Received: January 22, 2022. Revised: February 24, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A New Clustering-Based Voxelization Scheme for Outliers Removing and Data Reduction of 3D Point Cloud

Sawsen Abdulhadi Mahmood[1]*

[1]*Computer Science Department, College of Education, Mustansiriyah University, Iraq*
*Corresponding author's Email: sawsenhadi@uomustansiriyah.edu.iq

Abstract: Point cloud voxelization methods are considered as significant topic in 3D computer vision field with variant relevant applications. The massive amount of point cloud data depicted a 3D model are extremely required considerable data storage along with processing time and ordinarily corrupted by outliers and noisy data. In this paper, we propose a new scheme for 3D point cloud clustering based on voxelization without pre-knowledge. Mathematical representation of 3D model is adopted in our framework in order to optimize the point cloud and remove the outliers with preserving the main characteristics of the whole 3D model shape. First, we segment the point cloud data points into uniform rectangular blocks (voxels) based on spatial 3D coordinates of vertices. Subsequence, each voxel is handled as cluster with gravity mass as voxel center and specifies the neighboring points to calculate cluster density. Then, the outliers at each voxel are detected based on geometry features of clusters and adjacency matrix representation. Finally, we reconstruct each voxel by removing the detected outliers. The proposed method is validated on different point cloud models with variant size. The experimental results highlighted that the new clustering-based voxelization method is suitable for real time applications due to its lower processing time. We have got a reduction rate close to 46.8% of the original 3D model size and precision percentage 96 %, which exhibits a higher noise removal rate.

Keywords: Voxelization, Point cloud, Outlier, Cluster, Gravity mass.

1. Introduction

3D model voxelization based 3D scenes simplification is essential issue in 3D computer vision. It's considered as knowledge of the environment and preparatory step for posterior functions such as robotics navigation, 3d registration and 3d visual matching [1, 2]. There is a considerable requirement for automated analysis, qualitative reconstruction methods and processing of 3D models to provide speed up for the existing frameworks and make them more efficient as well as generate higher-order geometric primitives [3, 4]. Recently, 3D point cloud images have attained growing attention as an efficient representation for patterns and objects. Therefore, it is decisive to eliminate the noisy data and outlier samples from point cloud models thereby retaining the properties, specifically, its fine details [5]. 3D point clouds raw data are commonly acquired by laser scanners which corrupted by noise and outliers. Outlier removing and data redundancy reduction in 3D point cloud models are extremely required in the frameworks of geometry processing. Its regularly includes; neglecting the outlier samples and de-noising the remaining vertices to retain the most informative regions in the scanned 3D surface. An efficient point clouds data reduction and outlier removal algorithm should have the ability for (i) keeping data precision by preserving serve edges and local details of the original 3D point cloud; ii) invariant property against rigid transformation regardless the angle of scanning or coordinate system [6]. The intrinsic challenge is to differentiate outlier or noisy data samples which have high frequency information, simultaneously prevent unnatural impacts through outlier removing process [7, 8].

In this paper, we aim to introduce a new clustering scheme based on voxelization (generating voxels grid from 3D point cloud), which is eligible for refining the surfaces and preserving the local details of 3D point cloud without affecting the properties of the original point cloud. Based on voxels grid representation of 3D models, we have developed outlier's removal approach to reduce the redundancy and unwanted data corrupted the original form. To this end, we split the 3D point cloud into voxels grid to simplify the detection of outliers for each voxel. Subsequence, two main geometric features are estimated from 3D point cloud data points to construct a weighted adjacency matrix for each voxel which contains a set of similar vertices. The proposed method is capable to detect flat, ceil and desktops regions using elevation angle property and geometry information of vertices, thereby significantly, we have obtained lower complexity and runtime.

The rest of this paper is structured as follows. Section 2 introduces an overview of filtering methods for 3D point cloud. The proposed method is given in section 3. The experimental results and analysis is demonstrated in section 4. The main conclusions obtained based on our experiments are given in section 5.

2. Related works

Point clouds simplification is considered as pre-process phase for computer vision systems. Based on standard errors metrics, the 3D point cloud models are specifically harmfully by diverse noises [9], which potentially lead to diminish the accuracy of 3D modeling. Thus, removing noisy data and outliers samples of 3D point cloud is the most critical step in 3D point cloud analysis before further implementations such as shape descriptor extraction, registration and 3D matching objects [5, 9-10]. The proposed work in [11] adopted the farthest point weighted mean down-sampling (FWD) approach to locate the feature missing due to filtering process and to keep feature samples while removing noisy data points. The authors employed 3d models in their experiments taken from Stanford University. The spatial construction of original point cloud is maintained in this method. Yao et al. [12] suggested a filtering approach using kernel density estimation (KDE), in addition to adopting threshold set depending on the maximum probability density of samples to reduce the run time of filtering process.

Quan et al. [13] proposed a filtering method based on the refinement of adjacent triangular point clouds in an organized triangular network to fruitfully reduce noisy points, in term of noise-free original model. However, several informative features in point cloud model will be removed in some scenarios; as consequence, manual limitations are desired to deny information missing of original 3D model. Leal et al. [14] introduced a filtering method based on median filtering and sparse regularization to preserve the sharp features and removing noise data as well as outlier from point clouds models. AC Carrilho et al. [15] proposed statistical outlier removal (SOR) filter which distinguishes outlier samples by computing the average distance between each point and its nearest K points. Shortly, these algorithms have realized specific results, however there is a need for enhancement with blurred images and removing Gaussian noise. X. Wang [16] proposed an adaptive ellipsoid searching filter based on radius filter. They aimed to detect the ellipsoid center. The outliers samples are specified based on the number of closely points in the ellipsoid. However, this algorithm is too difficult to implement in real-time application.

Point clouds segmentation methods are commonly based on features extraction of point cloud such as distance between vertices or point-to-point relations via neighbors' vertices in term of spherical neighborhood or voxels [17]. The proposed work in [3, 17] adopted Octree hierarchical segmenting to perform voxelization technique due to low memory usage and the indexing facilities. Bilateral filtering -based de-noising and outlier removing algorithms are widely used [11, 18-19] which can achieve efficient de-noising. However, these types of methods cannot fruitfully preserve the detailed features of large scale point cloud model. An optimized bilateral filter along with principle component analysis (PCA) method were adopted in the work [20] to remove the outliers from point cloud based on three vertex features; brightness, position and normal vector.

J. Papon et al. [21] introduced voxel cloud connectivity segmentation (VCCS) method to exploit full advantages of 3D geometry features of similar points regards to their normal vectors, RGB colors, and fast point feature histograms (FPFHs) descriptor to collect these points into a super voxel. De-noising methods based on deep neural networks have made great proceed; a batch normalization-based noise removal network was presented by [22]. This research has addressed the problem of covariant displacement and slight batches are resolved. However, their method is not appropriate for low-intensity and blurred images. The presented work in [23] utilized the previous of patch flow for

Received: January 22, 2022. Revised: February 24, 2022.

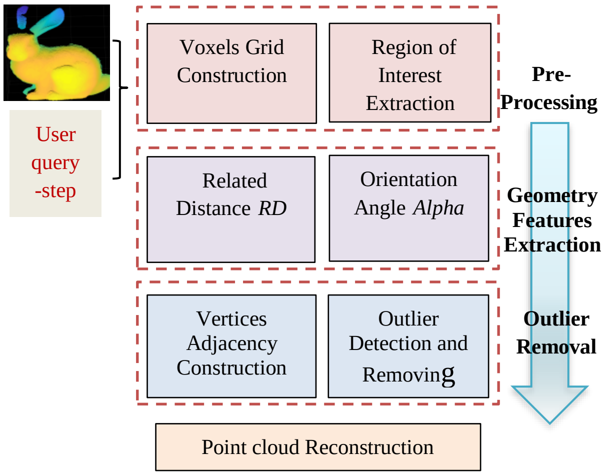


Figure. 1 The diagram of the proposed outlier removal scheme of point cloud model.

noise removing task, which conserves best edge features. However, this method works well when the input image is filtered using low-pass filter.

3. The proposed method

As s an essential step in segmenting 3D point cloud models, we ordinarily acquire a set of unorganized point samples corrupted by noise or outliers. In this context, the acquired 3D models can be expressed in Eq. (1):

$$PC' = \{p_i \mid p_i \in PC\}_{i=1\ldots s} \cup \{o_k \mid o_k \in O\}_{k=1\ldots r} \quad (1)$$

where PC, PC' are original point cloud and corrupted point cloud respectively, while O is set of outliers. The main objective of this paper is to obtain the original point cloud approximately by removing the set of outlier samples O. To this end, first; we construct a voxelization grid from the original PC based on spatial coordinates only. Second; the flat regions are detected and removed to extract the interested regions in PC. Third, the geometry properties of surface (Shape Descriptors) at each voxel are calculated to detect the outlier samples. Then, an outlier removal approach is implemented to remove outliers and noisy data samples. Finally, a reconstruction of the new point cloud is performed as a post processing step of the proposed framework. Fig. 1 illustrates the framework of the proposed outlier removal approach of 3D point cloud models.

3.1 Voxelization grid construction

The voxelization process of point cloud is a critical pre-processing stage for reducing the consuming time needed to extract significant features from raw data. Based on voxels-based 3D models representation, the 3D point clouds are segmented into several patches and the handling can be executed later in a patch-wise manner [24] which leads to trade-off between storage space and processing time. In this paper, 3D point cloud models are acquired by LiDAR scanners which composed of unarranged and massive amount of vertices. In order to construct the voxelization grid of a given point cloud, we have to partition the 3D space into set of equally-spaced (voxels) along three dimensions axis's with non-overlapping manner. For a given 3D point cloud model PC with s data points (vertices) represented as set of three dimensional points such that; $PC = \{p_1, p_2, \ldots . p_s\}$, where $p_i = (x_i,\ y_i,\ z_i)^T$ refers to p_i coordinates in $\mathbb{R}^3$. For the purpose of simplifying the 3D data points and estimating the geometry features, first; we have segmented PC model into uniform size of voxels using simple representation as shown in Fig. 2.

Along each axis coordinate direction, the higher and lower values are specified, then a cubic-bounding box is determined and aligned to surround the entire vertices belong to PC with dimensions d_1, d_2, d_3, where $d_1 = X_{max} - X_{min}$, $d_2 = Y_{max} - Y_{min}$ and $d_3 = Z_{max} - Z_{min}$.

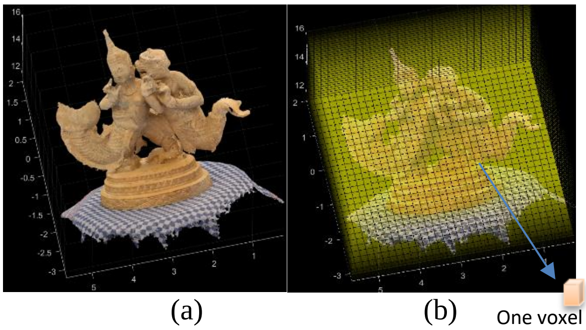
The voxelization scheme adopted in our framework has been designed to segment each direction by a user-query step S_t such that; the three dimensions (X_d, Y_d, Z_d) of voxel V_j are determined according to the following equations:

$$X_d(V_j) = \frac{d_1}{S_t},\ Y_d(V_j) = \frac{d_2}{S_t},\ Z_d(V_j) = \frac{d_3}{S_t}.$$

In this way, the vertices coordinates that lie within each voxel V_j are gathered based on voxel dimensions (X_d, Y_d, Z_d) and user-query step S_t as illustrated in algorithm (1). The output of algorithm (1) represented by set of M uniform rectangular (voxels) based on spatial segmentation scheme such that; $PC = \{V_1, V_2, \ldots . V_M\}$ and each voxel V_j composes of variant number of vertices.

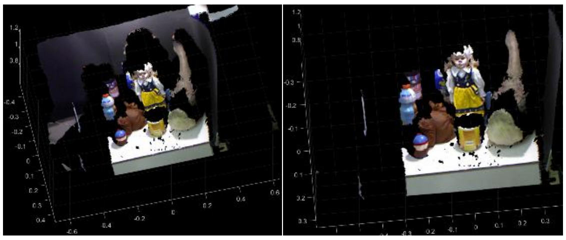
According to algorithm (1), region of interest *ROI* of voxel V_j is restricted by minimum and maximum spatial coordinates along each 3D- axis (X, Y, Z) and $|V_j|$ denotes the cardinality of voxel V_j. Subsequently, the voxelization grid is constituted based on the initial spatial coordinates of bounding-box with dimensions d_1, d_2 and d_3 which surrounds the entire point cloud PC and user-query step S_t to partition point cloud into smaller voxels as shown in Fig. 2.

Received: January 22, 2022. Revised: February 24, 2022.



(a) (b)

Figure. 2 Results of voxelization grid construction; a) original point cloud, b) voxelized point cloud



(a) (b)

Figure. 3 An illustration of plane regions extraction process; a) original point cloud, b) point cloud without plane regions

3.2 Plane regions elimination

This section describes the process of plane regions extraction from original point cloud in order to remove these regions and detect the most prominent objects in 3D model. In this way, the data points are reduced, and thus preserved for downstream processing. To this end, we have partitioned the point cloud into two distinct feature regions; plane regions and non-plane regions according to plane region detector. The elevation angle is used as benchmark to discard the plane regions. Based on our experiments, we found the values of elevation angle close to zero denote that the vertex is part of a flat region as shown in Fig. 3.

3.3 Geometry features extraction

In this section, we introduce an efficient representation of surface variation of *PC* model using geometry features. We aim to extract surface shape descriptors of each vertex $p_i \in V_j$ based on cubic volume definition and gravity of mass of voxel. Algorithm 2 describes the essential steps of extracting the geometry features required to implement the proposed approach of outlier removal and data redundancy reduction in 3D models. These steps can be clarified as follows:

a. Determine the gravity of mass $G_{Vj}(X, Y, Z)$ for each voxel V_j, where $j=1,\ldots.M$ based on the scheme used in [25].

b. Calculate the Euclidian distance $d(p_i)$ between each vertex p_i belongs to V_j and GV_{V_j}, according to Eq. (2) :

$$d_{pi}^{j} = \left\| p_i - GV_{V_j} \right\|_2, \ \forall\, p_i \in V_j \tag{2}$$

c. The local geometry information of each vertex p_i in voxel V_j concerning specifically to the geometry connection between vertex p_i and its neighbor vertices.

Accordingly, we have derived two local descriptors represented by related difference distance *RD* and orientation angle *Alpha* as follows: **First**; The related difference distance $RD(p_i)$ between each vertex $p_i \in V_j$ and GC_{Vj} is defined according to Eq. (3):

$$RD(p_i) = \frac{d(p_i)}{\frac{1}{n}\sum_{j=1}^{n} d(p_j)} \tag{3}$$

where n is voxel size (number of vertices). The higher values of *RD* refer to be p_i is considered as outlier point (approximately > 0.5). **Second;** For each vertex $p_i \in V_j$, we find the orientation angle

Algorithm 1: Voxelaization Grid Construction

Input: PC, S_t
Output: V //Set of voxels
Initialization: $V \leftarrow \varphi$
Step1:Estimate maximum and minimum coordinates along each direction (X, Y, Z) of point cloud
$[X_{max}, X_{min}, Y_{max}, Y_{min}, Z_{max}, Z_{min}] \leftarrow Mx\text{-}Mn(Pc)$
Step2: Specify Bounding Box Dimensions
$d_1 = X_{max} - X_{min}$, $d_2 = Y_{max} - Y_{min}$, $d_3 = Z_{max} - Z_{min}$
Step3:Specify Voxels Dimensions
$X_d = \frac{d_1}{S_t}$, $Y_d = \frac{d_2}{S_t}$, $Z_d = \frac{d_3}{S_t}$
$M = X_d \times Y_d \times Z_d$ *// M=No. of voxels in PC*
j=1
while $j \le M$ *do*
  $ROI_j = [X_{min}, X_{min} + X_d, Y_{min}, Y_{min} + Y_d, Z_{min}, Z_{min} + Z_d]$
  $V_j \leftarrow select(PC, ROI_j)$ *// select ROI vertices from PC*
  if $|V_j| = 0$ *then*
    Remove V_j *from list of voxels V*
    continue
  end if
  $X_{min} = X_{min} + X_d$
  $Y_{min} = Y_{min} + Y_d$ *//updating* X_{min}, Y_{min}, Z_{min} *coordinates*
  $Z_{min} = Z_{min} + Z_d$
  j=j+1
end while
Return V

 DOI: 10.22266/ijies2022.0630.11

Received: January 22, 2022. Revised: February 24, 2022.

$Alpha(p_i)$ which represents the angle between the normal vector of p_i and normal vector of GC_V according to the Eq. (4):

$$Alpha(p_i) = \operatorname{acos}\left(\frac{\langle N_p^i \ , N_{CV}^j \rangle}{\|N_p^i\| \ \|N_{CV}^j\|}\right) \tag{4}$$

where N_p^i, N_{CV}^j are the normal vectors of vertex p_i and GC_V of voxel V_j respectively. It's worth noting that the angle $Alpha$ represents the orientation (direction) of neighbor vertices with respect to GC_{Vj}.

3.4 Outliers removal

In order to achieve the outliers removal of a given 3D point cloud, we have adopted an improved approach to construct an adjacency matrix $AV_j(n \times n)$ for each voxel V_j in PC. To this end, the similarity degree between any two adjacent vertices in the voxel V_j is depicted in Eq. (5) as follows:

$$AV_{pi}^j = \begin{cases} 1 & if\ (RD(pi) \leq \ d_{th})\ and(-45^{\circ} \leq Alpha(p_i) \leq +45^{\circ}) \\ 0 & otherwise \end{cases} \tag{5}$$

based on our experiments, d_{th} is set to 0.5.

Accordingly, the outlier weight of vertex p is determined according to the distances between point p_i and its neighbors $RD(p_i)$ in addition to its orientation angle $Alpha(p_i)$. Using Eq. (5), we labeled the vertex by 1 whose its relative difference distance RD less than d_{th} and its orientation angle lying within the range $[-45^{\circ}, +45^{\circ}]$, otherwise the vertex is labeled by 0. Further, the vertex is considered as outlier data point when it's row and column elements in adjacency matrix are all equal to zero values. Thus, the higher divertive data points are excluded by this outlier removal technique as clarified in Eq. (6):

$$O = \{p_i \,\big|\, p_i \in V_j\,,\ AV_{i,k}^j = 0\ and\ i = k\} \tag{6}$$

where i=1….n, k=1….n, and O represents the set of outlier data points detected in specific voxel.

3.5 Point cloud reconstruction

In order to reconstruct each voxel $\widehat{V_j}$ based on its corresponding adjacency matrix AV_j in away such that:

Table 1. Data structure of voxel

Parameter	Description
V_j	Voxel with index j
n	Number of vertices in voxel
$GV_{X,Y,Z}$	Gravity of mass coordinates
$N[n]$	Normal vectors
$RD[n-1]$	Radial distance features
$Alpha(n)$	Orientation angle
$AV[n\times n]$	Adjacency matrix

Algorithm 2: **Geometry Features Extraction**

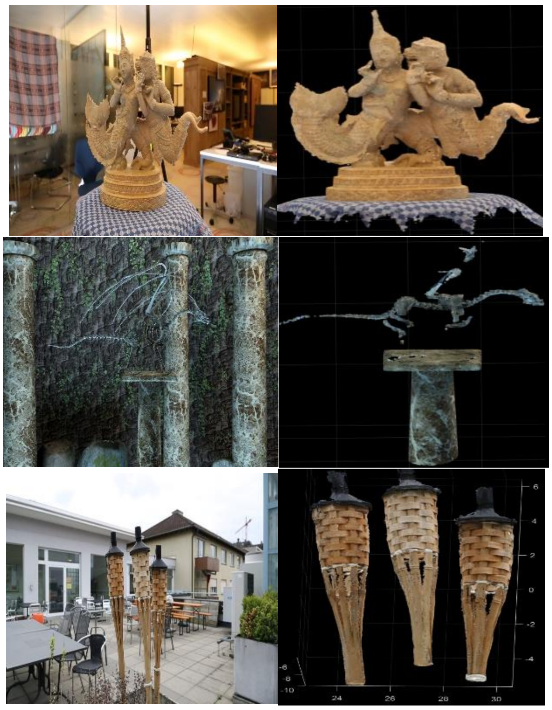
Input: V, d_{th} //V: set of voxels,
d_{th} *is distance threshold*
Output: $PC^{\sim}$ // reconstructed point cloud
Step1: Geometry Features Extraction- rotation angle
for each $V_j \in V$
GV_{Vj} = *Estimate-GV*(V_j) // *find* gravity of mass *using the formula used in*[25]
for each vertex $p_i \in$ **to** V_j
$d_{pi}^j = \left\| p_i - GV_{V_j} \right\|_2$ // *Calculate the Euclidian distance between* p_i *and* GV
N_p^i = *calculate normal vector of* p_i
N_{GV}^j =*calculate normal vector of GV*
$$Alph_{pi}^j = \arccos\left(\frac{\langle N_p^i \ , N_{GV}^j \rangle}{\|N_p^i\| \ \|N_{GV}^j\|}\right)$$
end for //i
end for // j
Step 2: Geometry Features Extraction- related distance
for each $V_j \in V$
n = sizeof (V_j)
for each vertex $p_i \in$ **to** V_j
$RD_{p_i}^j = \frac{d_{pi}^j}{\frac{1}{n}\sum_{j=1}^{n} d(p_j)}$ //*Calculate the related difference distance* $RD(p_i)$
end for //i
Step3: Outliers Removing
$AV_j^{n\times n}$ = *Adjacency*(V_j *,RD, Alpha,* d_{th} *, n*)
// Construct an adjacency matrix using Eq. (4)
Step4: Vertices labeling
$V_j \leftarrow$ *labeling*(V_j , $AV_j^{n\times n}$)
end for //j
Step5: Point Cloud Reconstruction
$PC^{\sim}$ = *New-Point Cloud*(*PC, V*) // *Re-construct point cloud from set of voxels*
Return $PC^{\sim}$

$$\widehat{V_j} = \{p_i \,\big|\, p_i \in V_j\ AND\ AV_{i,k}^j = 1\}, \tag{7}$$

where i=1….n and k=1….n. In this context, the vertices that lie far away from the gravity of mass of each voxel as well as its normal vectors directions differ from normal vector of *GV* are ignored and preserve the vertices that spatially closest to *GV*. We handle outlier removing as a local issue; the decision about specific point $p \in PC$ only based on the neighborhood points of p within its voxel. Concentrating on points neighbors within its voxel allow us to treat dense point clouds without missing local detail. In this paper, we have predefined a threshold value d_{th} to select the closest vertices to *GV*. The data structure of each voxel *V* is clarified in Table 1.

4. Experimental results and analysis

In order to evidence the feasibility of the proposed clustering method of outlier removal and data reduction in point cloud models based on voxelization approach, comprehensive experiments were implemented using scanned point cloud models which comprised various types of outliers. To investigate the performance of the proposed outlier removal model, we use set of 3D point cloud samples paired noisy samples and coinciding ground truth samples (cleand). The 3D models utilized in our experiments involved 3D point cloud models taken from [26] including dragon, torch, scarecrow and STATUE, models, stanford 3D scanning repository from (http://graphics.stanford.edu/data/3Dscanrep) and Kinect Views from (http://vision.deis.unibo.it/research/78-cvlab/80-shot) including frog and duck models. The Stanford 3D Scanning gallery consists of four models: Armadillo, Happy Buddha, Bunny and Dragon scanned by a Cyberware 3030 Scanner. Fig. 4 describes samples of 3d models and their two dimension images. Our method is implemented using MATLAB 2019 and run on a desktop PC with (quad core, 3.4 GHz).



(a) (b)

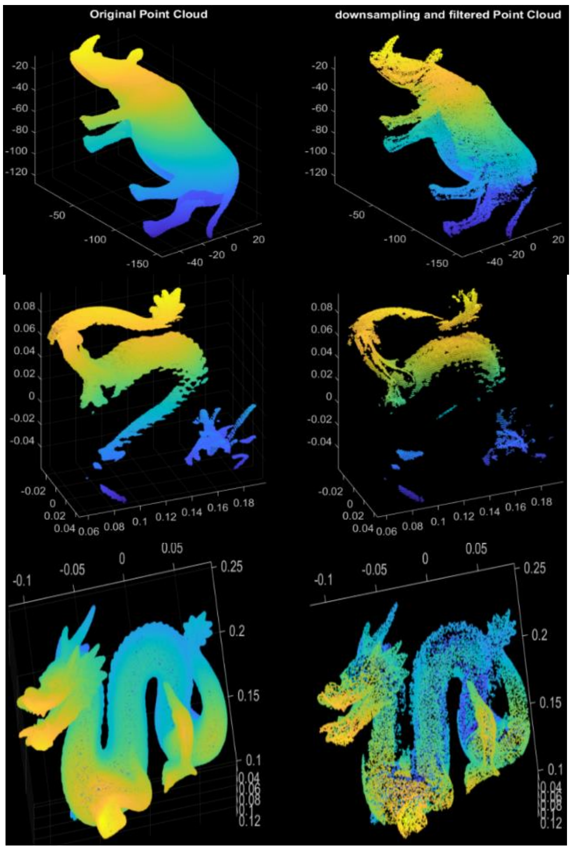
Figure. 4 samples of 3d point cloud models taken from Statue, dragon: (a) 2d image and (b) 3d model.

4.1 Point cloud reconstruction

The performance evaluation of data reduction approach is achieved using 3d point cloud models with massive data point. The experiments results proved the reduction rate reach approximately to 46.8 % of the original point clouds as shown in Fig. 5. Table 2 stats the size of six original models, the reduced size of 3d model after applying the data reduction approach and the reduction rate percentage.

4.2 Outliers removal results

The experimental results of the proposed outlier removal method are demonstrated upon several 3d point cloud models taken from public databases



(a) (b)

Figure. 5 An illustration of data reduction approach: (a) original point cloud and (b) reduced point cloud

Received: January 22, 2022. Revised: February 24, 2022.

Table 2. Results of data reduction approach

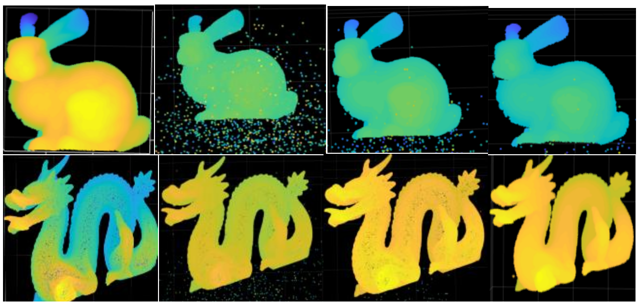
3D model	***# points of original* Pc**	***# points of downsampling* Pc**	***Reduction Rate %***
Rhino	31822	14479	45.4
Statue	2245684	1090125	48.5
Armadillo	172974	81407	47
buny	40256	16347	40.6
cheff	30391	13577	44.7
Dragon	100207	54906	54.7

mentioned in the previous section as shown in Fig. 6. The first column represents the original 3d model; the second column presents the unfiltered 3d models, the third column illustrated the filtered 3d model using traditional method, the last column demonstrates the 3d model after implementing our method for outlier removing. In order to evaluate the performance of the proposed outlier removal method, we have adopted precision benchmark identified as; *Precision* = N_q/N_s , where N_q represents the detected noisy data points through implementation, N_s referred to the actual number of noisy data points corrupted the point cloud. In all experiments, we have got a precision percentage close to 96%, which exhibits a higher noise removal rate. In other hand, distance measure *D1* was used in our experiments to estimate the difference between the detected outliers set points and ground truth outlier set points as depicted in Eq. (8):

$$DI = \left\| O_i - O_i^G \right\|_1 \tag{8}$$

Where O_i represents the detected outlier points and O_i^G is set of ground truth outlier points that added to the original point cloud.

The idle implementation of this equation is considered a zero value of *D1*. The experiments of outliers removal method involves adding Gaussian noise to the point cloud models with mean and standard deviation values related to data size of the input model, i.e. point cloud scale. This technique assisted in being more consistency of spatial coordinates for both original data points and noisy data points. Throughout experiments implementation, we have set σ to 5 % of the point cloud scale. Fig. 6 shows the results of two point clouds samples corrupted by Gaussian noise as shown in Fig. 6-b, then we have applied the traditional noise removal technique (based on threshold) as illustrated in Fig. 6-c. Obviously, the proposed method of outlier removing has higher performance than the traditional threshold outlier method as shown in Fig. 6-d. The Quantitative results of noise and outlier removing approach are illustrated in Table 3.



(a) (b) (c) (d)

Figure. 6 An illustration of outlier removal point cloud: (a) original point cloud, (b) noisy point cloud, (c) filtered point cloud based on threshold technique, and (d) filtered point cloud based on the proposed method

Table 3. Quantitative results of outlier removal approach

3D model	**# points of original Pc**	**# points of Noisy points**	**# points after noise removal**	**DI**	**T (s)**
Rhino	31822	1591	30962	0.54	0.25
Statue	2245684	112284	2282428	0.32	4.14
Arma-dillo	172974	8649	174551	0.18	0.46
buny	40256	2013	40344	0.04	0.27
cheff	30391	1529	30190	0.03	0.19

Table 4. Comparison study based on PSNR metric

Method	[22]	[23]	[11]	The proposed
	PSNR			
std=1.5	31.10	33.15	35.28	35.38
std=2.5	29	30.8	32.45	33

Table 5. Comparison study based on precision and running time metrics

Method	Precision (%)	Running time (s)
Threshold-denoising method	32	759
[11]	96	72
[20]	92	189
The proposed	96	5

4.3 Comparison study

To predicate the obtained results in term of quantitative evaluation, we have quantified the bias of the filtered model from ground truth model. To this end, Peek Signal to Noise ratio (PSNR) metric was utilized to demonstrate the effectiveness of the proposed method for outlier removal purpose. Peek signal to noise ratio (PSNR) metric is adopted in our experiments due to its effects for evaluating the outlier removal method and to clarify the difference between the corrupted point cloud quality and the output point cloud after applying the outlier removal method. In the experiments, two distinct coefficients regard Gaussian noise (std=1.5 and std=2.5) were utilized. We have accomplished the comparison study using 3d models taken from Stanford University data sets. Meanwhile, the proposed outlier removing method has been compared with three denoising schemes, including [11, 22, 23] as clarified in Table 4.

In order to have a fully depiction the influence of the proposed outlier removal method, another comparison study was performed based on precision and average running time (in seconds) metrics. Three methods including; threshold-denoising, [11] and [20] have been compared against the proposed approach which exhibited an efficient result regards the running time and precision benchmarks. The obtained results of this comparison have been demonstrated in Table 5.

Obviously, there is a considerable drop in processing time when we used the proposed method for outlier removing comparing to the other methods in addition to have a higher precision rate close to 96 %.

5. Conclusions

The essential advantages of adopting voxelization representation for 3D models are that (i) the output voxels are related by ordinary hierarchical structure with regular spatial size of all voxels; (ii) analysis simplification of 3d models; (iii) specifying a considerable trade-off between the 3d shape resolution and the storage space. The main contributions of the proposed outlier removal approach based on voxelization technique are; lower consuming time of processing as well higher reduction rate with preserving the objects shapes. In the experiments, a given point cloud is reconstructed (down-sampled) into another point cloud formulated by the super-points or super-boxes and the size of the entire point cloud is reduced with 46.8 % reduction rate of the original point cloud for processing simplification purpose. Voxelization-based 3D representation can be exploited in the perception module of autonomous driving, in addition to get a lower storage space of 3D models with preserving the objects shapes. Based on our experiments, using adjacency matrix in decision making about outliers detection task leads to exhibit the spatial relevance among 3D points effectively. Furthermore, accelerating the decision making of being specific point is outlier or not, reducing the computation processes or run time. Discarding ground and plan regions from point cloud model contributes to reduce the size of point cloud in addition to minimize the execution time. Compared to other methods, we have got a precision benchmark 96 % to detect and remove outliers with average processing time reach to 5 seconds.

Conflicts of interest

The author declares no conflict of interest.

Acknowledgments

This research was confirmed and supported by computer science department at college of education-Mustansiriyah University.

References

[1] M. Aleksandrov, S. Zlatanova, and D. J. Heslop, “Voxelisation Algorithms and Data Structures: A Review”, *Sensors*, Vol. 21, No. 24, pp. 8241-8262, 2021.

[2] M. Huang, P. Wei, and X. Liu, “An Efficient Encoding Voxel-Based Segmentation (EVBS) Algorithm Based on Fast Adjacent Voxel Search for Point Cloud Plane Segmentation”, *Remote Sensing*, Vol. 11, No. 23, pp. 2727-2747, 2019.

[3] F. Poux and R. Billen, “Voxel-based 3D Point Cloud Semantic Segmentation: Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods”, *International Journal of Geo-Information*, Vol. 8, No. 5, pp. 213-245, 2019.

[4] X. F. Han, J. S. Jin, M. J. Wang, W. Jiang, L. Gao, and L. Xiao, “A review of algorithms for filtering the 3D point cloud”, *Signal Processing Image Communication*, Vol. 57, pp. 103–112, 2017.

[5] S. A. Mahmood and F. S. Mohamed, “Surface Patch Detection of 3D Point Cloud Using Local Shape Descriptor”, In: *Proc. of 2019 First International Conf. of Computer and Applied Sciences*, Baghdad, Iraq, pp. 163-168, 2019.

[6] H. K. Abbas1 and A. H. A. Saleh, A. A. A. Zuky, “Optical Images Fusion Based on Linear Interpolation Methods”, *Iraqi Journal of Science*, Vol. 60, No.4, pp. 924-936, 2019.

[7] H. K. Abbas, F. Faris, S. Sami, and A. Fadel,

"Adopting Image Integration Techniques to Simulate Satellite Images", *Iraqi Journal of Science*, Vol. 61, No. 12, pp. 3446-3456, 2020.

[8] C. Hu, Z. Pan, and P. Li, "A 3D Point Cloud Filtering Method for Leaves Based on Manifold Distance and Normal Estimation", *Remote Sensing*, Vol. 11, No. 2, pp. 198-216, 2019.

[9] M. Rakotosaona, V. L. Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds", *Computer Graphics Forum*, Vo. 39, No. 1, pp. 185-203, 2020.

[10] N. Zhu, Y. Jiaa, and L. Luo, "Tunnel Point Cloud Filtering Method Based on Elliptic Cylindrical Model", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLI-B1, pp. 735-740, 2016.

[11] B. Zou, H. Qiu, and Y. Lu, "Point Cloud Reduction and Denoising Based on Optimized Downsampling and Bilateral Filtering", *IEEE Access*, Vol. 8, pp. 136316-136326, 2020.

[12] X. G. Tian, L. J. Xu, L. X. Li, T. Xu, and J. N. Yao, "Filtering of Airborne Lidar Point Cloud with a Method Based on Kernel Density Estimation (Kde)", *Lasers in Engineering*, Vol. 34, No. 4, pp.221–237, 2016.

[13] G. Y. Quan, J. Song, X. Guo, Q. Miao, and Y. Yang, "Filtering LiDAR data based on adjacent triangle of triangulated irregular network", *Multimedia Tools and Applications*, Vol. 76, No. 8, pp. 11051–11063, 2016.

[14] D. L. Esmeide, G. S. Torres, and W. John, "Sparse Regularization-Based Approach for Point Cloud Denoising and Sharp Features Enhancement", *Sensors*, Vol. 20, No. 11, pp. 3206-3223, 2020.

[15] D. A.C. Carrilho, M. Galo, and R. C. Santos, "Statistical outlier detection method for airborne LiDAR data", In: *Proc. of the ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Göttingen, Germany, Vol. XLII-1, pp. 87–92, 2018.

[16] X. Wang, C. Glennie, and Z. Pan, "An Adaptive Ellipsoid Searching Filter for Airborne Single-Photon Lidar", *IEEE Geoscience and Remote Sensing Letters*, Vol. 14, No. 8, pp. 1258-1262, 2017.

[17] D. Y. T. Su, J. Bethel, and S. Hu, "Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications", *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 113, pp. 59–74, 2016.

[18] D. H. Yuan, J. K. Pang, and J. W. Mo, "Denoising algorithm for bilateral filtered point cloud based on noise classification", *Journal of Computer Applications*, Vol. 25, No. 8, pp. 2305-2310, 2015.

[19] D. C. Lv and M. Li, "Point Cloud Denoising Algorithm Based on Noise Classification", In: *Proc. of International Conf. on Culture-oriented Science & Technology*, pp. 123-127, 2020.

[20] Zhang, C. Zhang, H. Yang, and L. Zhao, "Point cloud denoising with principal component analysis and a novel bilateral filter", *Traitement Du Signal*, Vol. 36, No. 5, pp. 393–398, 2019.

[21] D. J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds", In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2027-2034, 2013.

[22] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization", *Neural Networks*, Vol. 121, pp. 461–473, 2020.

[23] J. Zeng, G. Cheung, M. Ng, J. Pang, and C. Yang, "3D Point Cloud Denoising Using Graph Laplacian Regularization of a Low Dimensional Manifold Model", *IEEE Transactions on Image Processing*, Vol. 29, pp. 3474-3489, 2020.

[24] D. Kong, Z. Wang, X. Jin, X. Wang, T. Su, and J. Wang, "Semi-Supervised Segmentation Framework Based on Spot-Divergence Supervoxelization of Multi-Sensor Fusion Data for Autonomous Forest Machine Applications", *Sensors*, Vol. 18, No. 9, pp. 3061-3085, 2018.

[25] X. Zhan, Y. Cai, H. Li, Y. Li, and P. He, "A point cloud registration algorithm based on normal vector and particle swarm optimization", *Measurement and Control*. Vol. 53, No. 3, pp. 265-275, 2020.

[26] K. Wolff, C. Kim, H. Zimmer, C. Schroers, M. Botsch, O. S. Hornung, and A. S. Hornung, "Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction", In: *Proc. of Fourth International Conf. on 3D Vision*, pp. 118-127, 2016.