Library Progress International
Vol.44 No.2, July-December 2024: P.631-638

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Performance Analysis of Point Cloud Preprocessing Algorithms Suitable for Construction Progress Analysis

You-Kyung Kim[1] and Seok-Heon Yun[*2]

[1]Master's Student, Department of Architecture Engineering, Gyeongsang National University, 501 Jinjudaero, Jinju-si, Gyeongsangnam-do, 52828, South Korea

[*2]Professor, Department of Architecture Engineering, Gyeongsang National University, 501 Jinjudaero, Jinju-si, Gyeongsangnam-do, 52828, South Korea

yukyung1229@gnu.ac.kr[1] and gfyun.@gnu.ac.kr[*2]

How to cite this article: You-Kyung Kim, Seok-Heon Yun (2024). Performance Analysis of Point Cloud Preprocessing Algorithms Suitable for Construction Progress Analysis *Library Progress International,* 44(2), 631-638.

ABSTRACT

As the scale and complexity of construction projects increase, the importance of progress management becomes even more significant. The current construction management systems have limitations in accurately analyzing project delays and cost overruns. To achieve precise analysis of construction site conditions, various studies are being conducted to collect and analyze point clouds from construction sites. The point cloud data collected from construction sites are composed of a vast amount of information. For effective analysis of this data, it is essential to use appropriate preprocessing algorithms that can extract meaningful key data. To extract significant data from the entire dataset, various algorithms can be utilized. This study aims to compare and analyze the performance of three different point cloud preprocessing algorithms and propose the most suitable one for understanding the current status of construction sites. In the study, data on the exterior of the building was collected using drones, and point cloud coordinates were extracted. These coordinates were then processed using preprocessing algorithms, namely sor (statistical outlier removal), ror (radius outlier removal), and cor (conditional outlier removal). These algorithms were compared and analyzed based on accuracy, data processing time, and data preservation rate. The results of the study indicate that the sor algorithm exhibited the highest outlier removal rate in terms of point cloud accuracy. Additionally, the data processing took 2 seconds. In terms of data preservation, 99.6% of the key wall data was retained. The performance of preprocessing algorithms is a critical factor in the development of a progress management system utilizing point clouds; therefore, the sor algorithm is determined to be the most suitable point cloud preprocessing algorithm.

Keywords: Point Cloud, Preprocessing Algorithm, Sor, Ror, Cor

1. INTRODUCTION

In construction projects, the project's status must be analyzed using essential information such as progress monitoring, predicting potential issues, and providing countermeasures for plan changes [1]. If effective data analysis of the project status is lacking, issues such as project delays, cost overruns, impacts on subsequent projects, and public complaints can arise. However, on many construction sites, the data analysis required for project status assessment based on the Critical Path Method (CPM) is either not conducted or only manually performed upon the client's request. The lack of systematic data analysis can lead to various project risks. By utilizing smart manufacturing technologies, predictive analytics can be employed to identify and establish optimized processes, leading to increased efficiency and higher production output [2]. Therefore, it is essential to proceed with the digitalization of progress management to establish a systematic progress management system.

Recently, the construction industry has seen an increasing effort to enhance productivity by integrating smart technologies such as BIM (Building Information Modeling), 3D scanning, digital fabrication, and digital twins [3]. Research on the use of point clouds in the planning and construction phases of architectural projects is also actively being conducted [4, 5]. During the extraction process of point cloud data, significant data dispersion and noise can occur, and data loss may happen in areas not captured during scanning [6]. Additionally, if appropriate algorithms are not applied during the preprocessing phase, further information loss and data accuracy issues may arise. Therefore, proper preprocessing of point cloud coordinates is necessary after collecting 3D data.

This study aims to apply three different preprocessing algorithms to point cloud coordinates and compare the results of each algorithm. Specifically, the study analyzes the performance of these algorithms based on three metrics: accuracy, data processing time, and data preservation rate. The goal is to propose the most suitable point cloud coordinate preprocessing algorithm for construction site status analysis.

The research was conducted on a building at G University, located in South Korea, where point cloud data was collected and the performance of preprocessing algorithms was analyzed. To enable a detailed comparison of the preprocessing algorithms, the study was limited to one exterior wall on the protruding sections of the 4th and 5th floors of the building. The methodology of the study is as follows: First, three algorithms to be applied in this research were selected through a review and analysis of prior studies related to point clouds. Second, photographic data of the target building was collected using drone equipment, and this data was then extracted into point cloud coordinates. Third, the three preprocessing algorithms were applied to the point cloud coordinates to extract key data. Finally, the results of the processed point cloud data were compared and analyzed.

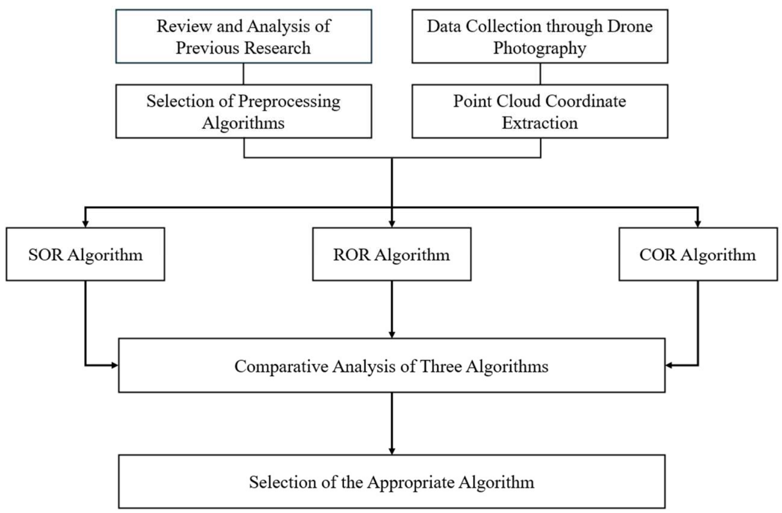


Figure 1. *Research Flowchart*

2. LITERATURE REVIEW

Currently, research utilizing point clouds is actively being conducted across various industries. The following are examples of studies related to point cloud preprocessing.

Wang [7] proposed a point cloud data preprocessing algorithm for obstacles to address issues in obstacle detection algorithms for intelligent vehicles. The study first introduced a 3D point cloud data filtering algorithm to remove noise and the vehicle body, followed by a point cloud down-sampling method. Through this research, problems such as the difficulty in distinguishing nearby obstacles due to a fixed threshold and the separation of obstacles into multiple parts at a distance were resolved, enhancing real-time performance.

Lee [8] proposed a method to efficiently process and manage the compression of dynamic point clouds stored in animation information within continuous 3D point clouds. To address the complexity of point clouds containing large amounts of information, the study introduced a voxel-based point reconstruction method and a preprocessing process that generates compressed point clouds. As a result, redundant information between frames was efficiently extracted and compressed, simplifying the format. The proposed algorithm resulted in an average file size reduction of 40%, optimizing data transmission and storage.

Kim [9] proposed a method for generating clustered solid building models based on point cloud data. To create clustered solid building models, a RANSAC (Random Sample Consensus) based model fitting algorithm was utilized in the preprocessing stage. This algorithm was employed to remove ground points and simplify building boundary points using line fitting. Additionally, the method of converting point cloud orthographic projection images was used to extract building areas. The study confirmed that texturing of the 3D model is feasible. Based on the results of this study, it is expected that accuracy in the 3D model generation process after preprocessing can be improved.

Kang [10] proposed an automated reverse engineering method for pipe shapes in large-scale MEP (Mechanical, Electrical, and Plumbing) point clouds. During the research process, a voxel-based statistical outlier removal technique was used for noise reduction in large-scale point clouds, simplifying the data and performing preprocessing. The prototype development and designed algorithms were implemented. The rendering performance was measured and verified, showing a standard deviation of 0.004, indicating minimal variation. The study concluded that this approach is suitable for processing large-scale data and overcomes inaccuracies in unit radius evaluation of existing point cloud density assessment technologies and limitations in ultra-low-density data.

Zhao [11] proposed a robust multi-task learning network consisting of three networks based on PointNet to preprocess complex 3D point cloud data collected by LiDAR. In the first stage, Denoising, unnecessary noise is removed based on the geometric characteristics of each point in the point cloud. The second stage, Single-object Segmentation, integrates a mechanism module to segment single objects according to the features of the point cloud. In the third stage, Completion, the FoldingNet network structure is employed to restore damaged or missing point clouds, aiming to generate complete 3D shapes.

Li [12] proposed a point cloud preprocessing method using an automated framework to remove noise and external data from laser-scanned datasets. In the first stage, a hybrid segmentation method is used to extract planar primitives from noisy points. In the second stage, nonlinear optimization is employed to determine the orientation of the building and realign the point cloud with the coordinate axes. In the third stage, the Graph-Cut algorithm is used to remove external points, preserving only the indoor points.

Kurup [13] presented the DSOR filter, an extension of the traditional SOR (Statistical Outlier Removal) algorithm, designed to remove outliers in LiDAR data collected in snowy conditions. The traditional SOR algorithm is ineffective at removing snow noise and does not account for the non-uniform distribution of point clouds. The DSOR filter improves upon this by dynamically adjusting the filtering threshold based on each point's distance from the sensor. The DSOR filter was shown to be 28% faster and 4% more accurate in recall than the DROR filter and is scalable to larger point clouds.

Ning [14] conducted research aimed at ensuring data quality by removing outliers from point clouds. The study proposed a simple yet effective method based on two geometric properties. The first method used local density characteristics to calculate the density of neighboring points around each point to remove isolated outliers. The second method employed local fitting plane characteristics to project points onto the plane, removing non-isolated outliers near the model surface. The study successfully removed complex outliers and demonstrated the effectiveness of using local density and local fitting planes for outlier removal.

Balta [15] proposed an effective method for processing large-scale outdoor 3D point clouds. The study aimed to quickly remove outliers from input data and downsample the data to save computational resources and memory for subsequent 3D modeling tasks. The method used Voxel-Subsampling to convert point clouds into uniform densities, enhancing computational efficiency. Following this, the Fast Cluster Statistical Outlier Removal (FCSOR) method was applied to remove outliers from large datasets. This algorithm was designed to function efficiently even with heterogeneous point clouds. Experimental results showed that the proposed FCSOR method outperformed existing methods in terms of computational speed and accuracy when evaluated on data collected from actual outdoor environments.

In this study, based on a review of prior research, examples of point cloud preprocessing algorithms developed across various industries were analyzed. The analysis of previous research indicates that there is active ongoing research aimed at improving point cloud preprocessing algorithms in various sectors. A comparison of these developed algorithms suggests that there is a need for developing preprocessing algorithms specifically suited for analyzing construction site progress. Therefore, this study aims to analyze the SOR (Statistical Outlier Removal) algorithm, the ROR (Radius Outlier Removal) algorithm, and the COR (Conditional Outlier Removal) algorithm, which integrates the SOR algorithm, to determine their suitability for progress analysis in construction projects.

3. Point Cloud Data Extraction

This study aims to compare preprocessing algorithms to improve the accuracy of point cloud coordinates. The process of extracting point cloud data for the comparison of preprocessing algorithms is carried out as follows. Point cloud data is collected using DJI's 'PHANTOM4' drone equipment. The data captured by the drone is then visualized using the 'Pix4Dmapper' program, and the point cloud extraction range is defined. The point cloud data within the scope of the study is then extracted in XYZ file format.

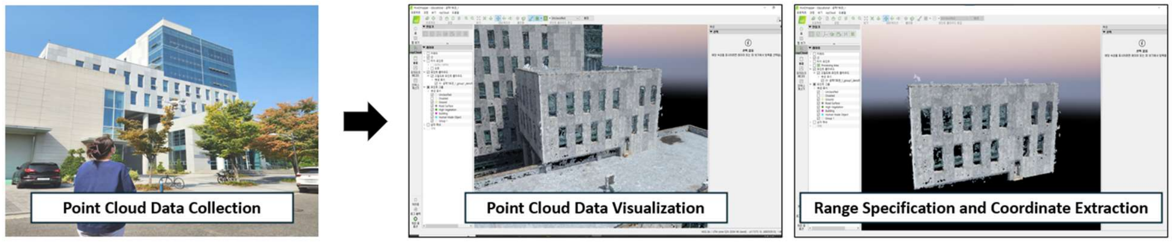


Figure 2. *Point Cloud Data Extraction Process*

To generate 3D object models from point clouds, it is essential to separate the object from non-object point clouds, such as the ground, and define the scope of the data [16]. In this study, the research scope for a detailed comparison of preprocessing algorithms was set to the protruding exterior walls of the 4th and 5th floors of the case study building.

4. Analysis of Preprocessing Algorithms

As previously explained, the raw point cloud data for applying preprocessing algorithms was collected in XYZ file format during the point cloud coordinate extraction process. The preprocessing algorithms to be applied to this data in this study are as follows.

4.1. *SOR (Statistical Outlier Removal)*

The SOR algorithm, or Statistical Outlier Removal, is a method that uses statistical techniques to remove outliers from point cloud coordinate data. The steps for applying the SOR algorithm are as follows: the average distance between neighboring coordinates is calculated for each point. Points that fall outside a statistically defined range are considered outliers and are removed. The SOR algorithm is well-suited for noisy data as it employs a statistical approach to outlier removal.

The SOR algorithm allows for balancing noise removal with data preservation by adjusting its parameters. For the purpose of preserving key point cloud data while removing noise, the parameters are set to nb_neighbors = 50 and std_ratio = 1.0.

The changes in point cloud outliers resulting from the application of the SOR algorithm are shown in Figure 3(b). When compared to the raw point cloud data shown in Figure 3(a), it is evident that the outliers distributed along the boundary of the external wall have been removed.

4.2. *ROR (Radius Outlier Removal)*

The ROR algorithm, or Radius Outlier Removal, is a method that removes points considered outliers if a certain number of neighboring points are not present within a specified radius. Since the ROR algorithm focuses on removing outliers based on radius, it is particularly effective at eliminating outliers in low-density areas.

The results of the ROR algorithm can vary depending on the radius setting. It is important to set the radius and threshold values considering the density of the region containing outliers. In this study, the parameters are set to min_neighbors = 10 and radius = 0.15.

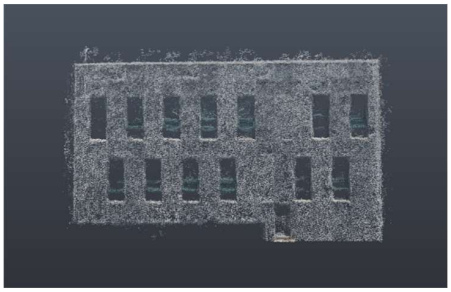
The changes in point cloud outliers resulting from the application of the ROR algorithm are shown in Figure 3(c). Compared to the figures of the raw point cloud data, the removal of outliers along the boundary of the external wall can be observed.

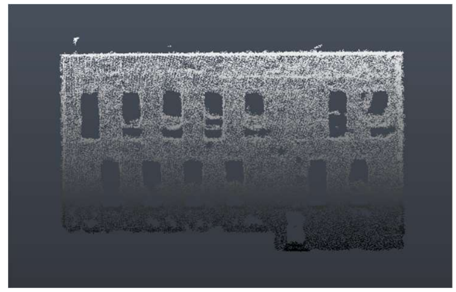
4.3. *COR (Conditional Outlier Removal)*

The COR algorithm, or Conditional Outlier Removal, is designed to remove outliers from point cloud data by setting specific conditions or by optimally aligning the point cloud to eliminate outliers. This algorithm is particularly useful for refining data in complex datasets by applying conditions based on specific patterns or structures. The steps for applying the COR algorithm are as follows: first, key features such as the density, surface slope, and distance distribution of the point cloud are identified. Then, specific conditions are set within the point cloud, and any points that do not meet these conditions are considered outliers and removed.

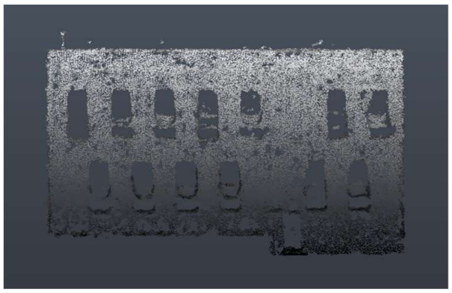
To determine the key features in the COR algorithm, adaptive density-based filtering is applied. In this study, the parameters are set to nb_neighbors = 20 and std_ratio = 2.0.

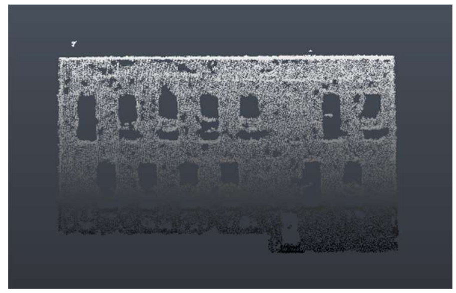
The changes in point cloud outliers resulting from the application of the COR algorithm are shown in Figure 3(d). Compared to the figures of the raw point cloud data, it is evident that outliers along the boundary of the external wall have been removed.





(a) *Point Cloud Raw Data* *(b)* *SOR Algorithm*





(c) *ROR Algorithm* *(d)* *COR Algorithm*

Figure 3. *Application of Preprocessing Algorithm*

5. RESULTS AND DISCUSSION

The performance of these point cloud coordinate preprocessing algorithms can be analyzed based on various criteria; however, in this study, the analysis is focused on three criteria: accuracy, data processing time, and data preservation rate.

5.1. *Accuracy Analysis (Outlier Removal Rate)*

Assessing the degree of outlier removal in point cloud preprocessing algorithms is the most critical factor. To analyse the accuracy of the three preprocessing algorithms, the number of outlier points removed by each algorithm is compared. The outlier removal rate for each algorithm is calculated based on the number of outlier points eliminated. Autodesk ReCap software is utilized for comparing the number of outlier points. Considering the study's scope and the density of point cloud coordinates, a single outlier point is defined as an area with 10 or more points.

Table 1: *Accuracy Result of Algorithms*

Elements / Algorithm	Outlier Locations	Removed Outlier Locations	Outlier Removal Rate (%)
Point Cloud Raw Data	151	-	-
SOR	20	131	86.75
ROR	34	117	77.48
COR	27	124	82.12

The results of the accuracy comparison for the point cloud preprocessing algorithms are presented in Table 1. The original point cloud raw data contained 151 outlier points. After applying the preprocessing algorithms, the number of outlier points was measured for each algorithm. The SOR algorithm reduced the outlier points to 20, removing 131 outlier points. The ROR algorithm and COR algorithm resulted in 34 and 27 outlier points, respectively.

The outlier removal rates for the preprocessing algorithms were 86.75% for the SOR algorithm, 77.48% for the ROR algorithm, and 82.12% for the COR algorithm. Based on the accuracy comparison analysis, the SOR algorithm appears to be the most suitable for outlier removal.

5.2. *Analysis of Data Processing Time*

As the size of the building increases, the amount of point cloud data also grows. As the data size increases, processing speed becomes a critical factor. To analyze the performance of appropriate preprocessing algorithms for application in a progress management system, data processing speed was compared. The analysis of data processing speed is based on the time required to process the data.

Table 2. *Result of Data Processing Time*

Algorithm	SOR	ROR	COR
Data Processing Time (s)	2	1.3	8

The data processing times for the point cloud preprocessing algorithms are presented in Table 2. The ROR algorithm was the fastest, requiring only 1.3 seconds. The SOR algorithm also performed similarly, with a processing time of 2 seconds. However, the COR algorithm took significantly longer, with a processing time of 8 seconds, which is more than four times the time required by the other algorithms.

Given the larger scale of building projects on construction sites, data processing speed is crucial in the development of a progress management system using point clouds. Therefore, while considering the development of such a system, the use of the COR algorithm should be carefully considered due to its slower processing time.

5.3. ***Analysis of Data Preservation Rates***

Overfitting in point cloud preprocessing can lead to the loss of critical data. In the development of a progress management system utilizing point clouds, the loss of key data can cause errors in subsequent alignment and coordinate comparison processes. Therefore, it is essential to verify the data preservation rate during the preprocessing stage. The data preservation rate is calculated by comparing the raw point cloud coordinates with the coordinates after the application of each preprocessing algorithm. The review of the point cloud preservation rate was conducted using two methods.

The first method calculates the preservation rate by comparing the number of points in the raw data with the number of points in the processed exterior wall. This method calculates the preservation rate by comparing the total number of points in the point cloud.

Table 3. *Preservation Rate Using the Point Cloud*

Elements / Algorithm	Number of Raw Data Points	Number of Processed Exterior Wall Points	Data Preservation Rate
SOR	91,039	83,332	91.54
ROR	91,039	79,521	87.36
COR	91,039	78,270	85.97

The results of the method for calculating data preservation rate using the entire point cloud are shown in Table 3. The analysis revealed that the SOR algorithm had the highest data preservation rate at 91.54%, followed by the ROR and COR algorithms with preservation rates of 87.36% and 85.97%, respectively. While this method for comparing data preservation rates is simple to calculate, it has the drawback of including unprocessed residual outliers in the preserved data count. As this method does not clearly define the number of points maintaining the building's shape, it is considered a less accurate comparison approach.

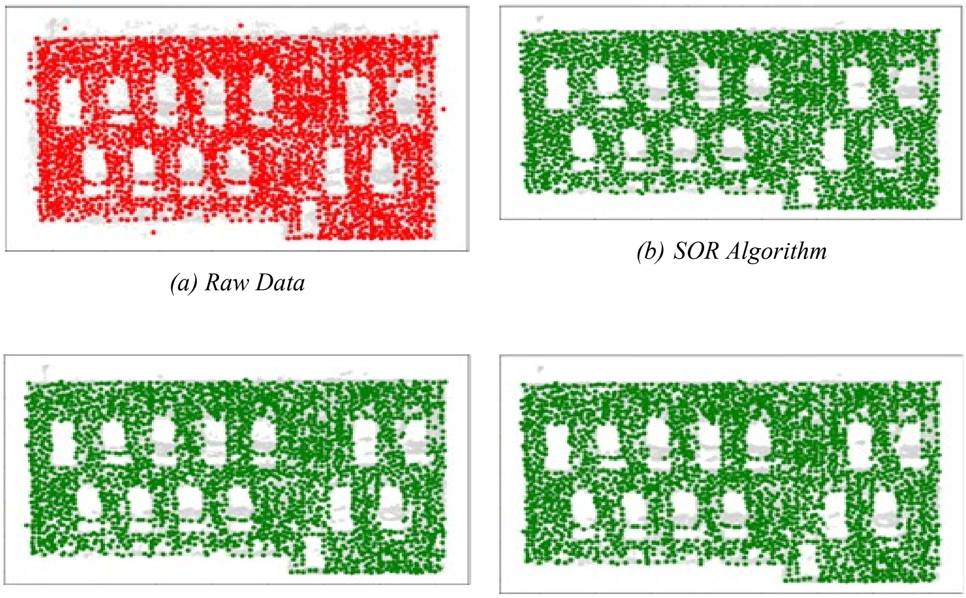
When calculating the data preservation rate by comparing the total number of points in the point cloud, the issue arose that residual outliers were included in the result. Therefore, an additional method was employed to calculate the data preservation rate by excluding residual outliers after the application of preprocessing algorithms. This method involves using a feature extraction algorithm to extract and compare feature points that maintain the shape of the building's exterior wall.

The second method involves calculating the data preservation rate by extracting feature points that maintain the shape of the exterior wall in the point cloud data. The process for calculating the data preservation rate is as follows: First, the k-means algorithm is applied to extract exterior wall feature points from the raw point cloud data. Second, the same feature points are identified in the preprocessed data. Third, the number of feature points is compared to determine the extent of data loss, and the preservation rate of key data is calculated.

In the process of extracting exterior wall feature points using the k-means algorithm, the exterior wall area is defined, and feature points are extracted. For the comparison of data preservation rates, the number of extracted feature points is set to 2,500.

The data preservation rates for the point cloud preprocessing algorithms are shown in Figure 4. The feature points extracted from the raw data were consistently applied to the data processed by each algorithm. The results of the feature extraction and data preservation rate calculations showed that the SOR algorithm achieved a rate of 99.6%. The ROR algorithm had a preservation rate of 98.68%, while the COR algorithm achieved 98.32%. These results suggest that the SOR algorithm is the most suitable for preserving data during the preprocessing stage.

The high data preservation rates observed across all algorithms are likely due to the study's focus on a single wall. As the scope of analysis decreases, the ratio of extracted feature points to the total point cloud increases, leading to a higher overall data preservation rate



(a) Raw Data *(b) SOR Algorithm*

(a) ROR Algorithm (b) COR Algorithm

Figure 4. *Feature Extraction Results*

6. CONCLUSION

The purpose of this study was to analyse point cloud data preprocessing algorithms and propose the most suitable algorithm for developing a construction progress analysis system. Through an analysis of existing research, the SOR, ROR, and COR algorithms were selected, and their performance was compared based on data accuracy, data processing time, and data preservation rate.

For the accuracy, the outlier removal rate was analyzed based on the number of outliers identified by each algorithm. For the data processing time analysis, the time taken by each preprocessing algorithm to process the raw data was measured. For the data preservation rate analysis, the number of point cloud coordinates before and after the application of the algorithms was compared to analyze the preservation rate. During this process, it was found that outliers included along the boundary of the wall compromised the reliability of the preservation rate. To improve the reliability of the data preservation rate comparison, a feature extraction algorithm was employed to reanalyze the preservation rate.

The study concluded that the SOR algorithm, which showed excellent results across all comparison criteria, is the most suitable for construction progress analysis. However, this study has the limitation that the analysis was restricted to a single exterior wall on the 4th and 5th floors of the building. As the scope and scale of data analysis increase, the results may vary. Indeed, the comparison of data preservation rates showed that the SOR, ROR, and COR algorithms all recorded high preservation rates of over 98%, which is likely due to the relatively small size of the dataset used in this study. This issue could be addressed by increasing the size of the raw point cloud data and adjusting the algorithm's parameter values. By doing so, potential problems associated with increasing data size could be mitigated, leading to more accurate results.

In the development of a progress management system using point clouds, applying the SOR algorithm and addressing its limitations is expected to enhance accuracy. By adjusting the algorithm's parameters to better suit construction progress analysis, the utility and efficiency of construction progress analysis could be further improved.

7. ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT). (No. RS-2024-00338908)

8. REFERENCES

1. Yu JH, Kim OK. A case study on the prevention of construction delays using the delay management index in program level construction projects. J Korea Inst Build Constr. 2021;21(4):347-359. URL: https://www.riss.kr/link?id=A107845531

2. Young-Hoon Kim, Jaehwan Lee. Technology and Service Trends for Ensuring Safety in Smart Manufacturing . The Journal of Innovation Industry Technology 2023;1(3):123-128. DOI: https://doi.org/10.60032?JIIT.2023.1.3.123

3. Park S-M, Kim Y-G, Ham N-H, Kim J-J. A Case Study on the Application of Digital-Twin to Small-size Building Remodeling Processes. Journal of KIBIM [Internet]. 2023 Mar 31;13(1):1–12. Available from: https://doi.org/10.13161/KIBIM.2023.13.1.001

4. Yu D, Lee C, Sim S-H. Point Cloud-based Automated Building Tilt Measurement. Journal of the Korea institute for structural maintenance and inspection [Internet]. 2023 Oct 31;27(5):84–8. Available from: https://doi.org/10.11112/JKSMI.2023.27.5.84

5. Kim T-H, Gu H-M, Hong S-M, Choo S-Y. Development of Deep Learning-based Automatic Classification of Architectural Objects in Point Clouds for BIM Application in Renovating Aging Buildings. Journal of KIBIM [Internet]. 2023 Dec 31;13(4):96–105. Available from: https://doi.org/10.13161/KIBIM.2023.13.4.096

6. Yoo J-S, Yoo T-K, Seo S-H. AI-based 3D Spatial Structure Estimation Using Indoor Floor Point Cloud. Journal of Digital Contents Society. 2022;23(6): 1013-1020. DOI: https://doi.org/10.9728/dcs.2022.23.6.1013

7. Wang, P., Gu, T., Sun, B., Huang, D., & Sun, K. (2022). Research on 3D Point Cloud Data Preprocessing and Clustering Algorithm of Obstacles for Intelligent Vehicle. World Electric Vehicle Journal, 13(7), 130. DOI: https://doi.org/10.3390/wevj13070130

8. Lee, M. Y., Lee, S. H., Jung, K. D., Lee, S. H., & Kwon, S. C. (2021). A Novel Preprocessing Method for Dynamic Point-Cloud Compression. Applied Sciences, 11(13), 5941. DOI: https://doi.org/10.3390/app11135941

9. Kim H, Hwang Y, Rhee S. Automatic Generation of Clustered Solid Building Models Based on Point Cloud. Korean Journal of Remote Sensing [Internet]. 2020 Dec 31;36(6_1):1349–65. Available from: https://doi.org/10.7780/KJRS.2020.36.6.1.6

10. Kang T-W, Kim J-E. Large Point Cloud-based Pipe Shape Reverse Engineering Automation Method [Internet]. Vol. 17, Journal of the Korea Academia-Industrial cooperation Society. The Korea Academia-Industrial Cooperation Society; 2016. p. 692–8. Available from: http://dx.doi.org/10.5762/kais.2016.17.3.692

11. Zhao, L., Hu, Y., Yang, X., Dou, Z., & Kang, L. (2024). Robust multi-task learning network for complex LiDAR point cloud data preprocessing. Expert Systems with Applications, 237(B), 121552. DOI: https://doi.org/10.1016/j.eswa.2023.121552

12. Li, F., Shi, W., Tu, Y., & Zhang, H. (2023). Automated methods for indoor point cloud preprocessing: Coordinate frame reorientation and building exterior removal. Journal of Building Engineering, 76, 107270. DOI: https://doi.org/10.1016/j.jobe.2023.107270

13. Kurup, A., & Bos, J. Dsor: A scalable statistical filter for removing falling snow from lidar point clouds in severe winter weather. arXiv 2021. arXiv preprint arXiv:2109.07078. DOI: https://doi.org/10.48550/arXiv.2109.07078

14. Ning X, Li F, Tian G, Wang Y (2018) An efficient outlier removal method for scattered point cloud data. PLoS ONE 13(8): e0201280. DOI: https://doi.org/10.1371/journal.pone.0201280

15. Balta, H., Velagic, J., Bosschaerts, W., De Cubber, G., & Siciliano, B. (2018). Fast statistical outlier removal based method for large 3D point clouds of outdoor environments. IFAC-PapersOnLine, 51(22), 348-353. DOI: https://doi.org/10.1016/j.ifacol.2018.11.566

16. Kim H, Hwang Y, Rhee S. Automatic Building Modeling Method Using Planar Analysis of Point Clouds from Unmanned Aerial Vehicles. Korean Journal of Remote Sensing [Internet]. 2019 Dec 31;35(6_1):973–85. Available from: https://doi.org/10.7780/KJRS.2019.35.6.1.8