bar{\theta}_i}{\bar{\theta}_i + \sum_{j=1}^{k}\bar{\theta}_j}\log_2\frac{\bar{\theta}_i}{\bar{\theta}_i + \sum_{j=1}^{k}\bar{\theta}_j} - \sum_{j=1}^{k}\frac{\bar{\theta}_j}{\bar{\theta}_i + \sum_{j=1}^{k}\bar{\theta}_j}\log_2\frac{\bar{\theta}_j}{\bar{\theta}_i + \sum_{j=1}^{k}\bar{\theta}_j} \qquad (5)$$

$$\bar{\theta}_i = \frac{1}{k}\sum_{j=1}^{k}\arccos\left(\frac{\bar{y}_i \cdot \bar{y}_j}{|\bar{y}_i||\bar{y}_j|}\right)$$

where $\bar{y}_i$ is the normal vector of p_i, $\bar{\theta}_i$ is the angle of p_i, and $\bar{\theta}_j$ is the angle of point p_i.

3.3 Preservation of Point Cloud Strong Features

Strong features are essential to retain the unique nature of the cloud model data. Thus, this study adopts the statistical principle together with point cloud curvature as the strong features preservation evaluation parameter (Hadi et al., 2021). Firstly, estimate the average curvature $\bar{\sigma}$ of all point clouds and standard deviation σ_{std} using equation (6). Then, it is followed by a simplification model by Yang et al. (2020), where the curvature threshold σ_0 will be computed. If $\sigma_i > \sigma_0$, point p_i is classified as a strong feature and will be preserved from the clustering process.

$$\bar{\sigma} = \frac{1}{n}\sum_{i=1}^{n}\sigma_i$$

$$\sigma_{std} = \sqrt{\frac{\sum_{i=1}^{n}(\bar{\sigma} - \sigma_i)^2}{n}} \qquad (6)$$

$$\sigma_0 = \bar{\sigma} + \beta \times \sigma_{std}$$

where n is the total point cloud, and β is a constant from 1 to 5.

3.4 Number of Clusters

This subsection discusses an automatic method for the number of clusters. The analysis was based on a conceptual model by He et al. (2020) called the Score and Minimum Distance of the Centre Point (SMD). This model consists of four steps as below:

STEP 1: Choose the dimension that has the largest discreteness.

A dimension with the largest degree of dispersion could reflect the actual distribution of the data, which means that the overlap in this dimension is more significant. Thus, equation (8) was used to compare the level of dispersion in a dimension called the discreteness function, and equation (7) was used to normalise the data beforehand. Then, a dimension was said to have a tremendous discreteness when the discreteness function was the lowest among all other dimensions. The chosen dimension was sent to the next step to get the density values based on the kernel function.

$$x' = \frac{x - min}{max - min} \qquad (7)$$

$$F = \sum_{i=1}^{n-1}|x_{i+1} - x_i| \qquad (8)$$

Where x' is the normalised data, and F is the discreteness function.

STEP 2: Calculation of density using kernel estimation and obtained $kmax$.

Triangular Kernel function, Epanechnikov Kernel function, Gaussian Kernel function, and Tricube Kernel function are among the types of kernel functions used to estimate the density values. He et al. (2020) adopted the Gaussian Kernel density function as their method so that the density curve would be smooth. However, Wang & Kim (2019a) mentioned that the time complexity of calculating the local density using the Gaussian kernel function was indefinitely long. Moreover, the Gaussian kernel function is inefficient for large-scale datasets. Hence, this study chose the Cosine kernel function to ensure that the result is accurate with the datasets of various shapes and scales.

Once the dimension with the largest discreteness was extracted, the density values would be evaluated using equation (9). Afterwards, $kmax$ were obtained by choosing the smallest value among $\sqrt{n}$ and the number of extreme values for density

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n}K\left(\frac{x - x_i}{h}\right) \qquad (9)$$

$$K(u) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)$$

where $f(x)$ is the density distribution function, K is the Cosine Kernel function, $h = 1.059\sigma n^{-\frac{1}{5}}$ is the window width, and σ is the variance of the random variable.

STEP 3: Determining the candidate's set of cluster centre

The first step in this process was to sort the data points, c_i in descending order based on their score, as in equation (10).

$$score(c_i) = \min_{density(c_j)>density(c_i),c_i,c_j\in P} distance(c_i,c_j) \cdot density(c_i) \tag{10}$$

where $density(c_i)$ is the density of point c_i, $distance(c_i,c_j)$ is the distance between point c_i and point c_j.

Then, the sorted points were cut into the size of the $kmax$ to get the candidate's set of centre points. The process continued by judging the score change between all candidates by score variation. He et al. (2020) defined score variation as the method to find the number of clusters when the last great score change occurred, decided using Equation (11). A new value of $kmax$ is obtained when $Gradrop(k)$ is greater or equal to 2. The centre points were said to change significantly at this stage, and all the centre points in the same class will be removed.

$$\begin{gathered} CGrade(k) = Grade(k) - Grade(k+1) \\ AverCGrade = \frac{\sum_{i=1}^{n-1} CGrade(k)}{n-1} \\ CGrade(k) > AverCGrade \\ Gradrop(k) = \frac{CGrade(k)}{\max\limits_{k<i<Kmax} CGrade(i)} \end{gathered} \tag{11}$$

$CGrade(k)$ is the score change value, $Grade(k)$ is the sorted points based on Equation (10) in descending order, $AverCGrade(k)$ is the average of the top $Kmax$ score changes, and $Gradrop(k)$ is the degree of the score change.

STEP 4: Generating the final number of clusters.

The last step in estimating the automatic number of clusters would be to investigate the change of the minimum distance between the central points so that the cluster points are not close to each other. He et al. (2020) developed equation (12) to judge the minimum distance, and the cluster centre was selected when $Disdrop(c_k, c_{k+1}) \geq$ a, a ≥ 2. This is because this condition portrays that no more small distance between the two centre points occurred. Then, the final number of clusters was decided by the size of the points before the last great change in the minimum distance between the centre points.

$$Disdrop(C_k, C_{k+1}) = \frac{\min\limits_{c_i\in C_k, c_j\in C_k \text{ and } i\neq j} distance(c_i,c_j)}{\min\limits_{c_i\in C_{k+1}, c_j\in C_{k+1} \text{ and } i\neq j} distance(c_i,c_j)} \tag{12}$$

3.5 Improved Fuzzy C-Means (IFCM)

Some alterations need to be done to the established FCM so that it can overcome the drawbacks of this algorithm. This study modified the cluster centre based on the density estimation instead of the random membership matrix. The improved method is named Improved Fuzzy C Means (IFCM). The flow of IFCM is as follows:

STEP 1: Generate random initialisation of the cluster centre $V = \{v_1, v_2, \dots, v_c\}$, together with membership matrix $F = [f_{ki}]_{c\times n}$, that follows the condition of FCM.

$$f_{ki} \in [0,1], \sum_{k=1}^{c} f_{ki} = 1, 0 < \sum_{k=1}^{c} f_{ki} < n \quad \forall i = 1,2,\dots,n, \ \ \forall k = 1,2,\dots,c \tag{13}$$

Where c is the total number of clusters that was computed previously.

STEP 2: Update the membership matrix using the cluster centre generated in equation (14). Zhang et al. (2020) suggested using the fuzzy index, $m \geq 1$. However, after a further investigation, a trend was noticed between the fuzzy index and cluster centre, by which the overlapping among cluster centre occurred significantly when the fuzzy index is increasing. For that reason, this study used a fixed value for the fuzzy index, which was $m = 1.15$ to avoid all cluster centres falling into the centre of the models.

$$f_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \tag{14}$$

Where $d_{ij} = \|p_i - c_j\|$, and $d_{kj} = \|p_k - c_j\|$.

STEP 3: Determine if the calculation procedure satisfies the threshold requirements. The computation procedure is complete, and the clustering result is computed when the number of iterations reaches the total number of iterations. Return to Step 2 if the termination condition is not satisfied.

3.6 Output Simplified Data

The point cloud data in different regions were simplified to complete the simplification process of point cloud based on the angle entropy in each region. Then, all the retained points from the simplification process will be combined with strong features preserved earlier to be the simplified dataset.

DOI:https//doi.org/10.22452/mjs.vol42no3.10

4. Results and Discussions

This section discusses the result computed based on an algorithm in the previous section. Four main characteristics were analysed: (i) number of clusters, (ii) size of simplified data, (iii) error evaluation, and (iv) processing time.

4.1 Analysis of the Preservation of the Strong Features

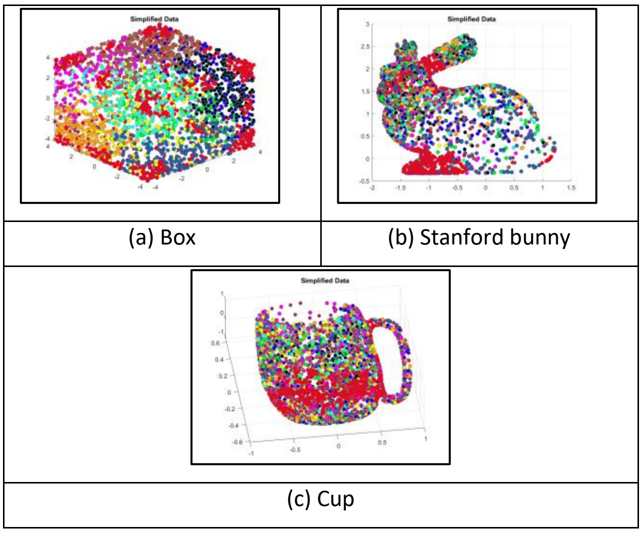
Based on the literature review, some researchers did not consider preservation of strong point clouds as one of the processes in their algorithm. Therefore, this subsection investigates the effect of preserving the critical points for each model based on the total simplified data, error evaluation, and figure comparison. The algorithm for this inspection was performed on the fixed number of clusters set to 10 and applied the established Fuzzy C-Means (FCM) clustering algorithm. The original data sets for all three models consist of 100,000 points. However, due to the processor capacity, 100,000 will be reduced to 5,000 points. The summary for this analysis is shown in Table 1.

Table 1. Analysis of Data Simplification with Strong Point Cloud Preservation.

Reservation of Strong Feature?	Number of Strong Points	Number of Clusters	Total Simplified Data	Error Evaluation	
				Maximum	Mean
BOX					
YES	711	10	2832	13.741	2.8378
NO	-	10	2469	13.7821	3.3582
STANFORD BUNNY					
YES	808	10	2659	3.9483	0.6471
NO	-	10	2216	3.9526	0.7713
CUP					
YES	725	10	3361	1.811	0.4867
NO	-	10	3111	1.811	0.5698

This result shows that the features preservation process gives the total number of final data somewhat higher than the unpreserved algorithm. Another aspect that stands out in the table is the error evaluation result. As for the maximum error, there is no comparable difference between the two algorithms. Regardless of that, it can be said that the algorithm with preservation of unique features effectively reduced the error for box and Stanford bunny. Mean errors show a significant contradiction for all three models. Thus, it is guaranteed that the features preservation step can minimise the error occurring in the point cloud simplification algorithm.

The following figures illustrate the distribution of the simplified data with the preserved points in red. Other colours in the figures show different clusters.



(a) Box

(b) Stanford bunny

(c) Cup

Figure 2. Analysis of strong features preservation on three different datasets.

Figure 2 shows the simplified models with preserved features in red colour. For the box, obviously, the strong elements are the edges. Unlike the box, the Stanford bunny has more round edges, and the method preserves points mainly at the bunny's face and legs. Similarly, the cup's strong features are located at its bottom part.

4.2 Analysis of the Developed Algorithm

This subsection discusses the outcome of the fully developed algorithm. The first modification of this study introduced a method to generate an automatic number of clusters. This modification was established by embedding the Score and Minimum Distance of the Centre Point (SMD) to rate the point clouds. The original Kernel Density Function by Wang and Kim (2019a) that used Gaussian Kernel Function was changed to the Cosine Kernel Function to increase the accuracy. At this point, the technique was named SMD-FCM because the SMD method is only used to get the number of clusters, and the cluster centres still depends on the random membership matrix.

The second modification is the cluster centre that was refined based on the score of the point clouds instead of the random membership matrix. This new technique is called the Improved Fuzzy C-Means (IFCM). The new algorithm with kernel and membership matrix modification is SMD(C)-IFCM. Table 2 shows the differences between these three algorithms.

Table 2. Differences among SMD-FCM, SMD-IFCM, SMD(C)-IFCM.

	SMD-FCM	SMD-IFCM	SMD(C)-IFCM (developed algorithm)
Kernel Density Function	**Gaussian Kernel Function**	**Gaussian Kernel Function**	**Cosine Kernel Function**
Cluster Centre	**Based on membership matrix**	**Based on score of the point clouds**	**Based on score of the point clouds**

The discussion for these three algorithms focuses on the number of cluster centres, total simplified data, error evaluation, and processing time in computing the algorithm. The results of the developed techniques are reported in Table 3.

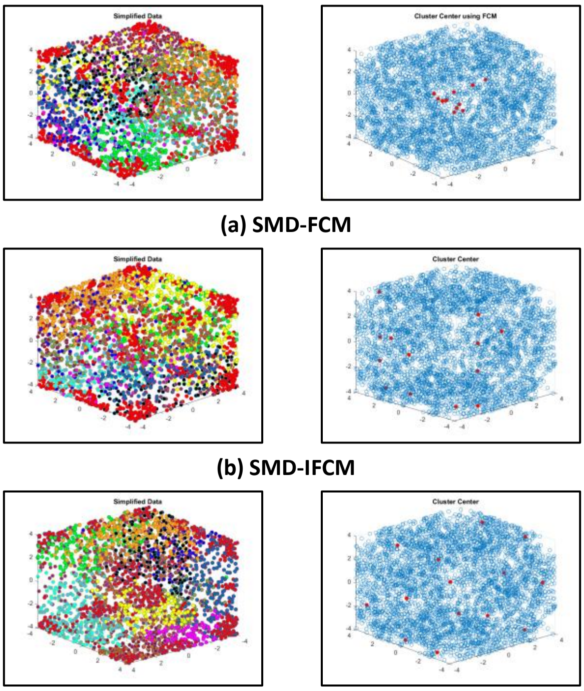
Table 3. Results of Simplified Algorithm for SMD_FCM, SMD-IFCM, and SMD(C)-IFCM.

	Number of Strong Points	Number of Clusters	Total Simplified Data	Error Evaluation		Time Taken (s)
				Maximum	Mean	
			BOX			
SMD-FCM	711	13	2860 (57.2%)	13.7821	3.9170	1756.044
SMD-IFCM	711	13	2866 (57.32%)	13.7821	3.9246	1840.031
SMD(C)-IFCM	711	14	2831 (56.62%)	13.7821	3.8726	1606.853
			STANFORD BUNNY			
SMD-FCM	808	17	2666 (53.32%)	3.9522	0.9210	802.928
SMD-IFCM	808	17	2649 (52.98%)	3.9526	0.9138	1253.575
SMD(C)-IFCM	808	10	2678 (53.56%)	3.9526	0.9247	775.776
			CUP			
SMD-FCM	725	5	3201 (64.01%)	1.811	0.5834	682.029
SMD-IFCM	725	5	3250 (65%)	1.811	0.5931	795.361
SMD(C)-IFCM	725	8	3174 (63.48%)	1.811	0.5783	717.929

The result shows that a change in Kernel Function affects the simplified result in which the total number of clusters can be different. The datasets are prone to be reduced more in the model with more clusters. For box and cup, the error evaluation between SMD-FCM and SMD-IFCM indicates that the Cosine Kernel Function produces a better result when applied to the IFCM technique. However, the SMD(C)-IFCM performed inadequately for the Stanford bunny, and it may be related to other parameters involved in the whole process, such as the kNN value. The overall performance showed that the SMD(C)-IFCM scored six (6) out of nine (9) measurements, where the SMD-FCM and SMD-IFCM scored one (1) and two (2) respectively. Note that the maximum error was not considered in the measurement since it showed an insignificant difference between the methods.

DOI:https//doi.org/10.22452/mjs.vol42no3.10

Figures 3 to 5 display the simplified data and the cluster centres accordingly.

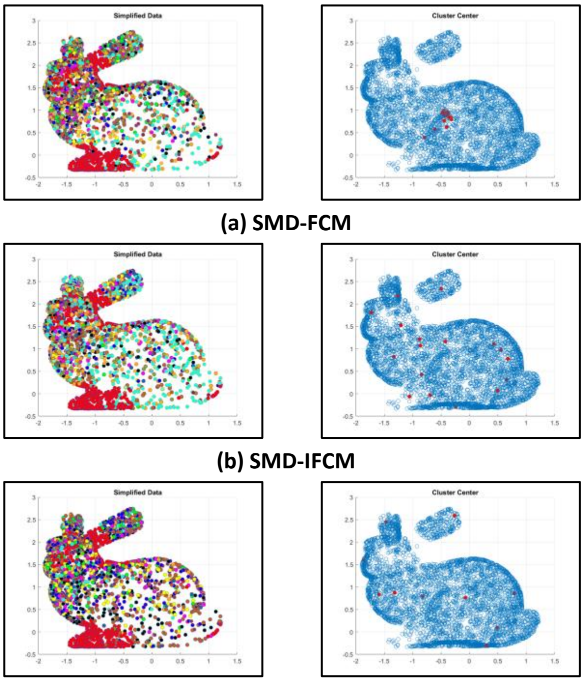


(a) SMD-FCM

(b) SMD-IFCM

(c) SMD(C)-IFCM

Figure 3. Analysis of simplified data and cluster centre for the box using three different methods.



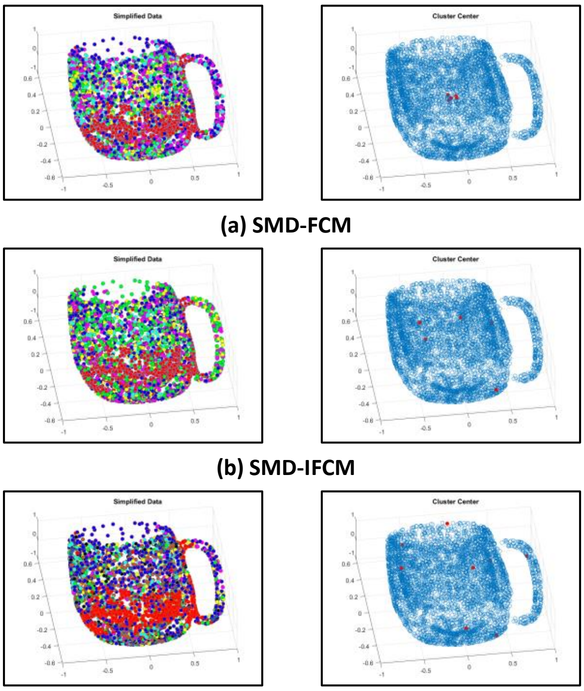
(a) SMD-FCM

(b) SMD-IFCM

(c) SMD(C)-IFCM

Figure 4. Analysis of simplified data and cluster centre for the Stanford bunny.

DOI:https//doi.org/10.22452/mjs.vol42no3.10



(a) SMD-FCM

(b) SMD-IFCM

(c) SMD(C)-IFCM

Figure 5. Analysis of simplified data and cluster centre for the cup.

In Figures 3 to 5, (a) is the resulted datasets and cluster centre using SMD-FCM, (b) is the resulted point clouds and cluster centre using SMD-IFCM, and (c) is the simplified model and cluster centre using the new model developed in this study which is the SMD(C)-IFCM. For SMD-FCM, the cluster centres for all datasets have gravitated to the model's centre, which shows the drawback of FCM. Meanwhile, for the SMD-IFCM, the distribution of cluster centres improved and got better with the SMD(C)-IFCM.

5. Conclusion

This study has developed the point cloud simplification algorithm named SMD(C)-IFCM with features preservation. The main characteristic of SMD(C)-IFCM is that it generates an automatic number of clusters according to the Cosine Kernel Function as the density estimator. Furthermore, the cluster centre for this algorithm is based on the score of the point clouds instead of a random membership matrix.

This study was carried out by analysing the results of the developed algorithm with the two other algorithms: (i) SMD-FCM, and (ii) SMD-IFCM. The analysis shows that the number of clusters mainly influences the total points after the simplification process. It can be observed that more data will be removed with a larger number of clusters. The results show that SMD-FCM had the weakest performance with one (1) score. This was due to the cluster centre in the methods of FCM being unstable and falling into the local optima. SMD-IFCM showed a better performance with a score of two (2). SMD(C)-IFCM had the best performance with a score of six (6). In this method, the cluster centres were distributed fairly. Furthermore, the number of cluster centres were automatically calculated, and the significant features were preserved before the simplification.

For improvements, future studies should focus on kNN as this parameter has a huge impact on the preservation process.

6. Acknowledgement

This research is supported by FRGS-RACER (600-IRMI/FRGS-RACER 5/3 (053/2019) and RACER/1/2019/STG06/UITM//4).

7. References

Alomoush, W., Alrosan, A., Almomani, A., Alissa, K., Khashan, O. A., & Al-Nawasrah, A. (2021). Spatial information of fuzzy clustering based mean best artificial bee colony algorithm for phantom brain image segmentation. *International Journal of Electrical and Computer Engineering*, *11*(5). https://doi.org/10.11591/ijece.v11i5.pp4050-4058

Askari, S. (2021). Fuzzy C-Means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: Review and development. *Expert Systems with Applications*, *165*. https://doi.org/10.1016/j.eswa.2020.113856

Hadi, N. A., & Alias, N. (2019). 3-Dimensional Human Head Reconstruction Using Cubic Spline Surface on CPU-GPU Platform. *Proceedings of the 2019 4th International Conference on Intelligent Information Technology*, 16–20. https://doi.org/10.1145/3321454.3321476

Hadi, N. A., Halim, S. A., & Alias, N. (2021). Statistical filtering on 3D cloud data points on the CPU-GPU platform. *Journal of Physics: Conference Series*, *1770*(1). https://doi.org/10.1088/1742-6596/1770/1/012006

Halim, M. S. A., Hadi, N. A., Sulaiman, H., & Halim, S. A. (2017). An Algorithm for Beta-Spline Surface Reconstruction from Multi Slice CT Scan Images using MATLAB pmode. *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 1–6. https://doi.org/10.1109/ISCAIE.2017.8074939

Halim, S. A., Hadi, N. A., & Mat Lazim, N. S. (2021). Segmentation and features extraction of Malaysian herbs leaves. *Journal of Physics: Conference Series*, *1770*(1), 1–7. https://doi.org/10.1088/1742-6596/1770/1/012005

He, Z., Jia, Z., & Zhang, X. (2019). A fast method for estimating the number of clusters based on score and the minimum distance of the center point. *Information*, *11*(16), 1–14.

He, Z., Jia, Z., & Zhang, X. (2020). A fast method for estimating the number of clusters based on score and the minimum distance of the center point. *Information (Switzerland)*, *11*(1). https://doi.org/10.3390/info11010016

Ji, C., Li, Y., Fan, J., & Lan, S. (2019). A novel simplification method for 3d geometric point cloud based on the importance of point. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2939684

Krishna, P. G., & Bhaskari, D. L. (2016). Fuzzy C-means and fuzzy TLBO for fuzzy clustering. *Advances in Intelligent Systems and Computing*, *379*. https://doi.org/10.1007/978-81-322-2517-1_46

Leal, N., Leal, E., & German, S. T. (2017). A linear programming approach for 3D point cloud simplification. *IAENG International Journal of Computer Science*, *44*(1).

Mahdaoui, A., & Sbai, E. H. (2020). 3D point cloud simplification based on k-nearest neighbor and clustering. *Advances in Multimedia*, *2020*.

Siringoringo, R., & Jamaluddin, J. (2019). Initializing the Fuzzy C-Means Cluster Center with Particle Swarm Optimization for Sentiment Clustering. *Journal of Physics: Conference Series*, *1361*(1). https://doi.org/10.1088/1742-6596/1361/1/012002

Wang, Q., & Kim, M.-K. (2019). Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, *39*, 306–319.

Wu, Z., Zeng, Y., Li, D. S., Liu, J., & Feng, L. (2021). High-volume point cloud data simplification based on decomposed graph filtering. *Automation in Construction*, *129*. https://doi.org/10.1016/j.autcon.2021.103815

Xuan, W., Hua, X., Chen, X., Zou, J., & He, X. (2018). A New Progressive Simplification Method for Point Cloud Using Local Entropy of Normal Angle. *Journal of the Indian Society of Remote Sensing*, *46*(4). https://doi.org/10.1007/s12524-017-0730-6

Yang, Y., Li, M., & Ma, X. (2020). A Point Cloud Simplification Method Based on Modified Fuzzy C-Means Clustering Algorithm with Feature Information Reserved. *Mathematical Problems in Engineering*, *2020*. https://doi.org/10.1155/2020/5713137

Zhang, L., Luo, M., Liu, J., Li, Z., & Zheng, Q. (2020). Diverse fuzzy c-means for image clustering. *Pattern Recognition Letters*, *130*. https://doi.org/10.1016/j.patrec.2018.07.004