

AI Indexing of Aerial LiDAR Point Cloud for Efficient Query

Mohana Murugan¹, Anandakumar M Ramiya²

¹ Indian Institute of Space Science and Technology, Thiruvananthapuram, India - mohanamurugan00@gmail.com

² Indian Institute of Space Science and Technology, Thiruvananthapuram, India - ramiya@iist.ac.in

Keywords: Aerial LiDAR point cloud, Tree species identification, 3D Query, PostgreSQL, Anomaly detection, Urban Planning.

Abstract

In the era of information revolution, with data being the fuel of AI and analytics, efficient information extraction from LiDAR point clouds becomes indispensable for solving real-world problems and aiding decision-making in geospatial domain. Despite having geometric richness, the massive LiDAR point clouds are not only computationally demanding but also lack inherent semantics. The lack of semantics in LiDAR constrains effective data analysis. This paper presents a novel workflow by incorporating Deep Learning derived embeddings as attributes in the geospatial database for the spatio-semantic querying on Aerial LiDAR point clouds. This work leverages AI-based indexing, such as IVFFlat(Inverted File Index with Flat Quantization) on LiDAR point clouds for fast retrieval of queries. The pgPointCloud and pgVector extensions of PostgreSQL aid in importing point clouds into the database and performing similarity-based query retrieval on the embedding space of the point clouds. The methodology developed in this paper explores how semantic embeddings can handle inadequate semantics of point clouds by enabling direct and complex 3D intelligent queries within the database environment, thereby overcoming the limitations of traditional LiDAR representations. Few queries presented in this paper highlight the applications of this proposed framework in individual tree detection, tree species identification, utility management, urban planning and anomaly detection.

1. Introduction

Countries across the globe are involved in LiDAR data acquisition for high-resolution terrain mapping and the development of digital twins. Applications in the fields of infrastructure management, urban planning, climate resilience, forest monitoring, and disaster preparedness are supported by these national and regional datasets. Digital twin city modelling further enhances decision-making with data-driven city management. To align with the applications, effective information extraction from the data is essential. Like raster and vector data, LiDAR data is now becoming the new norm in geospatial analysis, providing rich 3D information on terrain, vegetation, buildings, and other natural and man-made features, enabling modelling the complexity of real-world environments. Despite its potential, analysing LiDAR data remains a challenge, as the points are unordered, unstructured, and irregular. As the volume and density of LiDAR datasets grow, deep learning-based architectures have emerged as powerful tools to classify the data with high accuracy by extracting meaningful patterns and semantic information that cannot be extracted using traditional machine learning approaches with handcrafted features. A comprehensive review is presented in (Jia et al., 2025) of deep learning-based architectures for efficient classification, segmentation, and object detection in LiDAR point clouds. PointNet++ (Qi et al., 2017) is widely regarded as the pioneering deep learning architecture for the classification of LiDAR point clouds. It can be applied to object classification, part segmentation, and semantic segmentation. This architecture incorporated an MLP-based architecture for classification. Although deep learning models successfully assign classification labels to LiDAR points, the difficulty lies in incorporating these classified data into geospatial databases for effective query operations. Traditional spatial databases are designed for storing and querying vector or raster data. However, a framework for querying large-scale LiDAR point clouds, especially based on semantic information, remains limited. Earlier work

by (Schön et al., 2009) addresses the shortcomings of traditional technology in storage, manipulation and visualisation of point clouds and emphasises the need for 3D indexing. The paper by (Sadineni, 2020) presents broader surveys on indexing techniques such as AI indexing, non-AI indexing and collaborative indexing for handling big data and underscores the growing importance of AI-driven indexing for managing dynamic and large-scale datasets. The comparison of Point Cloud Database Management (PCDM) systems such as Oracle, PostgreSQL and MonetDB is carried out in terms of scalability and performance by (Lokugam Hewage et al., 2022) and attempted to serve as a guide for decision-making in the development of PCDM systems. The study by (Ikawa et al., 2019) discusses the performance of querying point clouds in the RDBMS (PostgreSQL and PostGIS) and revealed that while R-tree or GiST indexing excels for spatial range queries, B-trees can outperform R-trees for rectangular queries. Semantic Data Based Layout(SDBL) framework is proposed by (El-Mahgary et al., 2020) for querying point clouds either through Python or PostgreSQL. The study by (Béjar-Martos et al., 2022) evaluates Cassandra, MongoDB, MySQL, and PostgreSQL for storing large LiDAR datasets and concludes that while all databases benefit from a unified repository with metadata and spatial querying, NoSQL systems, especially Cassandra, deliver consistently better performance and scalability. Despite the advantages of NoSQL systems, DBMS-based workflows are often preferred for geospatial analytics because of seamless integration with spatial information. PostgreSQL is a powerful open-source object-relational database that has extensions such as PostGIS for spatial data analysis, pgPointCloud for point cloud storage, and pgvector for high-dimensional vector indexing and similarity search. These extensions make seamless integration of querying point clouds with GIS vector data and deep learning embedding vectors. Spatial query is querying based on the location/coordinates. A semantic query addresses the inherent meaning of the objects. In this work, we first developed a

feature-enriched spatial database to explore spatial queries. To overcome limitations in spatial querying, we then constructed a spatio-semantic database by integrating embedding vectors as attributes. AI indexing helps with fast retrieval of semantic similarity search queries. The embedding space is a transformed feature space or can be seen as a new representation of point clouds. Just like how NLP embeddings group similar meanings, even though the words are far apart in a sentence, LiDAR embeddings group similar objects or clusters in the embedding space. For instance, two coconut trees that are far apart spatially will be close in embedding space. However, two different tree species that are located near each other will be far apart in the embedding space.

2. Methodology

A workflow for effectively integrating LiDAR characteristics into a database for effective spatio-semantic queries is developed in this study. Two databases are constructed: one for spatial querying and another for spatio-semantic querying, to evaluate the performance of enriched point cloud retrieval. The database for spatial queries is extended with PostGIS and pgPointCloud extensions, and spatial indexing is performed. Likewise, the database for spatio-semantic queries is extended with PostGIS and pgVector extensions, and AI indexing is employed to efficiently retrieve the embedding vector. The overall methodology involves:

- Constructing spatial and spatio-semantic databases, by enriching the point cloud with additional features (clusters, height above ground, embeddings). Semantic segmentation is done only for constructing a spatio-semantic database.
- Executing spatial and spatio-semantic queries directly within PostgreSQL.

2.1 Datasets

The primary focus of this paper is on aerial LiDAR point cloud data querying to support applications in urban planning, forestry, environmental monitoring, and disaster management. Two benchmark datasets of distinct geographic regions are used to explore the spatial and spatio-semantic queries: TALD-Thiruvananthapuram Aerial LiDAR Dataset (Vijaywargiya and Ramiya, 2025) and DALES-Dayton Annotated LiDAR Earth Scan (Varney et al., 2020). Both datasets provide semantic segmentation annotation. The average point density of TALD and DALES varies, the former having 12 points/m² covering urban, peri-urban, and rural areas and the latter having 50 points/m² consisting mainly of urban layouts. TALD is semantically annotated into 4 main classes, such as ground, shrubs, trees, and buildings. DALES is semantically annotated into 8 major classes, such as ground, vegetation, cars, trucks, poles, power-lines, fences, buildings, and an unknown category.

2.2 Spatial Database creation

The LiDAR data must be enhanced with its features for performing effective queries. More enrichment results in more effective queries. For spatial querying, the spatial database is enriched with class labels (from the benchmark datasets), cluster IDs, and height-above-ground (HAG). The spatial database is created for both TALD and DALES datasets. The enriched

point cloud is imported into PostgreSQL using pgPointCloud, which stores data efficiently as PcPatch objects via the chipper filter. PcPatches partition the point cloud into density-based patches, thereby reducing search space and improving query speed. An example of PcPatch generation for TALD shrubs is shown in Figure 1. These PcPatches will be exploded when point-level interactions are required for spatial query operations.

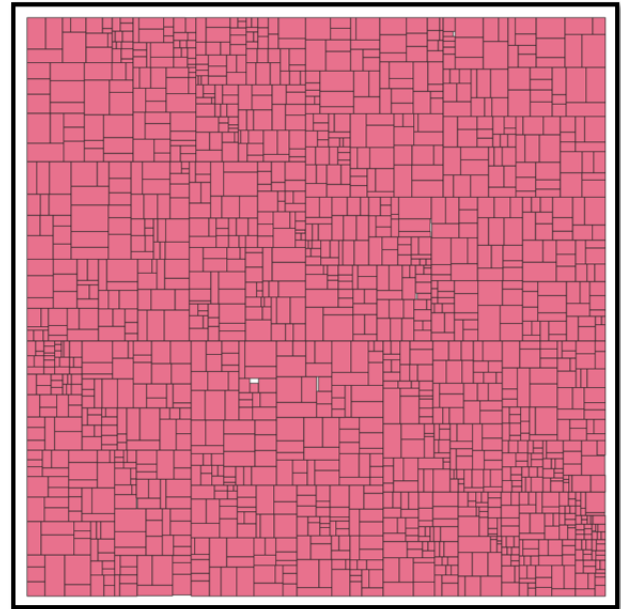


Figure 1. PcPatches of TALD Shrubs

2.3 Enriched features

As discussed earlier, two features are created to enrich the point cloud: clusters and HAG. Clusters are made for individual classes rather than the whole point cloud. The data is separated into different classes, and clustering is performed using the PDAL library as Class-wise clustering provides meaningful object-level grouping (e.g., individual trees or building units). Parameters for clustering, such as clustering tolerance and minimum points for each class, are altered based on the clustered patch accuracy. The cluster having a zero clusterID is ignored. The number of clusters for each class is listed in Table 1

Class Name	Count of Clusters
Ground	98
Shrubs	182
Trees	2784
Buildings	608

Table 1. TALD classes and cluster counts

The z value of the aerial point cloud is the absolute terrain elevation. A shrub on a hill may have the same Z as a tall tree in a valley. So to make sure of the distinction between these and to query the height of trees, buildings, and poles, etc., calculating height above ground is vital. PDAL's Height Above Ground Nearest Neighbor filter is used to derive this attribute.

2.4 Spatio-semantic Database creation

To support spatio-semantic querying, the point cloud is further enriched with embedding vectors. Embeddings can be derived from various point cloud based deep learning architecture. Since PointNet++ is a pioneering architecture and widely

adopted one, it is employed in this study as a representative approach. These embeddings encode the geometric and semantic characteristics of each 3D point. The methodology to construct the spatio-semantic database is shown in Figure 2.

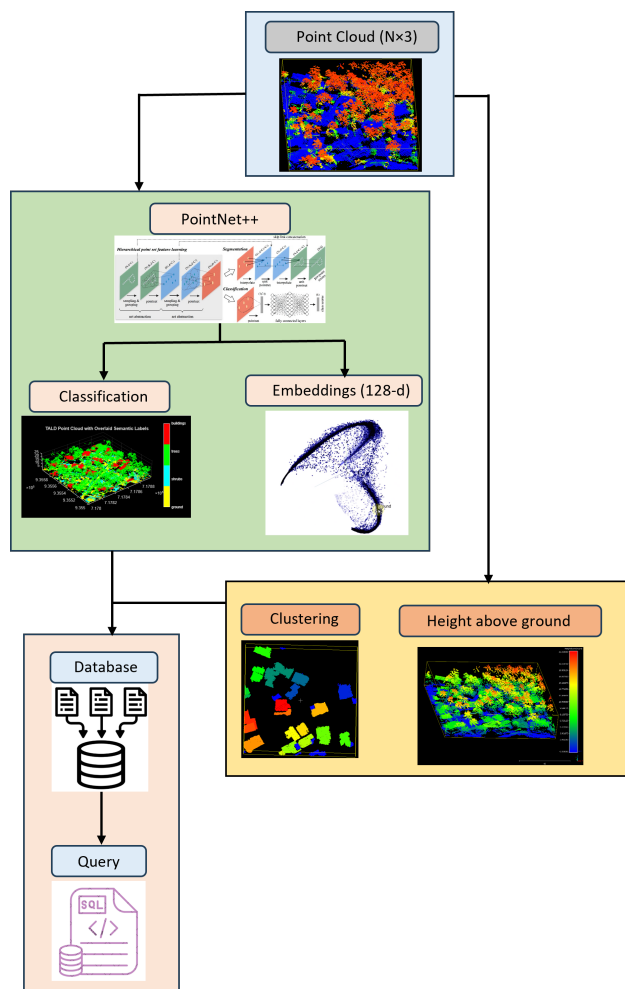


Figure 2. Methodology for creation of spatio-semantic database

The spatio-semantic database is constructed only for the TALD dataset. Once constructed, for querying L2 distance function is used to measure the similarity between embedding vectors, where semantically similar objects have smaller Euclidean distances in the embedding space. For fast effective nearest-neighbour retrieval, embedding vectors are indexed using the IVFFlat (Inverted File Index with Flat Quantization) method, which partitions the embedding space into coarse clusters and accelerates similarity search on large datasets. In 3D LiDAR data, each object possesses distinct geometric characteristics, similar to how objects exhibit unique spectral properties in multispectral imagery. By representing these geometric features in a embedding space, objects can be identified through feature-based matching. The embedding vectors extracted by PointNet++ generate embeddings that capture these semantic relationships, enabling effective object differentiation.

2.5 Embedding Extraction

The TALD benchmark dataset is semantically segmented using PointNet++ architecture, and the deep point set features in the final interpolation layer, consisting of a 128-d embeddings, are extracted as embedding vector for each LiDAR point

(X, Y, Z) for the TALD test set, which consists of approximately 3.6M points. Among the 128 extracted features, two representative dimensions are visualized in Figure 3 and Figure 4 for visualization. Embedding 1 attempts to extract the features

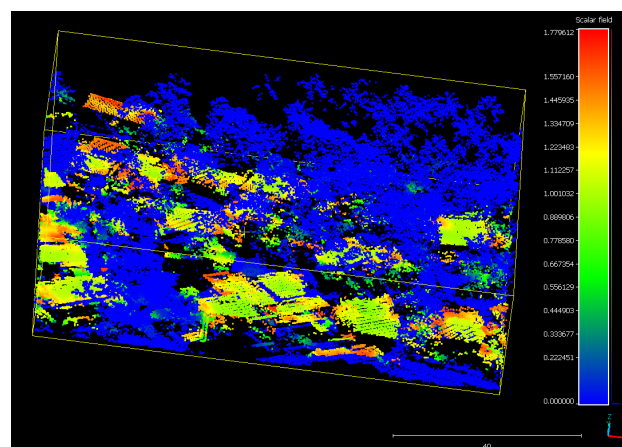


Figure 3. Embedding 1 of 128-d embedding vector

of buildings, and Embedding 3 will play a significant part in tree semantic segmentation. These embeddings are encoded with high-level contextual and geometric information learned by the network. pgVector is employed to store the embeddings

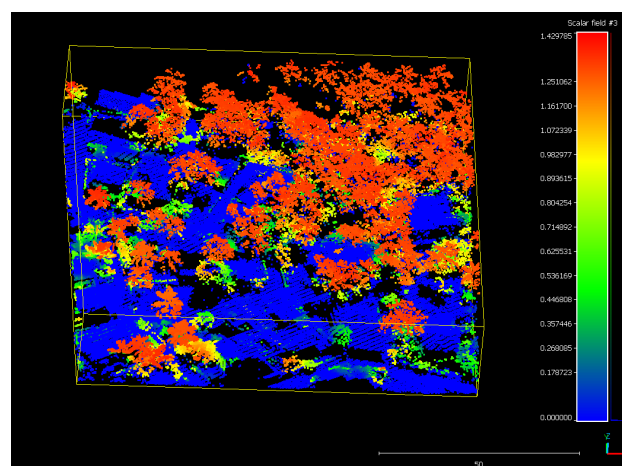


Figure 4. Embedding 3 of 128-d embedding vector

as a vector in PostgreSQL. Figure 5 shows the PCA (Principal Component Analysis) reduced visualization of TALD embeddings using tensorflow embedding projector. In the embedding space, ground and shrubs occupy adjacent regions with a gradual transition rather than a sharp boundary; the same diffuse separation is observed between trees and buildings. These continuous transitions demonstrate that the learned embedding space captures nuanced semantic relationships that are not visible in the raw geometric LiDAR data.

3. Results and Discussions

To highlight the functionality and efficiency of the database environment, four representative queries are presented. The queries performed are given in Table 2. Queries 1 and 2 focus purely on spatial reasoning, showcasing how the database can directly operate in 3D space by querying the point cloud

Query	Dataset	Description	Query Latency
Query:1	DALES	Assessing the vertical clearance between powerlines and the road network at all intersection locations.	1 sec
Query:2	TALD	Identifying nearby buildings and trees that are in close proximity with each other	5.5 min
Query:3	TALD	Find the coconut trees in a buffer of 50 sq. m. of an input coconut tree location	8 sec
Query:4	TALD	Find the anomalous clusters	9 sec

Table 2. Summary of Queries on DALES and TALD

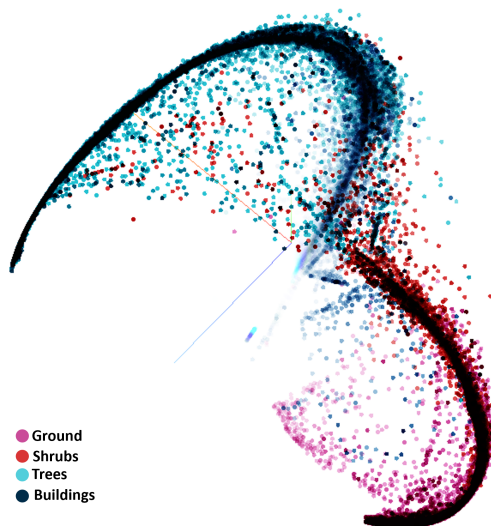


Figure 5. TALD data embedding space

without requiring intermediate raster or vector transformations, whereas Query 3 demonstrates how AI indexing on the embedding vector enables semantic neighbourhood search capabilities that cannot be achieved with spatial and non spatial attributes alone, and Query 4 shows the effectiveness of the framework in identifying anomalous clusters based on geometric and semantic deviations. Query 1 is executed in the DALES spatial database. Queries 2, 3, and 4 are executed in the TALD database. The storage size and the importing time of the spatial database table and spatio-semantic database table into the PostgreSQL is given in Table 3. The following sections discuss the results and interpretations of each query.

3.1 Implementation details

All algorithms, model architectures, and queries in this study are executed on a system equipped with an NVIDIA RTX 4000 Ada 20 GB and 128GB of system RAM.

As we have seen earlier, the TALD is semantically segmented using Pointnet++ architecture. It consists of 50 training files and 30 testing files each covering an area $100 m^2$. To analyze the effect of spatial partitioning, two block sizes are considered. For a block size of 50 ($B_s = 50$), each file is divided into 4 blocks, and for a block size of 25 ($B_s = 25$), each file is divided into 8 blocks. From each block, 8192 points are uniformly sampled and used as input to the network. Only the coordinates X, Y, and Z ($N \times 3$) are utilized for semantic segmentation, without incorporating additional features. The model is trained for 100 epochs using Adam optimizer with a learning rate of 0.0005. The performance metrics for semantic segmentation of

the TALD for both block sizes are given in Table 4 and Table 5. After importing the spatio-semantic database into the PostgreSQL, the embedding vectors are indexed using IVFFlat based indexing with 1000 inverted lists for fast query retrieval. The L2-distance metric is used to measure the similarity between embeddings.

3.2 Spatial Query

Query 1 demonstrates the efficiency and practical value of integrating road vector data with LiDAR point clouds for infrastructure assessment or utility management. The 3D view of the combination of these two data is shown in Figure 6. Ver-

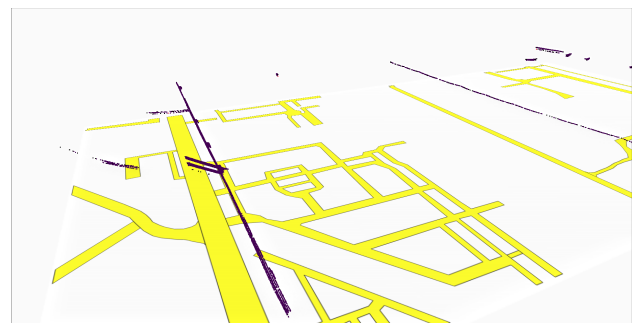


Figure 6. DALES powerlines and intersecting roads

tical clearance refers to the height of an object above the underlying terrain or infrastructure. In the constructed database environment, it becomes possible to compute the minimum vertical clearance of powerlines at their intersections with the roads, as demonstrated in Figure 7. The red points indicate the locations where the minimum vertical clearance is observed between the powerline clusters and the intersecting road segments. This query illustrates the efficiency of spatial indexing for fast queries and the advantage of querying on high-resolution 3D data, and aids in real-world utility management. Query 2 locates trees that lie in proximity to buildings. In a region like Thiruvananthapuram, where forest areas coexist with urban infrastructure, such proximity analysis is essential for disaster preparedness and vegetation management. Query 2 is formed in a way to get the intersecting bounding boxes of buildings and trees. The 3D view of the building bounding box is shown in Figure 8. A significant limitation of aerial LiDAR point clouds of TALD is the loss or occlusion of points below buildings, which restricts ground-level visibility and limits the accuracy of the queries performed. The result of query 2 is shown in Figure 9

3.3 Spatio-Semantic Query

The third query incorporates both spatial reasoning and semantic similarity search using the embedding vector concepts.

Database Type	Importing time	Storage size (.csv)	Storage size (.laz)
	(min)	(GB)	(MB)
Spatial database (TALD)	2.5	0.59	105
Spatial database (DALES)	4	0.98	130
Spatio-semantic database	28	4.44	788

Table 3. Summary of point cloud imported into the database

	Global Accuracy	Mean Accuracy	Mean IoU
	(%)	(%)	(%)
Bs = 50	83.06	77.06	63.78
Bs = 25	85.53	81.32	68.79

Table 4. Performance metrics of PointNet++ on TALD

Class	IoU (Bs = 50)	IoU (Bs = 25)
	(%)	(%)
Ground	68.07	72.95
Shrubs	36.42	46.91
Trees	82.38	84.14
Buildings	68.25	71.14

Table 5. Class-wise performance metrics of PointNet++ on TALD



Figure 9. Trees nearby buildings

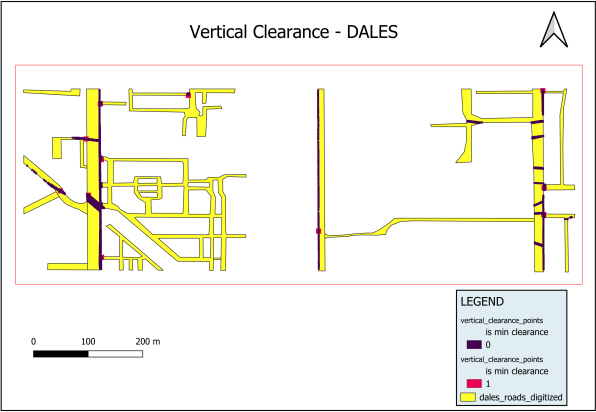


Figure 7. Vertical Clearance

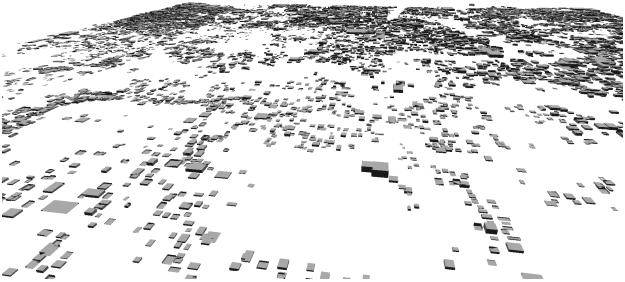


Figure 8. 3D building boundingboxes

In regions like Kerala, where coconut trees form an major component of the landscape and economy, accurately identifying them enables effective plantation mapping, land-use planning, and sustainable resource management. The embedding vector associated with each point (X, Y, Z) consists of the geometric features of those points relative to the nearby points. Unlike class labels, which provide only coarse semantic categories, embedding vectors capture fine-grained geometric signatures, allowing similar objects to remain close in the embedding space

even when they are spatially distant. By identifying the nearest embedding vector for a given coconut-tree cluster, the database can retrieve other clusters that exhibit similar semantic characteristics within the embedding space. Using this property, query 3 as shown in Figure 10, retrieves coconut tree clusters that are both semantically similar and spatially near within 50m buffer. The accuracy of these spatio-semantic queries relies heavily on the quality of the embedding vectors, which is directly dependent on the deep learning architecture and training strategy, as well as the clustering parameters defined to group points into individual objects. As we can see in Table 4, the mean IoU differs 15-20% from the global accuracy. The Mean IoU plays a key role in determining the accuracy of the queries. In datasets where the % of class contribution is very less, class imbalance may lead to very less or nearly 0% per-class IoU, there queries cannot be executed for the particular class. Despite these limitations, even though the TALD dataset only contains 4 class semantic labels, with this technique species level distinctions can be addressed, highlighting the advantage over instance segmentation. For large-scale geodatabases, where dynamic querying and semantic retrieval are needed beyond predefined instance level predictions, this allows for flexible, semantic similarity searches that are independent of the fixed instance labels. Extending this capability, Query 4 applies the same embedding space reasoning to detect anomalous clusters/objects whose embeddings are significantly different from the expected patterns of their assigned class. Figure 11 highlights a building cluster that contains points misclassified as trees. The detected anomaly points are visualised in Figure 12, which overlap the building and tree points. This demonstrates that embedding-based semantic queries enable the detection of mislabelled points, unexpected object shapes, and clusters that deviate from normal patterns. This strengthens data quality checks and enhances the reliability of large-scale LiDAR analytics.

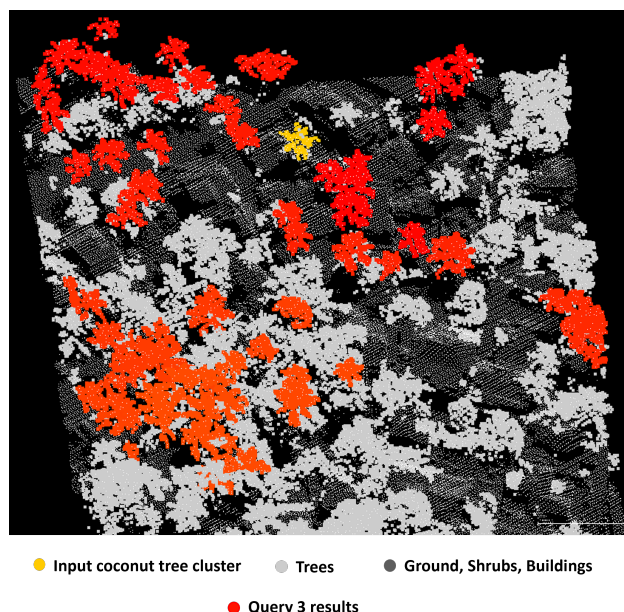


Figure 10. Coconut trees in 50m buffer of a input coconut tree cluster

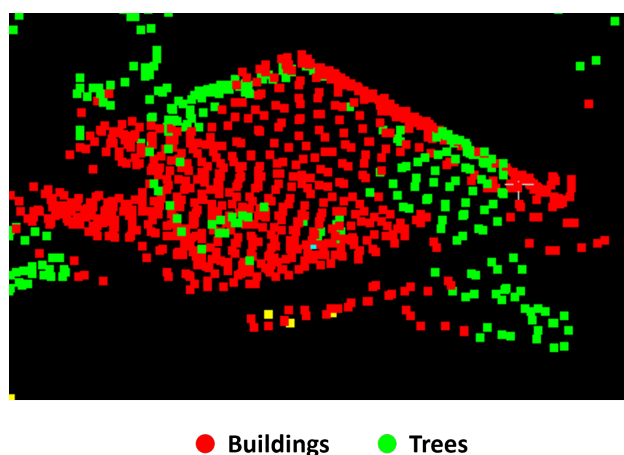


Figure 11. Misclassified building clusters

4. Conclusion

This paper tackles the challenge of insufficient semantic representation in LiDAR data and introduces an approach to resolve it. It discusses efficient query retrieval for massive point cloud data. The queries demonstrated show the capability in tree species identification, individual tree detection, utility management, anomaly detection, and can be extended to LiDAR Target detection. Because the database is built to handle generic LiDAR point clouds, the types of queries that can be executed are determined by the class categories available in the dataset. A few spatio-semantic queries, such as identifying flat or steep terrain, finding different types of roofs and their counts, can be achieved in the TALD spatio-semantic database. The presented spatio-semantic database environment can also be seamlessly integrated with pgPointCloud to accelerate query execution on large-scale LiDAR datasets, making it a flexible and scalable solution for advanced LiDAR analytics.

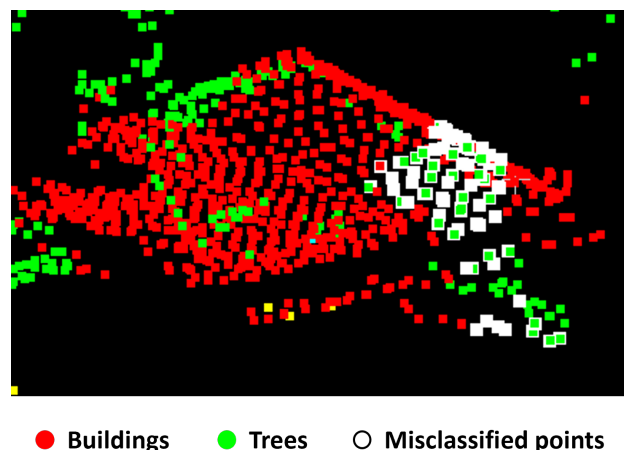


Figure 12. Detected anomalous clusters

References

- Béjar-Martos, J. A., Rueda-Ruiz, A. J., Ogayar-Anguita, C. J., Segura-Sánchez, R. J., López-Ruiz, A., 2022. Strategies for the Storage of Large LiDAR Datasets—A Performance Comparison. *Remote Sens.*, 14(11), 2623.
- El-Mahgary, S., Virtanen, J.-P., Hyypä, H., 2020. A Simple Semantic-Based Data Storage Layout for Querying Point Clouds. *ISPRS Int. J. Geo-Inf.*, 9(2).
- Ikawa, G., Watanabe, Y., Yamada, S., Takada, H., 2019. Performance Evaluation of Querying Point Cloud in RDBMS. *Proc. IEEE Int. Conf. on Big Data and Smart Computing (Big-Comp)*, Kyoto, Japan, 1–4.
- Jia, S., Gong, X., Liu, F., Ma, L., 2025. AI-Powered LiDAR Point Cloud Understanding and Processing: An Updated Survey. *IEEE Transactions on Intelligent Transportation Systems*.
- Lokugam Hewage, C. N., Laefer, D. F., Vo, A.-V., Le-Khac, N.-A., Bertolotto, M., 2022. Scalability and Performance of LiDAR Point Cloud Data Management Systems: A State-of-the-Art Review. *Remote Sens.*, 14(20).
- Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 5099–5108. Part of NeurIPS 2017.
- Sadineni, P. K., 2020. Comparative study on query processing and indexing techniques in big data. *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers Inc., 933–939.
- Schön, B., Bertolotto, M., Laefer, D. F., Morrish, S. W., 2009. Storage, Manipulation, and Visualization of LiDAR Data. *Proc. 3rd ISPRS International Workshop 3D-ARCH: 3D Virtual Reconstruction and Visualization of Complex Architectures*, XXXVIII Number 5/W1, 8–15.
- Varney, N., Asari, V. K., Graehling, Q., 2020. DALES: A Large-Scale Aerial LiDAR Data Set for Semantic Segmentation. *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops*, Seattle, WA, USA, 186–187.

Vijaywargiya, J., Ramiya, A. M., 2025. Thiruvananthapuram Aerial LiDAR Dataset (TALD): A Benchmark for Complex Urban Point Cloud. *IEEE Access*, 13, 42350–42363.