
UNIVERSITY OF CAPE TOWN


DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project Final Paper 2022

Title: Machine learning for semantic segmentation of Cultural Heritage Sites

Author: Jemma Sundelson

Project Abbreviation: CH segmentation

Supervisor(s): Patrick Marais and Luc Hayward

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Machine Learning For Semantic Segmentation Of Cultural Heritage Sites

Jemma Sundelson
University of Cape Town
Cape Town, South Africa
sndjem001@myuct.ac.za

Abstract

Digital technologies such as laser scanners are popular techniques used to scan a site and create a 3D model. The recordings of these scans are called point clouds; however, often they are imperfect and contain noise which must be removed (scaffolding, vegetation, foliage, people, animals). Removing these unwanted points is called point cloud cleaning, a mainly manual and time-consuming task due to the millions of point data in a single scan. Removing these points requires a classification task in which points in the point cloud are labelled as noise or some semantically meaningful class. After classification, one can construct accurate 3d models for several tasks, such as cultural preservation. This paper explores machine learning methods, namely clustering and classification, to automate this mainly manual process of point cloud cleaning. The unsupervised approach investigated includes using two partitional-based methods: K-medians and Fuzzy c-means, and the Gaussian mixture model to assist the semantic segmentation of 3D point clouds. The K-means clustering algorithm is used as a baseline for comparison with the aforementioned. Additionally, to achieve the most accurate results, we explore feature clustering through CloudCompare and PointNet++. From there, the classification task is completed using the majority ground truth label of points per cluster so binary classification can occur. Results show all algorithms are capable of producing semantically meaningful clusters on the raw dataset and PointNet++ dataset. K-means produces optimal classification results on a considerable input cluster amount; however the clusters are not useful to a human. Gaussian mixture model maintains a strong performance in producing semantically meaningful clusters and robust classification results throughout all experiments. It's performance is optimised when using PointNet++.

CCS Concepts

• **Computing methodologies** → **Machine learning approaches**; **Feature selection**; **Cluster analysis**; **Classification**.

Keywords

Point Clouds, Machine Learning, Classification, Clustering, Semantic Segmentation, Feature Extraction

1 Introduction

To learn and explore the life of humans before us, cultural heritage (CH) sites must be preserved. A popular preservation tactic uses digital technologies such as terrestrial laser scanners to scan the CH site and construct applicable 3D models. These models allow preservationists to determine where the site is spoilt for the development of appropriate conservation strategies. The recordings of the scans are stored as point clouds. Point clouds consist of points defined by x,y,z coordinates. However, the point clouds often contain noise and unwanted points from the structure, such as trees, people, animals and vegetation, making an accurate 3D model difficult to obtain [25]. Point cloud cleaning has been explored to discard unwanted points, but this process is manual, labour-intensive and time-consuming even for a highly skilled individual. Additionally, the structures of CH sites vary, making it tough to formulate a general cleaning model. Therefore, finding an accurate and automatic solution presents a unique challenge. One such solution is utilising an unsupervised machine learning method, such as clustering. Clustering is a powerful tool that has been applied to a variety of complex domains, therefore, making it suitable to address the problem at hand. The aforementioned method uses unlabelled points in a point cloud, to accurately decompose these point clouds into semantically meaningful clusters. Semantically meaningful clusters form all or part of some structure in a point cloud that a human can identify, for example, a wall, tree, or person. Therefore even if there are multiple clusters for an object, undesired structures can still be easily selected and removed compared to removing points one by one.

Therefore, this paper aims to explore how precisely unsupervised methods perform in this context. Specifically, we explore the application of several clustering algorithms to create 3D models with only desirable geometric features. This will be implemented by analysing and comparing the performance of two partitional-based clustering algorithms, namely k-medians and fuzzy c-means (FCM) and the Gaussian mixture model (GMM). The clustering algorithms' performance will be determined by internal cluster validity metrics such as the silhouette coefficient and Davies Bouldin index. These metrics will assess how compact, cohesive, and well-separated the produced clusters are. K-means will be used as a baseline in which our results will be compared. Therefore, we propose our first research questions as *"does K-medians GMM and FCM produce clusters on point clouds of CH sites that are more semantically meaningful than those produced by k-means algorithm, using cluster validity metrics of Silhouette Coefficient and DB Index?"*.

In addition, clustering will not only be performed on the raw dataset (x,y,z). Feature extraction will be implemented on the point cloud to form two rich feature sets. CloudCompare, a 3D point cloud processing software, is used to extract several human-interpretable features. We also utilise PointNet++, a neural network structure for processing point clouds and learning feature sets[30] to compute a point cloud with around 1000 highly abstract features. These rich datasets will be fed into the several clustering algorithms increasing the chances of more accurate results.

Moreover, a further interest of this paper concerns classification. The classification step in this project assigns points in the dataset to different classes with labels as a means to implement binary classification. This will show whether the clusters created are part of the CH site, labelled 'keep' or noise, and labelled 'discard'. Therefore, the second research question we aim to answer is, *"what is the upper bound accuracy when using the resulting clusters to perform binary classification using' keep' and' discard' labels on a point cloud?"*

F1 score, Recall, precision, and Intersection over union are used to evaluate the classification task. This project will materialise through a five-stage pipeline, which includes data acquisition, feature extraction, clustering, visualisation and analysis and evaluation.

2 Background and Related Work

3D point cloud models generated from laser scans are an important means to record and document Cultural heritage sites. Segmentation strategies have been explored for several decades, and various approaches have been implemented to explore 3D point cloud classification to class points in a scan as relevant or irrelevant. Marais et al. [25] introduce a binary classification scheme that classifies points as 'keep' if they are relevant and part of the CH site, else 'discard' for noise and outliers. The semi-automated approach trains a random forest classifier on the original 'keep'/'discard' labels, achieving a 98 % accuracy. We follow their binary classification strategy in our implementation. Belton et al. [5] tackles point cloud classification to decompose tree point clouds into various components (leaves, branches, trunk). The 3D tree models are acquired from terrestrial laser scans. The Gaussian mixture model is utilised to cluster and classify points into the different tree components to separate the leaves from the rest of the tree structure. Gaussian mixture model (GMM) is a probabilistic model that assigns different degrees of belonging to every point in the point cloud. Similarly, Zhenyang et al. [19] implement a Gaussian mixture model to improve the accuracy of individual tree extraction from a forest landscape. The proposed method achieved above 87 % correctness in classification.

Moreover, Shan et al. [32] investigate k-means clustering and feature selection for building extraction from LiDAR data. K-means successfully decomposes noisy 3D point clouds into semantically meaningful clusters representing specific roof segments. This result is based on reliable feature extraction, which is shown to be a crucial step in the process. Likewise, Albano et al. [3] explores automatic segmentation for 3D building detection and modelling using fuzzy c-means clustering. Fuzzy c-means achieved robust geometry accuracy (mean, standard deviation, Root Mean Square Error of the Euclidean distance) and time computation. They conclude that the fuzzy c-means algorithm has the potential to perform successfully in other contexts. Therefore, this paper will apply this algorithm to the context of cultural Heritage sites. Finally, Cardot et al. [8] compare k-medians and k-means by determining television audience profiles. The results of the L1 error and classification error rate exhibited that k-medians outperformed k-means as the presence of outliers had an adverse effect on the k-means algorithm; however, the k-medians algorithm was unaffected. Additionally, k-medians proved successful in clustering large datasets due to its low computation time. Moreover, Mahdaoui et al. [24] compare k-means to Fuzzy c-means. Experimental analysis is executed to find an optimal clustering method to simplify 3D point clouds. K-means outperformed fuzzy c-means in terms of accuracy; however, fuzzy c-means makes simplification faster due to its lower computational time.

Furthermore, feature extraction plays a major role in assisting various clustering methods in producing semantically meaningful groupings. Features are descriptors of a point, for example, density, curvature, roughness, and covariance. Grilli et al. [17] investigate machine learning segmentation on a point cloud with additional descriptors added to each point. He explains that the Canupo segmentation algorithm, which is implemented in a 3D point cloud processing software called CloudCompare, was used to form meaningful groupings on a cultural heritage site in Naples, Italy. The segmentation algorithm manages to separate vegetation from stones successfully. Similarly, Farella et al [14, 17] found success in using the same classification algorithm to distinguish natural components from artificial structures in point clouds of military forts during the First World War. We implement feature extraction in our research to maximise robust results.

Although the above shows that there has been a success in decomposing 3D point clouds into semantically meaningful clusters, there are still challenges that are faced in the process. According to Nguyen et al. [27], segmentation can be complex due to the uneven sampling density, unclear structure of point cloud data and high repetition of points in a sample. Additionally, the points recorded from the laser scans can be noisy, making it hard to differentiate between the desired model and the background noise. Machine learning techniques are becoming more popular, and evidence shows that they are outperforming methods based on geometric reasoning, such as region growing and model fitting techniques. However, the machine learning methods can be prolonged, making such large point clouds challenging to work with. Additionally, they rely heavily on the success of feature extraction, which, unfortunately, is also a complicated task.

3 Design and Implementation

The design of our approach is split into a five-stage pipeline.

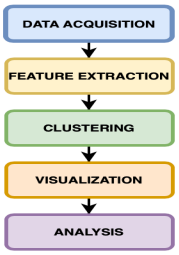

Figure 1: The five-stage implementation pipeline

3.1 Data Acquisition

Data acquisition comprises the first stage of our pipeline. The dataset used throughout the project is 3D point clouds from TLS campaigns conducted by the VCG (visual computing group) CNR-ISTI in Italy.

3.2 Feature Extraction

The second step in our pipeline is feature extraction. Features are descriptors of a point; therefore, all points lying on the same surface will have similar features creating a denser region in the feature space [6]. This allows more information to be encoded into the dataset, giving the clustering algorithms a better chance of forming meaningful clusters.

3.2.1 ***CloudCompare*** CloudCompare is utilised to retain several human comprehensible features. CloudCompare is an open-source 3D visualisation and computation software which has gained popularity over recent years due to its user-friendliness and increasing feature set [11]. Due to time constraints, feature combinations were not experimented with to evaluate which combination gives the best results. Instead, the features were manually chosen by following the point cloud classification strategy of Marais et al. [25] and Weinmann [33].

Name	Notes
Anisotropy Planarity Sphericity Linearity Omnivariance Eigen entropy Surface variation Verticality	features computed from eigenvalues/ vectors of the structure tensor
Volume density Surface density	For k-nearest neighbours
Gaussian curvature Mean curvature Normal change rate	principal curvatures at **p**

Figure 2: Selected CloudCompare features adapted from [25]

Covariance features were chosen as they describe the properties of the church surface. Linearity, planarity and sphericity, in specific, provide information about the dimensionality of the church structure. Features describing the curvature of the surface were additionally picked, namely Gaussian and mean curvature, as well as normal change rate. Geometric features such as surface and volume density were selected as they describe the point position and measurement in space [5]. As more features are added, there is a trade-off between accuracy and complexity as testing time increases. Therefore not every CloudCompare feature was chosen.

3.2.2 ***PointNet++*** Additionally, to take the dataset into an even higher dimensional space, we utilise PointNet++.

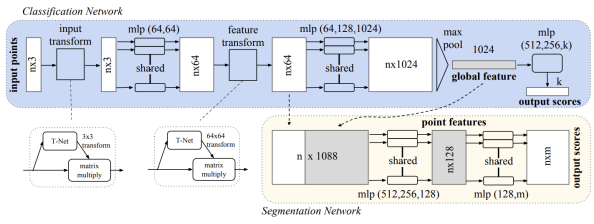

Figure 3: PointNet architecture [30]

PointNet is a pioneering deep neural network that directly takes in a 3D point cloud and provides a unified approach for tasks such as classification and segmentation. The architecture depicted in Figure 3, can learn numerous feature types, including global and local point features, providing a practical and uncomplicated method for various 3D recognition applications [30]. PointNet++ is an extension of PointNet as it has an additional hierarchical design that captures the local structure created by the metric that space points live in. This adaptation is critical for the success of convolutional architectures as it can reliably learn features even in irregular environments [31].

The Yanx Pointnet_Pointnet2_pytorch2 GitHub repository was used for the implementation. Multiple lines of code were discarded and adapted to suit our dataset. PointNet++ generated a dataset with 128 abstract features added to each point. Due to the point cloud being so large, voxel downsampling was implemented using Open3D. voxel downsampling makes working with such a significant point cloud feasible as it uses a regular voxel grid to create a uniformly downsampled point cloud from an input point cloud [34].

3.3 Clustering

After the raw dataset as well as the two feature sets, have been prepared, clustering occurs on the unlabelled data, forming the third step in our pipeline. The goal of clustering is to find a structure in a collection of unlabelled data [23]. In this case, it is used to separate the noise from the CH model by forming semantically meaningful groupings that contain points with similarities called clusters. Ideally, these clusters will correspond to a collection of points that make sense to humans, such as all the points that comprise the scaffolding in the three-dimensional church model. Therefore, the taxing task of point cloud cleaning can be significantly simplified if the aforementioned is achieved. The clustering techniques explored are partitional algorithms and a Gaussian mixture model. K-means is the baseline clustering algorithm in which the three aforementioned algorithms can be compared. Partitional algorithms assign a centroid to each group and then iteratively allocate the closest points to that centroid to form a cluster [1]. These algorithms require the user to select the input amount of clusters.

3.3.1 ***K-means*** Recall that k-means will be used as a baseline. K-means was chosen due to its simplicity, wide use and low time and space complexity. The python library Scikit-learn [1] is used to implement the algorithm on the point cloud data.

3.3.2 ***K-medians*** K-medians is an adaptation of the popular k-means algorithm. The difference being, the centroids in the k-medians algorithm are approximated by the median[21] instead of the mean value. K-medians overcome the sensitivity to outliers that k-means concerns. This is alleviated by calculating the Manhattan distance instead of the euclidean distance. K-medians is implemented using the python library pyclustering. [2]

3.3.3 ***Fuzzy c-means*** Similarly, fuzzy c-means is also an adaptation of the k-means algorithm; however, fuzzy c-means is a soft clustering algorithm, not a typical hard clustering algorithm. Each point in a cluster is assigned a probability score to be a part of that

[1] Scikit learn is a valuable and robust library for machine learning in Python
[2] pyclustering is a Python, C++ data mining library

cluster. All points are related to all clusters; however, they have different degrees of belonging [16]. Fuzzy-c-means is a Python module available on PyPI [3], implementing the Fuzzy C-means clustering algorithm.

3.3.4 ***Gaussian mixture model*** A Gaussian mixture model is also soft clustering algorithm that assigns various probabilities to points, making the algorithm more flexible. Every component (Gaussian density) has its own mean, as well as covariance [18]. Clusters are formed due to different Gaussian distributions. The Gaussian mixture model is implemented using the Scikit-learn library.

3.4 Visualization

The fourth step of the pipeline includes visualisation. Once we have our clustering results, two different platforms are used to visualise the clustered point clouds. The first is open3D, which is a rapid development software that deals with processing point clouds [34]. The second visualisation tool is a point cloud processing tool kit (pptk), a python package used for visualising and processing 3D point clouds [29].

3.5 Evaluation

Once all our data has been processed, clustered and visualised, analysis and evaluation must occur. Evaluation metrics are explored to assess whether or not the results of the clustering algorithms were a success or failure in producing semantically meaningful clusters. The primary aim is to produce clusters where intra-cluster similarity is maximised whilst inter-cluster similarity is minimised. Therefore, internal validation methods [28], such as the silhouette score, Davies Bouldin index and Bayesian information criterion, are utilised to evaluate the quality and performance of the clusters produced. Additionally, as mentioned, the classification task is of further interest and will be completed after the clustering process is achieved. This step aims to use the additional ground truth label in the dataset to classify the clusters into 'keep' and 'discard'. To evaluate the classification task, metrics such as F1 score, precision, Intersection over union and recall are implemented. They describe the algorithm's ability to label points in a point cloud as part of the CH site (keep) or irrelevant and noise (discard).

3.5.1 ***Silhouette Score*** The silhouette score is a standard metric used to validate clustering algorithms as it combines cohesion and separation into one single metric. Cohesion is measured between points in a cluster, while separation is measured between points in different clusters [28]. The above are both based on a distance function which determines the similarity of points in the dataset. The silhouette score is normalised therefore lying in the range [-1,1]. A score close to 1 means the algorithm has produced clusters of high quality, whilst a score near -1 means the clusters are not well separated.

$$s = \frac{b - a}{max(a, b)}$$

Silhouette Coefficient Formula

[3]Python Package Index (PyPI) is a repository of software for the Python programming language.

3.5.2 ***Davies–Bouldin Index*** The Davies-Bouldin Index evaluates cluster quality by dividing the intra-cluster distance by the inter-cluster distance. A minimised DB Index means that the distance between points within a cluster is low, and the distance between points in separate clusters is high. This implies that the algorithm has computed dense clusters that are well separated.

3.5.3 ***Bayesian information criterion*** The Bayesian information criterion (BIC) is an additional model selection criterion penalised by the model's complexity [9] to avoid overfitting. Model selection concerns the number of components in the model and the covariance type. In this paper, the BIC score is only used on the Gaussian mixture model. The implementation strategy followed the Scikit-learn model selection criterion as a guideline.

3.5.4 ***Precision*** To explain precision, a confusion matrix is utilised to depict the clustering algorithms' four possible classification performance results. These results include 'True Positive' (TP), 'True Negative' (TN), 'False Positive' (FP), and 'False Negative' (FN). A true positive is when the sample is positive and, therefore, correctly classified as positive. A false positive is when the sample is incorrectly classified as positive and is actually negative. True negative and false negative follow the same approach. Therefore, precision is a metric that divides the true positive by predicted positives [15]. Precision is a valuable metric as it depicts how accurately the algorithm predicts true positives.

$$precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

3.5.5 ***Recall*** As mentioned above, the confusion matrix is additionally utilised for the recall metric. Recall is calculated by the division of true positives over anything that should have been predicted as positive.

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

3.5.6 ***F1 score*** The F1 score measures the classification models' performance by combining precision and recall into one weighted index. The F1 score is maximised when both the model's precision and recall are high.

3.5.7 ***Intersection over Union*** The Intersection Over Union (IOU) score quantifies how well the model distinguishes between noise points and points from the cultural heritage site.

4 Experimental Setup

The following experiments are executed on 2,4 GHz Quad-Core Intel Core i5 and Google Colaboratory [1]. The code is written in python3. We received a npy file [2] and a ply file [3] that consist of the same point cloud with approximately 21 million points. Each point (x,y,z) has an intensity value and a ground truth label attached. To utilise the scans correctly, each dataset is pre-processed differently.

[1]Google Colaboratory allows python code to be executed in a browser providing free access to computing resources, including GPUs.
[2]an npy file is a standard binary file format
[3]a ply file stores 3D data from 3D scanners

- **Raw data**: The npy file is loaded into python, and the point cloud is stored as a grid of binary values called a NumPy array.
- **CloudCompare**: To obtain selected features, the ply file is placed into CloudCompare and then exported as a las file. [4] The las file is then read into the python code.
- **PointNet++**: The process of obtaining 128 additional features is mentioned above. The PointNet++ dataset is exported as a npy file, loaded into python, and stored as a NumPy array.

After the pre-processing is completed, each dataset is voxel downsampled. Voxel downsampling was chosen over random subsampling to avoid losing important information. Additionally, The ground truth label is removed as clustering with the truth labels yields biased results. Therefore the x,y,z values and an intensity value are used for the raw data implementation. Similarly, the x,y,z values, intensity values and feature descriptors are used for the CloudCompare and PointNet++ implementation. Hyperparameter tuning is implemented on the unsupervised models before clustering occurs to maximise accurate results. Recall that once the clusters are formed, the binary classification task is implemented following the strategy of Marais et al [25]. To obtain the predicted labels, the majority ground truth value of a particular cluster is assigned to each point in that cluster.

Once clustering and classification is completed, the models are visualised and evaluated. The experiments aim to determine whether the algorithms can simplify and accelerate the point cloud cleaning task by producing semantically meaningful clusters visually and numerically. Additionally, we want to discover if the produced clusters can augment the classification task. The above is executed by utilising internal and external validation metrics to assess the performance of the clustering and classification tasks, respectively. Silhouette score, Davies Bouldin (DB) index and BIC score are chosen as they are the most recommended cluster validity indices used throughout the literature [4, 7, 22, 26]. Similarly, that is why recall, F1 score, Intersection over Union (IOU), and the precision metric are selected.

All metrics are implemented using Sklearn.metrics library. This choice was made based on the ease of testing the models with these libraries. The Sklearn.metrics library allows the user to select desired parameters. The 'average' parameter for the classification metrics determines the type of averaging performed on the data. 'Macro' averaging is chosen as it treats all the binary classes equally by calculating the metrics for each label to find their unweighted mean. Moreover, the cluster validity metrics are tested up to 100 clusters, so an exhaustive evaluation is completed while trying to ensure a semantically meaningful visual result. The results are written to an external file where graph construction is implemented to visually compare the metrics.

Note on downsampling: Due to the effect of downsampling, the labels lose their binary values of 0 and 1 and become decimal values ranging between 0 and 1. Ceil(x) is implemented to combat this adverse effect. A ground truth label bigger than 0 became 1, categorising the point as 'discard'. Therefore the results of the classification task throughout the following section could exhibit potential bias.

[4] a las file is a file format designed for the interchange and archiving of lidar point cloud data

5 Experimental Results

5.1 Hyper Parameter Tuning

The hyperparameters of each clustering algorithm are investigated to achieve the most satisfactory clustering results. Experimenting with the parameters of k-means such as number of initialisations and maximum iterations exhibited minimal difference from using the default parameters. Therefore, the original parameters remained. The implementation of fuzzy c-means (FCM) did not allow for parameter tuning, only changing the number of input clusters. K-medians was initialised using random initialisation from the pyclustering library instead of the user choosing the initial medians. However, the algorithm that did yield robust results when hyperparameter tuning was utilised is the Gaussian mixture model (GMM). The Scikit-learn implementation of GMM comprises various parameters. The majority remained as the default, such as tolerance for the stopping criteria, maximum iteration, initial parameters (k-means), weights, means, precision, random state, warm start, and verbose interval. However, the covariance type and the regularisation covariance parameter were explored. Choosing the correct covariance type makes a substantial difference to results, as a Gaussian distribution is determined by its covariance matrix. Spherical covariance comprises spherical contours and is circular shaped in higher dimensional spaces, whereas clusters taking any shape and position correspond to full covariance. Spherical covariance exhibited robust results in Silhouette Score and Davies Bouldin index on all three datasets meaning the clusters that were formed are more separated and compact. However, full covariance showed more success in classification. These results will be unpacked and discussed in greater detail throughout this section.

5.2 Raw Dataset

The raw dataset was voxel downsampled by 0.085, outputting a model with approximately 100 thousand points.

5.2.1 ***Raw data Clustering results*** As mentioned above, partitional algorithms require the user to specify 'k', the cluster amount, beforehand. Therefore, choosing a suitable 'k' value is critical to the success of the algorithm in producing semantically meaningful clusters. The input amount of clusters for the partitional algorithms was chosen based on the silhouette score and DB Index. Two clusters gave the highest silhouette score and lowest DB Index. However, as these metrics are just a guideline, visually and intuitively, two clusters do not make sense.

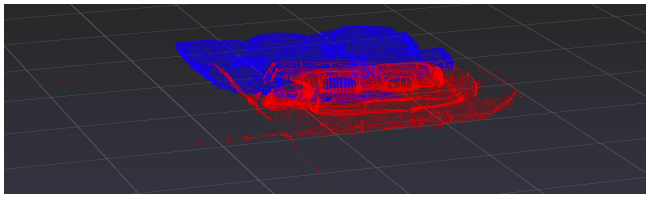

Figure 4: Two clusters fed through K-means, visualised using pptk

To obtain semantically meaningful clusters, more detail is required than shown in Figure 4. Ideally, the clustering results should show outliers as several clusters as well as place different parts of the church model into different clusters. Therefore, the five highest silhouette scores and lowest DB Indices were considered to achieve a more semantically meaningful result. 13, 39, and 38 clusters were chosen for K-means, K-medians and FCM, respectively.

Moreover, to select the correct model for GMM, the Bayesian information criterion (BIC) score was utilised. Therefore, the optimal model and cluster amount is a 'full' covariance model with 44 clusters. However, as mentioned, 'spherical' covariance yields superior cluster validity metrics. This result could be because the method used to initialise the weights, means and precision in the Scikit-learn implementation of GMM is based on K-means. K-means produces clusters that are spherical in shape [12]. However, the BIC score, classification results, and a visual that is as semantically meaningful as using 'spherical' covariance prove that using 'full' covariance is optimal despite what the internal validity metrics indicate.

Therefore, all algorithms can produce clusters that are separated and distinct from one another exhibited in appendix section A.1. However, k-means and k-medians perform slightly better due to their minimised DB index and maximised silhouette score respectively. Refer to appendix section A.2.

5.2.2 ***Raw data Classification results*** The results of the classification task are exhibited in the figures below. Figure 5 is a visual of the actual ground truth values. The blue part of the model is the points considered as 'keep' as they belong to the CH site. The red part is the 'discard' points.

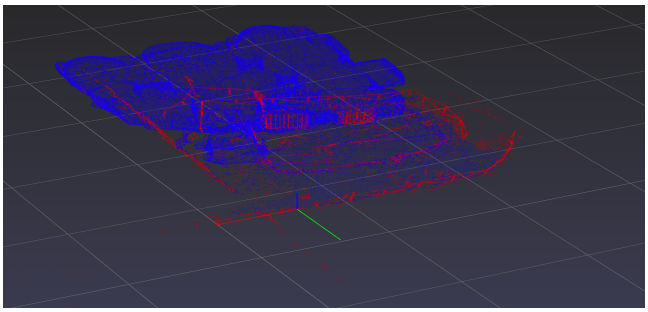

Figure 5: Visualisation of the actual ground truth labels

Figure 6: Comparison of classification performance metrics on the raw dataset

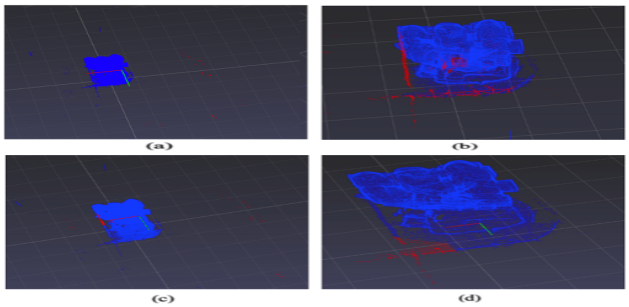

Figure 7: Visualisation results of the predicted ground truth labels on the raw dataset. (a): K-means, (b): Gaussian mixture model, (c): K-medians, (d): Fuzzy c-means

GMM exhibits superior classification results, which can be confirmed numerically in figure 6 and visually in figure 7. Due to the mixture model being multi-modal, it can classify the CH site's different components and distinguish between noise and relevant points. Numerically, this is shown by the F1 score, IOU and recall metric being closer to one than any other algorithm. Visually this is depicted by a very similar figure of the actual ground truth labels versus the predicted ground truth labels obtained in the classification task. Obtaining such robust classification results means that the model's previously produced clusters are compact, cohesive, well separated and meaningful.

K-medians and FCM classify a small part of the side of the CH site as 'discard'; however, the algorithms fail to capture the more intricate details.

K-means has the highest precision score, which can be attributed to the algorithm correctly classifying many of the 'keep' points. However, the recall, F1 and IOU scores are the lowest. K-means subpar performance is depicted visually in Figure 7, with a tiny portion of the outliers correctly classified.

Since GMM, FCM, as well as k-medians produced satisfactory to average results, respectively, it was difficult to accept that k-means performance was of a much lower standard. Therefore an exhaustive k-search was executed. Results found that using k-means with an input k value of 3000 yields successful classification results. All metric scores are very close to 1 and the visual is almost identical to the actual ground truth visual. Refer to appendix, section A.3, for classification performance metrics.

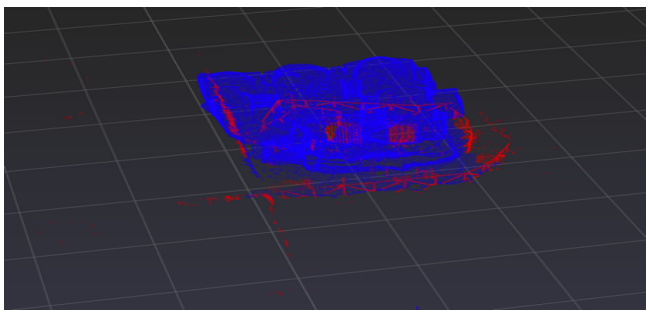

Figure 8: Results of the classification task on k-means using 3000 clusters

Although 3000 clusters improves classification results, the clusters do not fully or partially represent any human-identifiable structure in the point cloud. K-means consequently falls short in that regard as 3000 clusters is meaningless for the point cloud cleaning task. Refer to appendix section A.3.

5.3 CloudCompare Dataset

The dataset with CloudCompare selected features was voxel downsampled by 0.085, outputting a model with approximately 100 thousand points. The selected features are described above in section 3.2.1

5.3.1 ***Clustering results with selected features*** Figure 9 shows the results of the cluster validity metrics

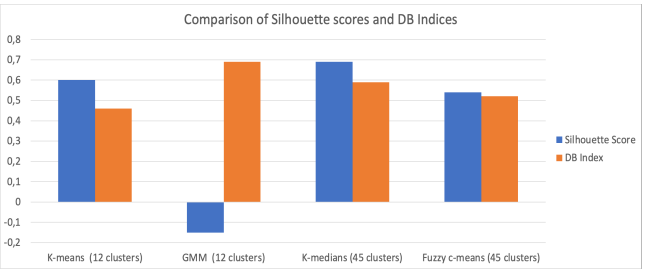

Figure 9: Comparison of clustering performance metrics on the dataset with CloudCompare selected features

K-medians and K-means once again have the highest silhouette score and lowest DB Index, respectively. Additionally, the clustering metrics for the aforementioned algorithms and FCM have improved. By using selected features, more helpful information is encoded into the dataset; therefore, superior metrics are as expected.

The silhouette score for GMM using a 'full' covariance does seem concerning as a negative result indicates that clusters formed are not distinct from one another. When the covariance type is changed to 'spherical', the silhouette score dramatically increases to range between 0.5 and 0.6, and the DB index lowers back down to around 1. However, once again, 'full' covariance is selected due to the same reasons mentioned in section 5.2.1. Refer to appendix section B.1 for clustering visuals.

5.3.2 ***Classification results with selected features*** Figure 10 and figure 11 exhibit the classification results on all the clustering algorithms.

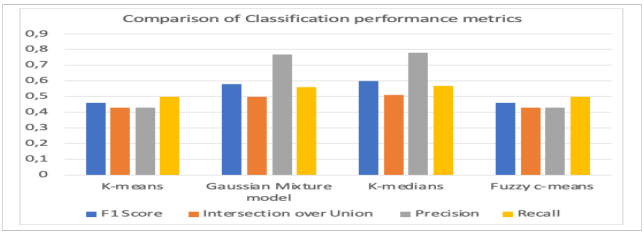

Figure 10: Comparison of classification performance metrics on the dataset with CloudCompare selected features

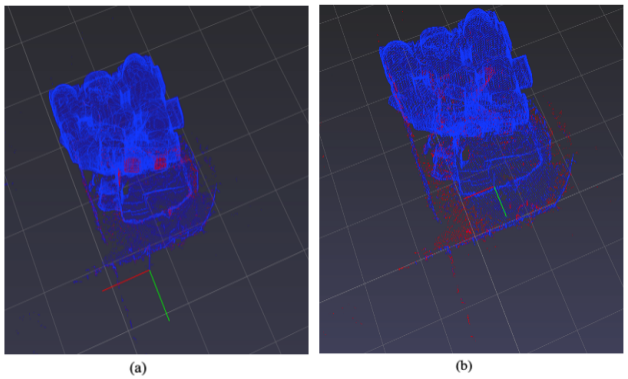

Figure 11: Visualisation results of the predicted ground truth labels on the dataset with CloudCompare selected features. (a): Gaussian mixture model, (b): K-medians

K-means, GMM and FCM have lower classification scores than previously; however, k-medians have improved as seen in figures 10 and 11. Unfortunately, the results of K-means and FCM were insufficient as all clusters were classified as keep resulting in an entirely blue model therefore, the visual was omitted. The results of the aforementioned algorithms and the GMM are not as expected as using a richer dataset, led to more densely packed and cohesive clusters. Therefore, one would expect superior classification results. However, it is clear that the produced clusters are not representing the correct structures of the CH site.

Although the performance of the GMM is successful, The model cannot classify the noise points as distinctly as it did previously. Additionally, FCM actually performs worse on the feature dataset. Initially, we thought the number of clusters was a contributing factor to the poor result; therefore, the algorithm was additionally tested on 38 clusters as before, as well as 100 clusters. However, the results were still insufficent. According to Deng et al. [10] FCM sometimes struggles to deal with more complex features, which could be one of the reasons contributing to its poor performance on the feature dataset.

Recall that k-means was tested with 3000 clusters on the raw dataset therefore, the same strategy was followed to see if we could obtain an improved classification result. The result is shown in appendix section B.2. K-means identifies a few noise points however, visually and numerically, its performance cannot compare to the above as it is a lot poorer. This could be attributed to clustering in a higher dimensional space. The clusters produced by k-means on the feature dataset are different sizes and non-globular in shape. A limitation of k-means is that it has trouble producing robust results on these kinds of clusters [13]. Additionally, according to Aggarwal et al. [2], when clustering in a higher dimensional space, the distance metrics which are used to measure the similarity between objects must be accounted for. They explain, that the maximum and minimum distances between any two points are likely to converge, rendering a proximity query meaningless for some distance measures. This adverse effect is noticeable when using the euclidean distance metric, which is what k-means and FCM are based on.

Contrariwise, the K-medians algorithm produced robust results on the feature dataset as significantly more noise points are classified as 'discard'. In specific, the F1 score, which combines precision and recall, is improved, showing that the proportion of points that were actually part of the CH site and classified as part of the site is more accurate. K-medians utilise a Manhattan distance metric and therefore do not suffer the same adverse effect as k-means and FCM when clustering in a higher dimensional space. Although the results are improved, the finer characteristics of the CH site such as the scaffolding in the middle are categorised as 'keep', which is incorrect.

5.4 PointNet++ Dataset

Due to resource constraints, the raw point cloud was down-sampled by 0.075 before being fed into the PointNet++ application. Once PointNet++ was completed, the new feature set was additionally down-sampled by 0.085 to obtain a final point cloud comprising approximately 85 thousand points.

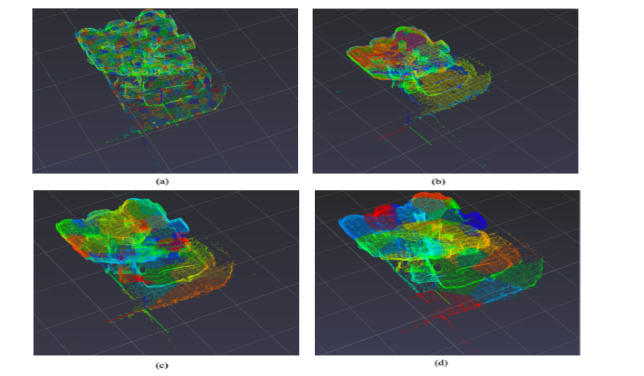

Figure 12: Visualisation of the clustering results on the PointNet++ dataset. (a): K-means, (b): Gaussian mixture model, (c): K-medians, (d): Fuzzy c-means

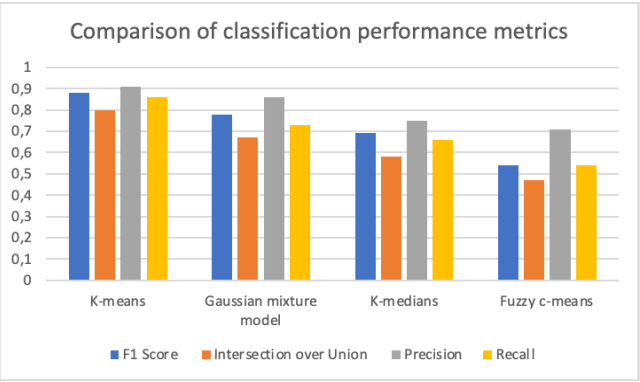

Figure 13: Comparison of classification performance metrics on the dataset with PointNet++ features

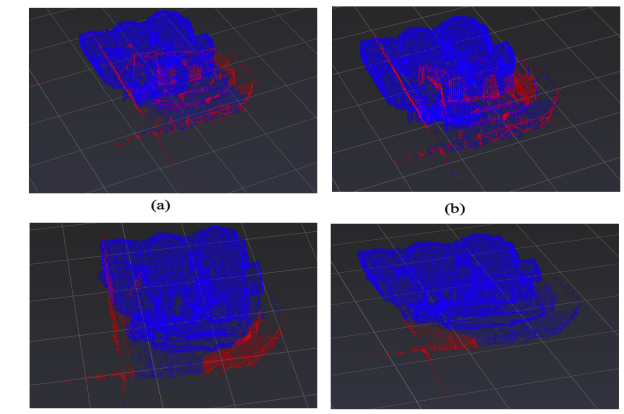

Figure 14: Visualisation results of the predicted ground truth labels on the PointNet++ dataset. (a): K-means, (b): Gaussian mixture model, (c): K-medians, (d): Fuzzy c-means

Recall that K-means performed optimally when using a more considerable input cluster amount. Therefore 100, 1000 and 3000 input clusters were tested as starting points. 3000 input clusters were chosen due to their more desirable results in the classification task. Choosing an input amount of k = 3000 does not produce semantically meaningful clusters that a human can interpret, as shown in Figure 12. However, K-means creates semantically meaningful clusters in terms of keep and discard, as shown in Figures 13 and 14.

Moreover, using the Silhouette score and DB Index as a guide and trial and error, 28 and 67 clusters were chosen for FCM and k-medians, respectively. Additionally, 44 clusters are selected for GMM based on the BIC criterion. Refer to appendix section C.1 for clustering results. The cluster validity metrics are poorer than the previous datasets; however, the clusters produced are semantically meaningful. Therefore, using the PointNet++ dataset is still beneficial as the manual task of point cloud cleaning can be simplified and accelerated.

In addition, the results of the classification task on the aforementioned algorithms have improved. FCM performs significantly better on the PointNet++ data set compared to the CloudCompare data set. This result suggests, The selected CloudCompare features may have not been entirely suitable for the FCM algorithm. The improved classification result for k-medians is expected, as we saw above, that k-medians seem to perform more optimally when feature descriptors are added to the dataset. Finally, GMM produces robust results as the classification performance metrics are closer to one than previously. The model can classify well and distinguish between noise points and points which belong to the CH site. Using more descriptors allows GMM to produce loosely defined clusters; however, the clusters form part or all of some structure of the CH site; therefore, they are *useful* to a human and in terms of 'keep' and 'discard'.

6 Discussion

The first research question we aimed to answer is *"does K-medians, GMM and FCM produce clusters on point clouds of CH sites that are more semantically meaningful than those produced by k-means algorithm, using cluster validity metrics of Silhouette Coefficient and DB Index?"*.

When answering this question, clustering metrics and visuals must be considered. Figure 15 shows the results of the cluster validity metrics throughout all three datasets.

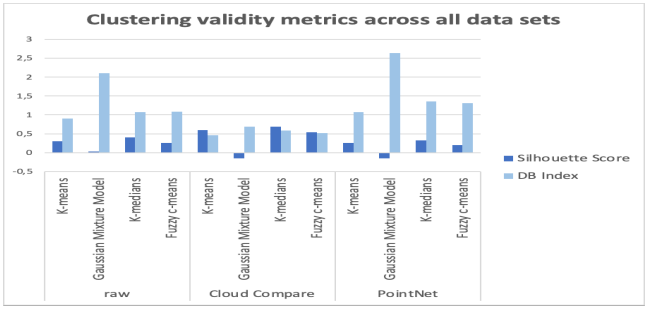

Figure 15: clustering metrics across all three datasets

As one can see, the silhouette scores are closer to zero, indicating that the clusters are not very compact and cohesive. However, this result is acceptable as the model that was tested contains many challenging regions and represents the upper bound complexity found in CH point clouds. Therefore robust metrics are difficult to achieve. K-means maintains the lowest DB index; however, k-means does not outperform GMM, FCM and K-medians in its entirety as its silhouette score is not the highest; k-medians is throughout. Therefore, in terms of cluster validity metrics as a whole, we cannot conclude that one algorithm outperforms the rest. However, we can say that on all three datasets, k-means is forming clusters with high intra-cluster similarity and low inter-cluster similarity due to its minimised DB Index. Whilst, k-medians are forming more cohesive clusters due to their maximised silhouette score.

Although these metrics are helpful, cluster validation is a difficult task and lacks the theoretical background other areas of machine learning have [4]. It is even more challenging when the underlying structure of the data is unknown such as in this context. Therefore, it is difficult to judge which algorithm and dataset are optimal for the task of point cloud cleaning based *only* on the above metrics. We must also assess how semantically meaningful the produced clusters are. If we were to consider the metrics in isolation, we would conclude that all algorithms perform best on the CloudCompare dataset; however, visually, that is not the case. The CloudCompare clustering visuals are complicated for a human to decipher, making this dataset relatively *meaningless* for the desired task of removing clusters one by one. It is likely that the features selected were not suited to the partitional algorithms and the GMM. If they were, the simplest algorithm would be able to manufacture visually separated clusters [20]. Selecting a smaller set of features may produce more semantically meaningful clusters as the dimensionality issue is less of a problem.

Moreover, all algorithms produce satisfactory visual results on the raw dataset and PointNet++ dataset as shown in appendix sections A.1 and section 5.4. The produced clusters are semantically meaningful as they correspond to particular structures of the CH site. This is useful for the task of point cloud cleaning as even though there is sometimes overlap in particular clusters between noise points and relevant points, the algorithms are producing results that can still accelerate the point cloud cleaning process. It is simpler and more effective to remove some clusters and a few noise points than removing thousands of points.

Additionally, no algorithm substantially produces more "useful" clusters visually than the other algorithms. For example, choosing a 'k' of 3000 for k-means does not produce clusters that are valuable to a human; however, we cannot say that the algorithm cannot produce clusters that are. The only reason 3000 clusters were tested was to maximise classification performance results.

Moreover, the second research question investigated was *"what is the upper bound accuracy when using the resulting clusters to perform binary classification using' keep' and' discard' labels on a point cloud?"*. Again, we must assess the classification performance metrics across all three datasets.

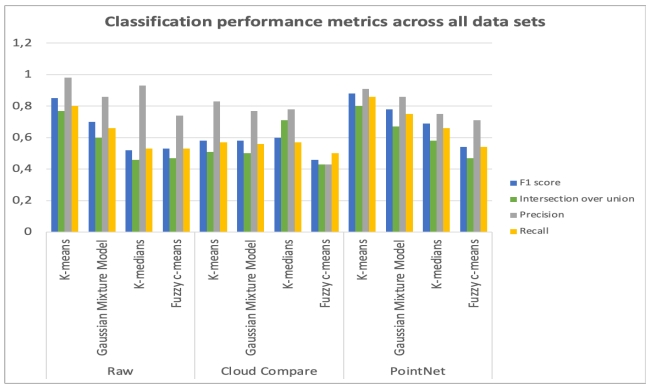

Figure 16: classification metrics across all three datasets

Figure 16 shows that all classification results are significantly better when utilising the PointNet++ dataset, as the classification performance metrics for all algorithms are closer to 1. Therefore, our expectation of using many additional feature descriptors to produce meaningful clusters is confirmed. K-means performance is superior as recall, precision, IOU, and F1 score are 86%, 91%, 80% and 88%, respectively.

However, It must be noted that vast numbers of clusters are not ideal, even if they optimise classification results. As mentioned, we would like a human operator to be able to select all clusters that they want to throw away. That means manually selecting thousands of clusters for 'cleaning' is not viable. Therefore, basing our conclusion on the above logic exhibits that the results of k-means are unsuccessful in this respect, as its optimal performance is based on 3000 input clusters. This leads us to conclude that GMM produces the most satisfactory results on the PointNet++ dataset for the *desired* task. Its high classification performance metrics, and semantically meaningful classification and clustering visuals

confirms the above. Additionally, the produced clusters mainly consist of keep or discard labels and not a potent mix of both.

7 Conclusions

The manual task of point cloud cleaning is taxing and laborious. We have proposed a framework that addresses the simplification and acceleration of the cleaning process through unsupervised learning. The unsupervised approach is implemented without requiring ground truth labels which is advantageous as CH sites have limited labelled training data.

Our research shows that the unsupervised methods produce satisfactory results on the raw dataset and PointNet++ dataset. K-means, GMM, FCM, and K-medians can produce semantically meaningful clusters that humans are able interpret. Therefore, to create 3D models of CH sites with only desirable geometric features, the unwanted clusters can be easily discarded instead of manually removing thousands of points.

Moreover, manually adding selected features to the dataset, as seen by the results of CloudCompare, proved to be insufficient. Although the internal validity metrics were high, the visuals of the algorithms are challenging to decipher, making them relatively meaningless for the desired task. However, this method should not be out-ruled as, given more time, different feature combinations could be explored to yield success. It is clear that selecting the correct feature extraction approach that identifies the underlying clustering structure of the data is beneficial as seen by most satisfactory results of PointNet++.

Although clustering exhibited success, the investigated algorithms present trade-offs as they require the user to select the input number of clusters. On the one hand, it increases the likelihood that the algorithms will produce semantically meaningful clusters. On the other hand, finding the optimal cluster amount proved to be a tedious task. Additionally, the cluster validity metrics do not always represent how *useful* the produced clusters are. This makes the decision of choosing the optimal algorithm complex.

This paper aimed to assess whether the unsupervised machine learning algorithms could form semantically meaningful clusters and then evaluate the upper bound accuracy on the resulting clusters to perform binary classification. Considering the above, the Gaussian mixture model proved optimal in all respects in comparison to k-means, FCM and k-medians. It must be noted that its superior performance is exhibited on the PointNet++ dataset. GMM produced semantically meaningful clusters. Additionally, these clusters augment the classification task.

7.1 Limitations and Future work

The findings of this study offer numerous prospects for additional investigation. A more exhaustive 'k' search should be implemented using additional internal validity metrics to improve performance. Additionally, we aim to extend the partitional and mixture clustering algorithms to different families of unsupervised algorithms.

8 Acknowledgements

We want to thank associate professor Patrick Marais and co-supervisor Luc Hayward for their dedicated help, support and contribution throughout the entire project

References

[1] Abbas, O. A. Comparisons between data clustering algorithms. *International Arab Journal of Information Technology (IAJIT) 5*, 3 (2008).

[2] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory* (2001), Springer, pp. 420–434.

[3] Albano, R. Investigation on roof segmentation for 3d building reconstruction from aerial lidar point clouds. *Applied Sciences 9*, 21 (2019), 4674.

[4] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., and Perona, I. An extensive comparative study of cluster validity indices. *Pattern recognition 46*, 1 (2013), 243–256.

[5] Belton, D. Processing tree point clouds using gaussian mixture models.

[6] Biosca, J. M., and Lerma, J. L. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS Journal of Photogrammetry and Remote Sensing 63*, 1 (2008), 84–98.

[7] Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., and Dougherty, E. R. Model-based evaluation of clustering validation measures. *Pattern recognition 40*, 3 (2007), 807–824.

[8] Cardot, H., Cénac, P., and Monnez, J.-M. A fast and recursive algorithm for clustering large datasets with k-medians. *Computational Statistics & Data Analysis 56*, 6 (2012), 1434–1449.

[9] Chen, S. S., and Gopalakrishnan, P. S. Clustering via the bayesian information criterion with applications in speech recognition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)* (1998), vol. 2, IEEE, pp. 645–648.

[10] Deng, S. Clustering with fuzzy c-means and common challenges. In *Journal of Physics: Conference Series* (2020), vol. 1453, IOP Publishing, p. 012137.

[11] Dewez, T. J., Girardeau-Montaut, D., Allanic, C., and Rohmer, J. Facets: A cloudcompare plugin to extract geological planes from unstructured 3d point clouds. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 41* (2016).

[12] Erman, J., Arlitt, M., and Mahanti, A. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data* (2006), pp. 281–286.

[13] Ertöz, L., Steinbach, M., and Kumar, V. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 2003 SIAM international conference on data mining* (2003), SIAM, pp. 47–58.

[14] Farella, E. M. 3d mapping of underground environments with a hand-held laser scanner. *Bollettino della società italiana di fotogrammetria e topografia*, 2 (2016), 1–10.

[15] Ferri, C., Hernández-Orallo, J., and Modroiu, R. An experimental comparison of performance measures for classification. *Pattern recognition letters 30*, 1 (2009), 27–38.

[16] Ghosh, S., and Dubey, S. K. Comparative analysis of k-means and fuzzy c-means algorithms. *International Journal of Advanced Computer Science and Applications 4*, 4 (2013).

[17] Grilli, E., Menna, F., and Remondino, F. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 42* (2017), 339.

[18] He, X., Cai, D., Shao, Y., Bao, H., and Han, J. Laplacian regularized gaussian mixture model for data clustering. *IEEE Transactions on Knowledge and Data Engineering 23*, 9 (2010), 1406–1418.

[19] Hui, Z., Jin, S., Li, D., Ziggah, Y. Y., and Liu, B. Individual tree extraction from terrestrial lidar point clouds based on transfer learning and gaussian mixture model separation. *Remote Sensing 13*, 2 (2021), 223.

[20] Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters 31*, 8 (2010), 651–666.

[21] Juan, A., and Vidal, E. Comparison of four initialization techniques for the k-medians clustering algorithm. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)* (2000), Springer, pp. 842–852.

[22] Lord, E., Willems, M., Lapointe, F.-J., and Makarenkov, V. Using the stability of objects to determine the number of clusters in datasets. *Information Sciences 393* (2017), 29–46.

[23] Madhulatha, T. S. An overview on clustering methods. *arXiv preprint arXiv:1205.1117* (2012).

[24] Mahdaoui, A., Bouazi, A., Hsaini, A. M., and Sbai, E. H. Comparison of k-means and fuzzy c-means algorithms on simplification of 3d point cloud based on entropy estimation. *Adv. Sci. Technol. Eng. Syst. J 2*, 5 (2017), 38–44.

[25] Marais, P., Dellepiane, M., Cignoni, P., and Scopigno, R. Semi-automated cleaning of laser scanning campaigns with machine learning. *ACM Journal on Computing and Cultural Heritage 12*, 3 (2019), 1–29.

[26] Milligan, G. W., and Cooper, M. C. An examination of procedures for determining the number of clusters in a data set. *Psychometrika 50*, 2 (1985), 159–179.

[27] Nguyen, A., and Le, B. 3d point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)* (2013), IEEE, pp. 225–230.

[28] Palacio-Niño, J.-O., and Berzal, F. Evaluation metrics for unsupervised learning algorithms. *arXiv preprint arXiv:1905.05667* (2019).
[29] Poux, F. Guide to real-time visualisation of massive 3d point clouds in python tutorial for advanced visualization and interaction with big point cloud data in python.(bonus) learn how to create an interactive segmentation "software".
[30] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660.
[31] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems 30* (2017).
[32] Shan, J., and Sampath, A. Building extraction from lidar point clouds based on clustering techniques. In *Topographic Laser Ranging and Scanning*. CRC Press, 2017, pp. 421–444.
[33] Weinmann, M. Feature relevance assessment for the semantic interpretation of 3d point cloud data.
[34] Zhou, Q.-Y., Park, J., and Koltun, V. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847* (2018).

A Appendix

A.1 Raw dataset clustering visuals

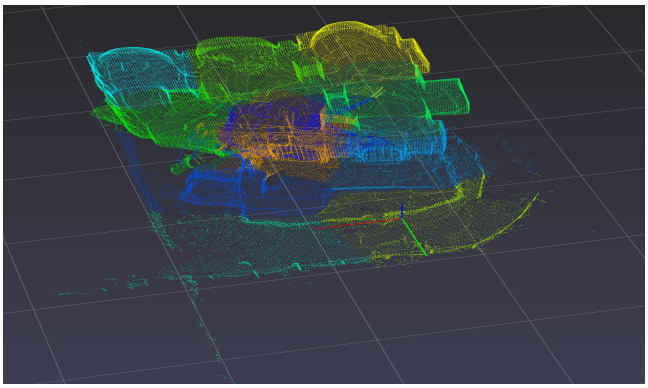

Figure 17: Resultant clusters formed by k-means using an input amount of 13 clusters

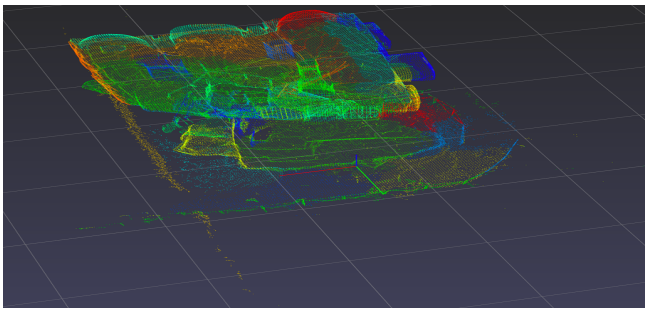

Figure 18: Resultant clusters formed by Gaussian mixture model using an input amount of 44 clusters

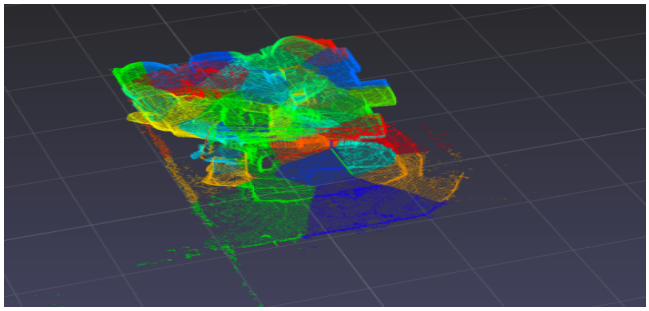

Figure 19: Resultant clusters formed by K-medians using an input amount of 39 clusters

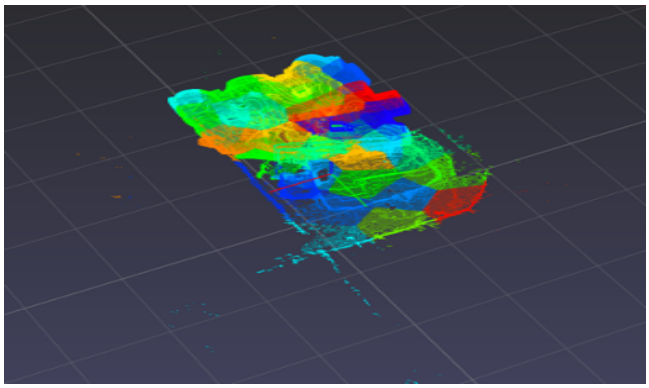

Figure 20: Resultant clusters formed by Fuzzy c-means using an input amount of 38 clusters

A.2 Raw dataset clustering metrics

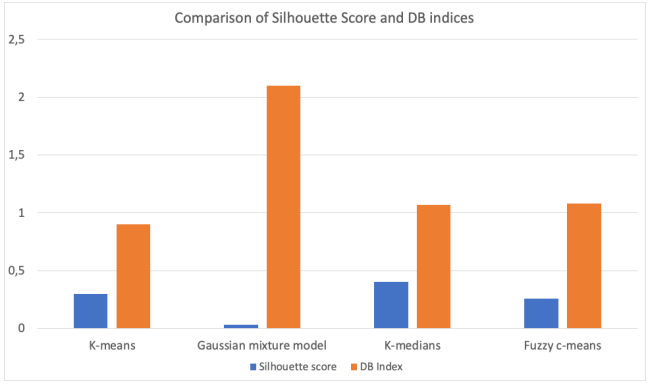

Figure 21: Comparison of clustering performance metrics on the raw dataset

A.3 K-means: 3000 clusters

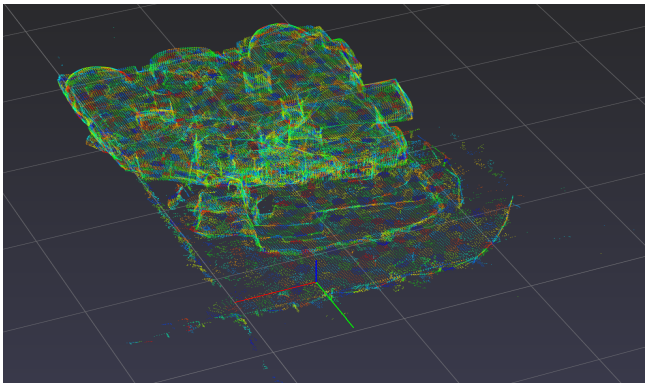

Figure 22: Resultant clusters formed by k-means using an input amount of 3000 clusters

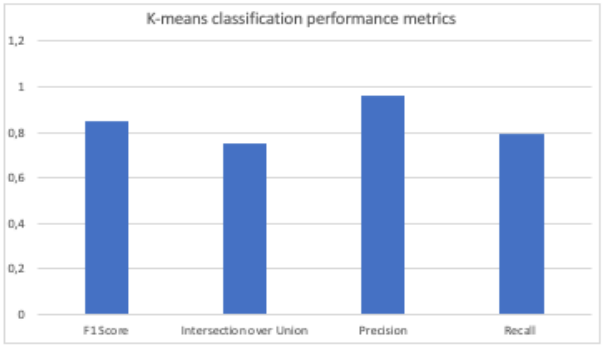

Figure 23: Results of the classification metrics on k-means using 3000 clusters

B CloudCompare results

B.1 CloudCompare dataset visuals

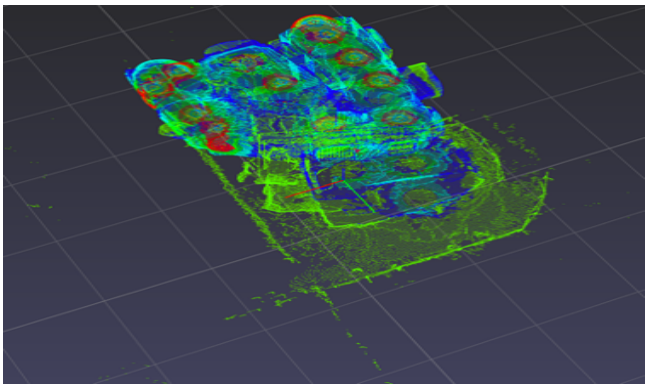

Figure 24: Resultant clusters formed by k-means using an input amount of 12 clusters

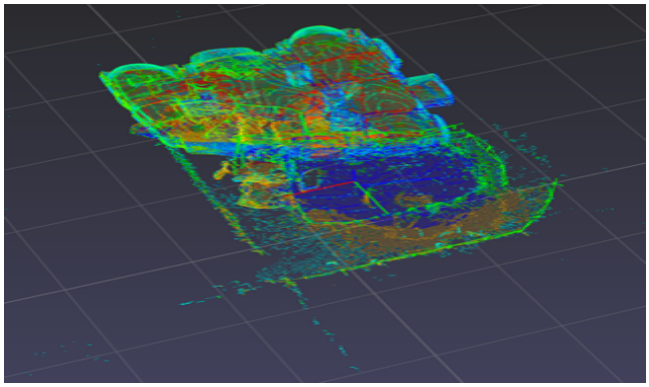

Figure 25: Resultant clusters formed by Gaussian mixture model using an input amount of 12 clusters

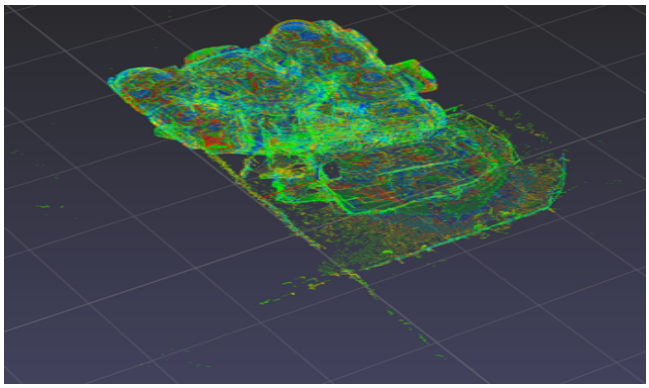

Figure 26: Resultant clusters formed by K-medians using an input amount of 45 clusters

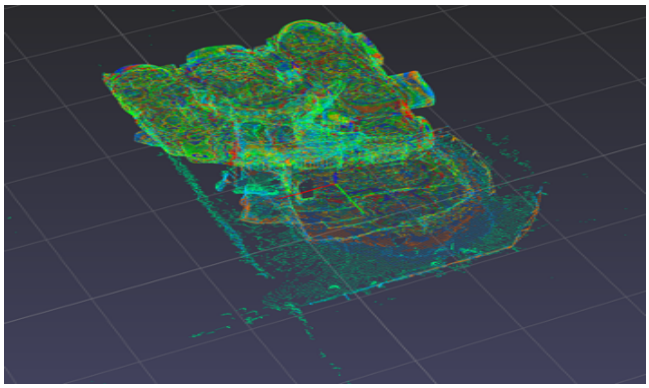

Figure 27: Resultant clusters formed by Fuzzy c-means using an input amount of 45 clusters

B.2 K-means classification on the CloudCompare dataset

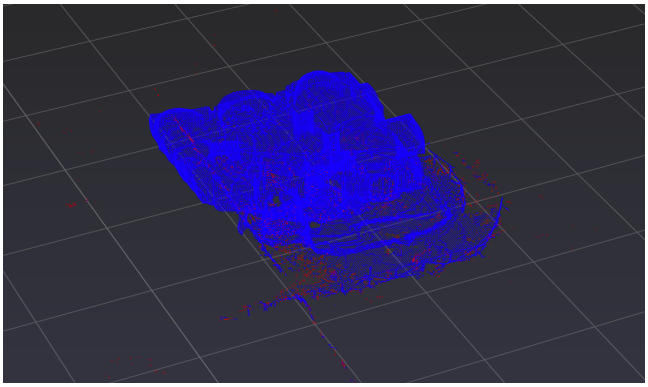

Figure 28: classification task on k-means with 3000 input clusters

C PointNet++ Results

C.1 PointNet++ dataset clustering metrics

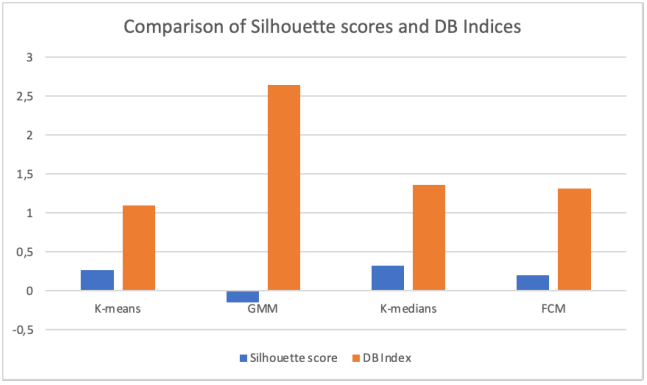

Figure 29: Comparison of clustering performance metrics on the PointNet++ dataset