SHORT-PAPER

# Demonstrating MAST: An Efficient System for Point Cloud Data Analytics

**JIANGNENG LI**, Nanyang Technological University, Singapore City, Singapore

**HAITAO YUAN**, Nanyang Technological University, Singapore City, Singapore

**JIE WANG**, Nanyang Technological University, Singapore City, Singapore

**ZITING WANG**, Nanyang Technological University, Singapore City, Singapore

**HAN MAO KIAH**, Nanyang Technological University, Singapore City, Singapore

**GAO CONG**, Nanyang Technological University, Singapore City, Singapore



**Published:** 22 June 2025

**Citation in BibTeX format**

SIGMOD/PODS '25: International Conference on Management of Data
*June 22 - 27, 2025*
*Berlin, Germany*

**Conference Sponsors:**
SIGMOD

SIGMOD/PODS '25: Companion of the 2025 International Conference on Management of Data (June 2025)
https://doi.org/10.1145/3722212.3725099
ISBN: 9798400715648

# Demonstrating MAST: An Efficient System for Point Cloud Data Analytics

Jiangneng Li
jiangnen002@e.ntu.edu.sg
Nanyang Technological University
Singapore

Haitao Yuan
haitao.yuan@ntu.edu.sg
Nanyang Technological University
Singapore

Jie Wang
jie013@e.ntu.edu.sg
Nanyang Technological University
Singapore

Ziting Wang
ziting.wang@ntu.edu.sg
Nanyang Technological University
Singapore

Han Mao Kiah
hmkiah,gaocong@ntu.edu.sg
Nanyang Technological University
Singapore

Gao Cong
gaocong@ntu.edu.sg
Nanyang Technological University
Singapore

## Abstract

The rapid growth of point cloud (PC) data calls for specialized solutions for data management and analytics. In this paper, we introduce MAST [3], a prototype system designed for efficient PC data analysis. MAST enables users to perform analytical queries efficiently, incorporating semantic and spatial predicates while ensuring high query accuracy. In this demonstration, we introduce the framework of the system prototype, including the storage layer, preprocessing layer, query processing layer, visualization and user feedback layer. We introduce the user interface of MAST and its workflow, and showcase the end-to-end usage of the MAST system with 2 analytical queries executed on a real PC dataset.

## CCS Concepts

• **Information systems → Database query processing**.

## Keywords

Point cloud data query processing, DB4AI

**ACM Reference Format:**
Jiangneng Li, Haitao Yuan, Jie Wang, Ziting Wang, Han Mao Kiah, and Gao Cong. 2025. Demonstrating MAST: An Efficient System for Point Cloud Data Analytics. In *Companion of the 2025 International Conference on Management of Data (SIGMOD-Companion '25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3722212.3725099

## 1 Introduction

With the development in 3D scanning technology and its widespread application in fields such as autonomous driving and robotics, the volume of Point Cloud (PC) data [5] has grown significantly. PC data is defined as a set of 3-dimensional points $\mathsf{P} = \{(x, y, z) \in \mathbb{R}^3\}$ that capture environmental contexts. The inherent spatial features in PC data enable the extraction of detailed 3D semantic information, such as the precise localization of objects. The increased volume and rich information of PC data calls for specialized management and analytics.


*SIGMOD-Companion '25, June 22–27, 2025, Berlin, Germany*

ACM ISBN 979-8-4007-1564-8/2025/06
https://doi.org/10.1145/3722212.3725099

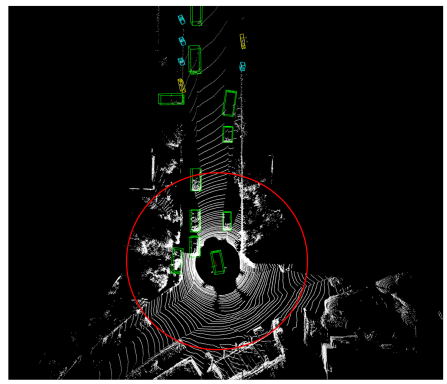


**Figure 1: A visualization of querying on PC data with objects detected, where the bounding boxes indicate objects, and the red circle indicates a spatial range query filter.**

Current research on PC data has primarily focused on enhancing the accuracy of deep models in tasks such as object detection and semantic segmentation, which often leverage deep learning techniques like deep neural networks. As shown in Fig. 1, querying PC data involves not only retrieving semantic objects (e.g., vehicles, pedestrians) identified through detection methods but also evaluating spatial predicates. However, to the best of our knowledge, no system supports such PC queries containing predicates on semantic information extracted using deep learning models. Current database systems that support the PC data type, such as PostgreSQL pgPointcloud [4] and Ganos [6], treat the PC data as typical spatial objects to support queries with spatial predicates (e.g., retrieving 3D points within a query region), but not queries with semantic predicates.

**MAST.** To bridge the gap between analysis requirements and current PC database management systems, we develop the PC data analytical system named MAST [3]. MAST is the first system that efficiently applies deep models to conduct analytical queries on PC data, especially on queries involving both semantic and spatial predicates. Examples of such queries can be found in [3]. Particularly, it selects only a fraction of PC frames to be processed by the deep model, while providing accurate query results enhanced by the dedicated spatio-temporal analysis.

In autonomous driving and robotics scenarios, PC data typically appear as continuous sequences of PC frames. To leverage this spatio-temporal characteristic, MAST incorporates a multi-layered architecture. It consists of a storage layer that stores and manages PC data, a sampling and preprocessing layer that samples a fraction of PC data and stores the information extracted by deep models, a query processing layer that generates approximate query results based on preprocessed and evaluated information, and a visualization & feedback layer designed to achieve higher query accuracy.

In MAST, PC frames are stored based on their timestamps. To optimize disk usage, MAST detects significant spatio-temporal redundancy among neighboring PC frames. It then compresses and stores the redundant PC frames together and removes the spatio-temporal redundancy for additional disk saving.

For query processing, we split the procedure into sampling stage and query processing stage, which reduces the need for deep model usage by processing only a subset of the PC data, thereby saving GPU resources. By leveraging the inherent spatio-temporal features in PC data, the method generates approximate query results for multiple queries based on the sampled and processed PC data.

Our method enables PC frame sampling for multiple queries without relying on the query workload as the input during the sampling process. It has the advantage of the generality of model selection and the stable efficiency improvement proportional to the sampling budget. A key challenge on PC frame sampling step is optimizing the querying performance with spatio-temporal information utilized, as different data points may vary in their significance and contribution to the overall query efficiency. To tackle this, we introduce a hierarchical multi-agent framework that hierarchically selects PC frames. Considering that PC data sequences inherently exhibit spatial and temporal characteristics, we design a reward based on the variance between the newly sampled PC and the predicted object mobility situation (i.e., location and velocity of objects) derived through spatio-temporal analysis. This approach aims to ensure that the sampled PC frames are of maximal importance.

Once the query results are produced, MAST visualizes them and provides users with an interactive way to enhance query performance. Specifically, users can provide feedback if the results are deemed incorrect based on the visualization. MAST then re-analyzes the query and generates results with a higher accuracy.

In this paper, we introduce the main framework of MAST and demonstrate its functionalities through end-to-end analytical query processing cases.

## 2 System Overview

We proceed to introduce the system built based on [3], which consists of four main layers, namely PC data storage layer (Section 2.1), sampling and preprocessing layer (Section 2.2), query processing layer (Section 2.3), and visualization & feedback layer (Section 2.4). The framework of our system is given in Fig. 2.

**Workflow.** When a sequence of PC frames arrives at the system, MAST first compresses the PC sequence and deduplicates the redundancy of PC data to save disk space. Next, the sampling layer performs an importance sampling on the PC frames and conducts spatio-temporal analysis on the PC sequence, resulting in a preliminary object movement prediction. During query processing, the system utilizes the pre-analyzed predictions from the sampling and preprocessing layer and produces approximate query results. Users can interact with system by checking the visualization and giving feedback.

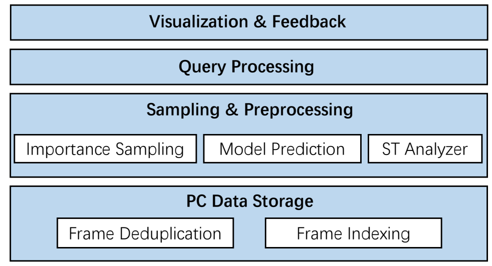



**Figure 2: Overview of the MAST system.**

### 2.1 PC data storage

**Storage optimization.** Storing PC frames can consume significant disk space. Existing systems, such as pgPointcloud [4], support to store PC data in LAZ format [2], which is also supported by MAST. Additionally, MAST supports joint storage of continuous PC frames to save additional disk space. Specifically, MAST detects whether the continuous PC frames contain a large portion of redundant 3D spatial points. To achieve spatio-temporal redundancy detection, MAST splits the spatial domain of each PC frame and partition the 3D spatial points into voxels. MAST then measures the similarity of each voxel across continuous PC frames within the same spatial partition. Redundant voxels in continuous PC frames are then merged.

**Indexing.** During the storage procedure, an octree-based index is constructed and stored alongside the encoded PC frames as part of the PC sequence storage process. This index is utilized for both decoding the PC frames and efficiently indexing the 3D spatial points for processing queries with spatial range filter.

### 2.2 Sampling

**Importance sampling.** Since the query includes querying on semantic objects within PC frames, deep models are involved for semantic information prediction. Processing all frames in a PC sequence can be costly. Therefore, to conduct efficient queries, MAST selects a fraction of PC frames out of the PC sequence to be preprocessed for the following query execution. This raises a vital problem for MAST to detect important PC frames, i.e., which PC frames mostly affect the query performance. Addressing this problem requires that the sampling method analyze the relation between the features of PC frames and the query performance.

To resolve this, we design the *Sampling Module*, which provides a sampling method that analyzes the spatio-temporal features of PC frames and selects PC frames based on the analysis. We develop a multi-agent sampling policy, and carefully design a reward. This reward reflects the difference between the real mobility information extracted from the sampled PC frame by the deep model and the estimated mobility information predicted by the mobility estimating module, named *ST-PC Analysis Module.* Therefore, MAST selects PC frames whose mobility situations differ the most from the estimated situations predicted by the currently sampled PC frames, which results in more accurate query results. Detailed sampling algorithm can be found in [3].

## 2.3 Query Processing

**Spatio-temporal analysis for query processing.** With sampled PC frames and the information extracted by the deep model, generating accurate query results becomes a vital issue. Current methods, such as linear prediction (direct prediction based on the object number changes), do not consider the mobility situation of PC frames, and thus result in sub-optimal query performance on queries. To address this issue, we propose predicting query results utilizing the spatio-temporal (ST) continuity of PC frames. Our method simulates the real object movement, and achieves outstanding performance on multiple query types. Furthermore, processing queries with ST-analysis will take more computation. To accelerate the query processing, we pre-compute and store the mobility prediction, thus preventing repetitive computation and achieving computation cost reduction. For the retrieval queries, if the specific query requires low false positive, the system supports the deep model to do the filtering on the results with low confidence.

**PC analytical queries.** MAST supports different kinds of queries including retrieval queries which return PC frames satisfying the query predicates, and aggregate queries which aggregate the information of the input PC sequence. MAST supports query processing for analysis within a single PC frame, namely single-frame query (see the demo Q1), and queries that analyze across multiple PC frames, namely multi-frame query (see the demo Q2). The queries involve semantic predicate and spatial predicate, defined as follows:

**Semantic predicate on PC data.** MAST supports constructing the semantic predicate that filters on the semantic objects. The predicate can filter the object type (e.g., car or human) and the number of objects (e.g., `car >= 3`). In a multi-frame query, the query can add an object motion filter (e.g., the car is moving closer to the auto-vehicle).

**Spatial predicate on PC data.** The system supports different spatial predicates on the extracted semantic objects. Besides the spatial predicates (e.g., the range filter with the location of the LiDAR sensor as the center point) introduced in [3], the system also supports spatial predicates that consider the 3D shape, which computes the closest distance from the surface of one object to the surface of another object, namely face-to-face distance.

## 2.4 Visualization & Feedback

The visualization and feedback layer allows users to actively check whether the system provides correct query results and feedback to the system if the result needs to be improved.

**Visualization.** MAST visualizes the query results after a retrieval query is processed. If the query includes a spatial predicate, the visualization allows to visualize only the spatial points of a PC frame within the spatial predicate. The index constructed during the storage procedure is leveraged to fast load the PC frames.

**Feedback.** The user can provide feedback to the system when the system produces incorrect query results. If the user finds the visualized frame does not meet the requirement of predicates, they can feedback that the selected frame is not accurate. After the feedback, the system will re-analyze the query. Specifically, the mis-predicted PC frames will be input to the deep model for exact prediction. Then, the ST-analyzer regenerates the motion prediction to get a more accurate result.

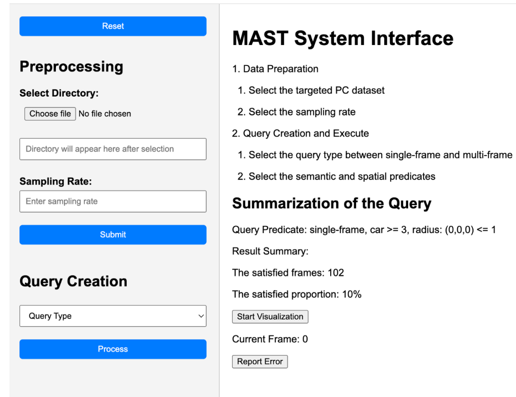



**Figure 3: User Interface of MAST.**

# 3 Demonstration

We demonstrate MAST with an end-to-end usage example. During the demonstration, we apply a real-world dataset named Semantic-KITTI [1], which is a widely used Point Cloud sequence dataset. We demonstrate the functionality of MAST with the following two queries:

**Q1.** Identify the PC frames in which more than three vehicles are within a face-to-face distance of less than one meter from the autonomous vehicle.

**Q2.** Identify the PC frame segments where vehicles are positioned behind the autonomous vehicle and are moving closer to it.

Here, Q1 includes the semantic interception within a single PC frame, and Q2 is to analyze across multiple frames to understand the objects' movement. The interface of MAST is shown in Fig. 3. We then introduce the usage of the interface with an example.

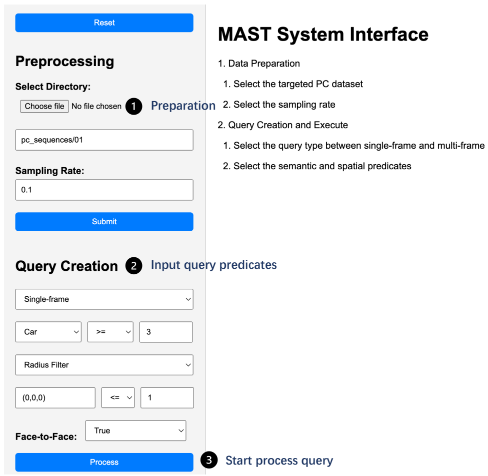
## 3.1 End-to-end process of Q1

Assume a user named Alice would like to evaluate whether the monitored vehicle is driving dangerously. Therefore, Alice applies MAST with the input of the PC data generated by the sensor. She would like to determine the proportion of point cloud (PC) frames in which more than three vehicles are located within a distance of less than one meter, relative to the entire PC sequence.

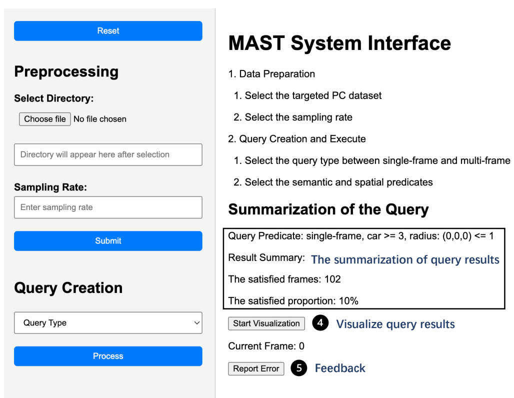
• **Step 1. Preparation.** Alice first inputs the data into the system for pre-sampling and preprocessing. The input data is compressed before the query is executed. In order to save GPU resources during the query execution, Alice limits the model usage budget to 0.1 (i.e., 10% of the whole PC frames will be processed by the object detection deep model) (As shown in Fig. 4a). The system then preprocesses the data based on the budget. After preparation, the data is ready to be queried.

• **Step 2. Query Creation.** Then Alice creates the targeted query in order to conduct analysis. First, Alice selects the single-frame query type to achieve analysis within one PC frame. She selects the semantic object type as Car and set the filter as `Car >= 3`. She then selects the range query and set the face-to-face option as `True`, and set the filter as `dist <= 1` (As shown in Fig. 4a).

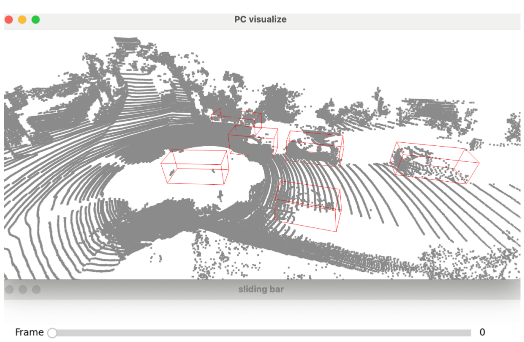
• **Step 3. Query Execution.** The aggregation results will be provided by default after Alice clicks the 'Process' button, i.e., a summary of the query results will be generated.




**(a) Preprocessing and query execution steps.**




**(b) Query result summary and visualization steps.**




**(c) Result visualization example.**

**Figure 4: The end-to-end usage of MAST to process Q1. Each subfigure represents a part of the query execution procedure.**

• **Step 4. Result Visualization.** Alice selects to start the visualization to see the satisfied PC frames. Alice chooses between 'partial' and 'fully' to decide whether to partially or fully visualize the PC frames. If partial visualization is selected, then only the 3D spatial points within the spatial predicate will be visualized.

• **Step 5. Result checking and Feedback.** After the visualization procedure is done, Alice can select whether the visualized PC frame satisfies the predicates. If Alice detects a false positive, the user can click 'Report Error' button to feedback the incorrect query result (As shown in Fig. 4b).

• **Step 6. Re-prediction.** If a feedback is conducted, the system will re-analyze the PC data, after the ST-analyzing procedure is conducted again, the query will be processed again and provide new results accordingly.

## 3.2 End-to-end process of Q2

Furthermore, Alice would also like to detect if there exists a collision risk during the driving procedure. She wants to detect if there are events where there is a vehicle getting closer to the back of the auto-vehicle. Since the query preprocessing has already been conducted for Q1, Q2 can be conducted based on the previously extracted information. Different from Q1, Q2 is a multi-frame query that needs to jointly analyze multiple neighboring PC frames and thus requires a different query creation step.

**Query Creation.** Alice selects the multi-frame query type, and the semantic object filter is set as `Car >= 1`. Different from Q1, Alice sets a semantic filter named velocity filter as '> `0`', indicating the objects are moving closer to the vehicle. And the spatial filter is set as 'Domain filter' and predicate value is set as 2, selected from (`0`, 1, 2, 3), restricting the car to the area behind the auto vehicle.

**Query Execution.** Alice clicks the 'Run' button to execute the query. The system will return the summary on how many PC frame segments satisfy the requirement (default segment length is set as 10 frames).

**Visualization.** The visualization of such query allows Alice to check between different PC frame segments. The feedback mechanism is also allowed here.

## 4 Conclusion

In this demonstration, we present MAST, a system prototype designed for efficient point cloud management and analytics. MAST facilitates the analysis of semantic events captured in PC data and supports resource-efficient query processing. Through visualization and user feedback, MAST enables interactive user experiences and delivers more accurate query results.

## References


[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. 2019. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 9296–9306. https://doi.org/10.1109/ICCV.2019.00939

[2] Martin Isenburg. 2013. LASzip: lossless compression of LiDAR data. *Photogrammetric engineering and remote sensing* 79, 2 (2013), 209–217.

[3] Jiangneng Li, Haitao Yuan, Gao Cong, Han Mao Kiah, and Shuhao Zhang. 2025. MAST: Towards Efficient Analytical Query Processing on Point Cloud Data. *Proc. ACM Manag. Data* 3, 1 (2025), 52:1–52:27. https://doi.org/10.1145/3709702

[4] n.d. 2025. PostgreSQL pgPointcloud. https://pgpointcloud.github.io/pointcloud/. [Online; accessed 17-Mar-2025].

[5] Radu Bogdan Rusu and Steve Cousins. 2011. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*. IEEE. https://doi.org/10.1109/ICRA.2011.5980567

[6] Jiong Xie, Zhen Chen, Jianwei Liu, Fang Wang, Feifei Li, Zhida Chen, Yinpei Liu, Songlu Cai, Zhenhua Fan, Fei Xiao, and Yue Chen. 2022. Ganos: A Multidimensional, Dynamic, and Scene-Oriented Cloud-Native Spatial Database Engine. *Proc. VLDB Endow.* 15, 12 (2022), 3483–3495. https://doi.org/10.14778/3554821.3554838