Surface Reconstruction from Unstructured Point Cloud Data for Building Digital Twin

F.A. Ismail[1], S.A. Abdul Shukor[2], N.A. Rahim[3], R. Wong[4]
Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis, Arau, 02600, Perlis, Malaysia[1, 2, 3]
Centre of Excellence for Intelligent Robotics & Autonomous System,
Universiti Malaysia Perlis, Arau, 02600, Perlis, Malaysia[1, 2, 3]
Geodelta Systems Sdn. Bhd., Damansara Utama, 47400 Petaling Jaya, Selangor, Malaysia[4]

***Abstract*—This study highlights on the methods used for surface reconstruction from unstructured point cloud data, characterized by simplicity, robustness and broad applicability from 3D point cloud data. The input data consists of unstructured 3D point cloud data representing a building. The reconstruction methods tested here are Poisson Reconstruction Algorithm, Ball Pivoting Algorithm, Alpha Shape Algorithm and 3D surface refinement, employing mesh refinement through Laplacian smoothing and Simple Smoothing techniques. Analysis on the algorithm parameters and their influence on reconstruction quality, as well as their impact on computational time are discussed. The findings offer valuable insights into parameter behavior and its effects on computational efficiency and level of detail in the reconstruction process, contributing to enhanced 3D modeling and digital twin for buildings.**

Keywords—Surface reconstruction; point cloud; building reconstruction; 3D mesh

I. Introduction

Buildings play a pivotal role in our everyday lives, and consequently, considerable endeavors have been directed towards enhancing them. One approach to achieve this enhancement involves integrating digital technologies throughout the entire life cycle of the building, encompassing various stages such as planning, construction, operation, renovation and demolition [1]. Notably, significant attention has been dedicated to integrating digital advancements into the construction life cycle in the past decade [2]. Throughout this life cycle, three-dimensional (3D) models have demonstrated their utility in facilitating decision-making, scenario modeling, and analysis of 3D data. In recent years, there has been a growing demand for effective and efficient monitoring of changes in buildings and construction installations within urban areas. This demand is particularly evident in the domains of architecture, engineering, construction/facility management (AEC/FM), urban planning, surveying and mapping. Various applications, such as progress tracking, profitability enhancement, quality control, security assurance and incident investigation, underscore the necessity for advanced methodologies that employ automated measurements, including 2D imaging, photogrammetry and 3D laser scanning, instead of relying solely on visual inspection and manual data collection.

3D building models are important in representing the urban environment and have numerous applications, including 3D Geographic Information Systems (GIS), urban planning, environmental simulation, energy consumption assessment, tourism, mobile navigation, heritage preservation and change detection [3]–[5]. Many studies have been conducted throughout the years on the reconstruction of building information models (BIM). BIM, as a valuable methodology, serves as a pivotal tool in facilitating the planning and construction processes of architectural and infrastructural projects. BIM models intricately replicate physical structures, offering a comprehensive foundation for undertaking assessments of their status, condition, and strategizing maintenance activities. Furthermore, BIM presents a unified and cohesive platform for the seamless integration of data collected from the construction site.

Recently, the concept of BIM has expanded to include the concept of a digital twin (DT). DT is a virtual representation of a physical entity that used to simulate and evaluate the performance of a building throughout its lifespan. The digital twin technology can be utilized for various purposes such as visualization, modelling, simulation, analysis and future planning [6], [7]. These virtual models can help optimize the design and building process, anticipate potential issues and enhance the building's performance and features over time [8]. The idea of a digital twin was initially introduced by M. Grieves during a Product Life-Cycle Management Symposium at the University of Michigan Lurie Engineering Center in 2002 [9]. The proposed model of a digital twin comprises three primary components: the physical product, the virtual product and the connection between the physical and virtual entities. In a subsequent publication [10], Grieves further defined digital twinning as the integration of three essential elements: a virtual twin, a physical counterpart (such as a product, system, model, or entity like a robot, car, power turbine, human, hospital, etc.), and a data flow cycle that facilitates the exchange of information between the physical and virtual twins. The virtual twin employs simulation algorithms to replicate (either fully or partially) the performance of its physical counterpart, generating equivalent outputs based on input values. This technology is commonly used in the context of smart manufacturing but is applicable to various domains, including construction, education, transportation, human and healthcare, also industrial production [7]. The primary advantages of digital twin models are their ability to access and query structured data and their visual representation of information. Digital twins undergo periodic updates to maintain alignment with their physical counterparts. The frequency of these updates varies contingently upon factors including the inherent characteristics of the product, its dynamic attributes and the specific objectives underlying the model's use. For instance, in the case of a jet engine, updates may occur at minute intervals, whereas for maintenance management of a building, annual updates may suffice. Notwithstanding, a notable complexity emerges due to the historical context of many extant buildings, often constructed decades ago, thereby necessitating the development of digital twin models that accurately represent these pre-existing assets.

To create a digital twin of a building, one of the essential steps is to capture its geometry and appearance using point cloud data. A point cloud is a collection of points in 3D space that represent the surface of an object. These data points typically comprise X, Y, and Z coordinates and are primarily utilized to depict the outer surfaces of an entity [11]. Point clouds can be obtained from various sources, such as laser scanners or cameras, by capturing the geometry of an existing facility. These techniques produce point cloud data as output. Compared to visual inspection, point cloud data offers shorter processing times and higher measurement accuracy. With the introduction of very precise data collection techniques involving terrestrial laser scanning, aerial oblique photography and satellite imagery, 3D point cloud has established itself as the principal data sources for large-scale building reconstruction. Point clouds can be processed and reconstructed to create 3D models of building interiors and exteriors in vector format. Furthermore, existing research projects have emerged proposing approaches for generating accurate building footprints and models by combining point clouds and imaging [12]–[14]. This interdisciplinary field, which includes photogrammetry, computer vision and modelling, has seen significant research efforts over the last two decades, delivering important and significant results.

While 3D building models and digital twins offer immense potential in urban development and architectural landscapes, there exists a notable gap in systematically and critically analyzing the various surface reconstruction techniques that form the foundation of these models especially on its simplicity, robustness, and broad applicability in surface reconstruction from point cloud data. This paper aims to bridge this research gap by addressing the following pivotal questions: (a) How does different surface reconstruction technique perform when applied to large, unstructured point cloud data like buildings, both in terms of quality and speed of calculation? and (b) In what contexts do these techniques exhibit optimal efficiency and precision? This paper central contribution is an in-depth comparative assessment of specific surface reconstruction techniques, bringing clarity to the challenges, advantages, and nuances of each. In doing so, professionals and researchers in the related field could equip with a clearer understanding and guide for their practical and academic endeavors.

In existing research, a rich set of methods has been proposed for surface reconstruction from point clouds, and some reviews and benchmarks of these methods have also been provided [15]. However, they still face challenges in terms of robustness, generalization and efficiency, especially for unstructured, complex and large-scale surfaces such as buildings. Furthermore, the scalability and parallelizability of these methods become critical when dealing with big data, as they must handle enormous point cloud collections while maintaining computational efficiency. Point cloud-based 3D reconstruction in buildings has many applications for the construction industry, such as automatic creation of as-built BIMs, damage detection and assessment, cultural heritage, and facility management [16]–[18]. Therefore, there is a need for developing an effective method for surface reconstruction from point clouds that can handle the specific characteristics and requirements of building surfaces.

This paper focuses on a few types of surface reconstruction techniques use for building data. Thus, a comparison between these techniques, based on the quality of the surfaces and the speed of calculation, will be made. This work is organized as follows: Section II presents a summary of reconstruction methods from a set of discrete data points or sample. Section III shows the visualization and discussion of the surface reconstruction results for more understanding. Lastly, Section IV explains the conclusion for this paper.

II. Methodology

The overview of the sample data used for reconstructing 3D surface of a building is as shown in Fig. 1. The point cloud data represents one of the buildings in Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis, Malaysia. The initial step involves converting the binary data into a more interpretable format compatible with standard 3D libraries and applications. Subsequently, downsampling the data is performed to make it efficiently better processed with shorter computational time. Next, data cleaning is implemented to mitigate imperfections in the real data, enhancing the efficiency of the surface reconstruction method for mesh generation. Finally, five different surface reconstruction techniques are applied and assessed based on the resulting processed point cloud data to generate a 3D triangulated mesh.

A. The Sample Data

The point cloud data of 3D coordinates is stored in six different files and saved in .ptx file format and consists of approximately 226,471,204 points. The data is collected by Geodelta Systems Sdn. Bhd. using 3D terrestrial laser scanner model Leica RTC360 where it has a scanning speed up to 2 million pts/sec and advanced HDR imaging.

To extract the 3D file format from the .ptx file and transform it into a commonly used 3D point cloud data format, such as the .pcd file format, a comprehensive understanding of the binary file data structure assumes paramount significance. Each .ptx file encompasses vital data elements, encompassing color information, the minimum depth value (zmin), the subsampled number of rows (nrows), the subsampled number of columns (ncols), the image file name, and an Nx5 matrix signifying 3D and 2D normalized coordinates falling within the [0,1] range [19]. Herein, N denotes the product of nrows and ncols, where values equivalent to zmin denote the background. The extraction process selectively focuses on the isolation and preservation of solely the 3D coordinates from the .ptx file, ultimately saving them in the .pcd file format, facilitating subsequent analytical endeavors. Various software tools, such as CloudCompare, MeshLab, Blender and Python 3D libraries, can read the .pcd file format, facilitating data visualization and manipulation. This enables easier understanding and modification of the data as needed. An example of an extracted point cloud dataset is illustrated in Fig. 1, where the image at the below shows a zoomed-in view of the original 3D point cloud shown on the top. From the figure, it can be inferred that each point exhibits ambiguous relationships with neighbouring points.

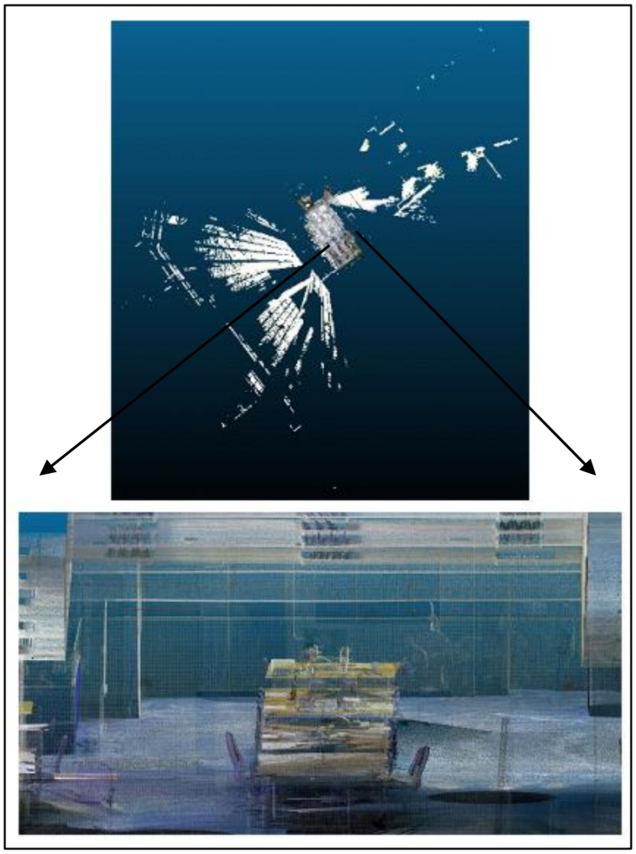


Fig. 1. The point cloud data used in this study.

B. *Data preprocessing*

In order to improve the data quality and computational efficiency, a preprocessing stage is employed to eliminate unnecessary data of point clouds. One of the often used as a pre-processing step for many point cloud data processing tasks is voxel downsampling. Downsampling is a technique to reduce the number of points in a point cloud as the original dataset is too large to handle. It can improve efficiency and accuracy by minimizing storage requirements, processing time and memory usage. One of the common methods for downsampling is voxel downsampling. It uses a regular voxel grid to create a uniformly downsampled point cloud from an input point cloud. The algorithm follows a two-step process: initially, data points are grouped into voxel containers and subsequently, each occupied voxel yields a single representative point through the computation of the average of all points contained within it. So, all of the point clouds are being downsampled to 0.01 resulted in easier to be processed due to lesser number of points as shown in Fig. 2. The downsampled point cloud data reduced to 12,704,776 points, from its original which consists of 226,471,204 points.

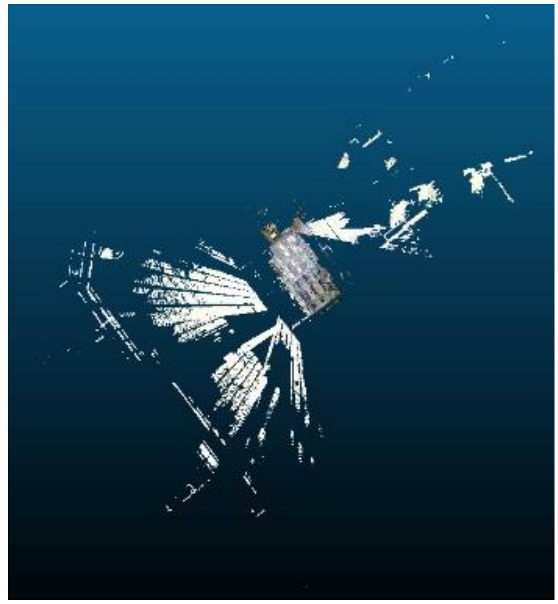


Fig. 2. Downsampled data of original point cloud.

Next, the preprocessing step involves removing statistical outliers. It removes points that are further away from their neighboring points. The mean inter-point separation is computed via the application of the k-nearest neighbors algorithm. Should the computed average distance between a query point and its neighboring points surpass a threshold established by the standard deviation, it is categorized as an outlier and subsequently excluded from the dataset. This initial preprocessing stage employs various standard deviation ratios, notably a factor of 0.75, to accentuate differentiation. The parameter kNN, signifying the number of nearest neighbors considered, is set to a value of 50. Fig. 3 shows the visualization of the point cloud data with red and grey colour that indicates the outlier and inlier. The red colour is the outliers that being filtered out. Fig. 4 shows the remaining point cloud in original colour of 12,225,180 points.

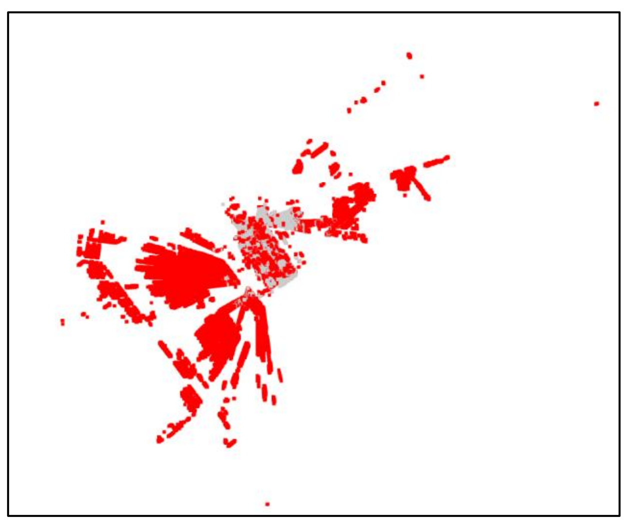


Fig. 3. Inlier (in grey) and outlier (in red) of the point cloud data.

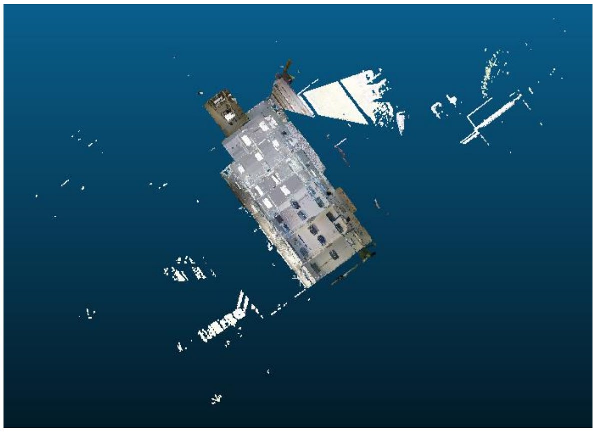


Fig. 4. Remaining point cloud after outlier removal.

However, the remaining point clouds are not accurate as the surrounding of the buildings still contains points that are considered as noise. So, a process of filtering takes over using CloudCompare by manually remove the excessive point cloud surrounding the building. Fig. 5 below shows the remaining point cloud data that will be processed for surfacce reconstruction which only consist of 7,258,055 points.

Fig. 5. Filtered point cloud.

C. Surface reconstruction

The field of geometry processing has significant emphasis on the foundational challenge of surface reconstruction from point clouds. This endeavor is rooted in the objective of generating a continuous 2D manifold surface from the inherent sparsity of raw, discrete point cloud data. It is worth noting that this problem is inherently ill-posed, and its complexity escalates when considering the various sensor-related imperfections that manifest within point clouds acquired through real-world depth scanning techniques.

Hence, this paper will focus exclusively on five distinct surface reconstruction approaches, aimed at visualizing a range of methodologies advanced by prior researchers addressing this issue. These methodologies can be broadly categorized into three primary groups, namely Alpha Shape, Ball Pivoting and Poisson reconstruction, as well as combinations of Alpha Shape with Ball Pivoting and the amalgamation of Alpha Shape, Ball Pivoting and mesh refinement.

The Alpha Shape technique serves as an extension of the convex hull method, allowing for the representation of point clouds with concavities and holes. However, choosing an appropriate value for "α" requires a balance. If the value is too small, the resulting mesh may include more noise or artifacts, and it may overfit to local irregularities in the point cloud. On the other hand, if the value is too large, the resulting mesh may oversimplify the shape and fail to capture fine details or concavities. By adjusting the parameter alpha or "α", the level of detail in the reconstructed surface can be controlled. A smaller "α" value results in a more intricate surface, while a larger "α" value produces a smoother and simpler surface. The reconstruction process involves computing the Delaunay tetrahedralization of the point cloud and extracting the faces belonging to the "α" complex as it contains points, edges, triangles and tetrahedrons.

Ball Pivoting Algorithm (BPA), another surface reconstruction method, is closely related to the Alpha Shape technique. It is an efficient surface reconstruction approach which operates by simulating the rolling of a ball with a predetermined radius across the point cloud. Triangles are created whenever the ball encounters three points without penetrating them. As the ball moving through the point cloud, a triangular interconnected 3D mesh is formed, linking the 3D points. The method is repeated until all the points form a triangle. Starting with a seed triangle, the algorithm pivots the ball around the edges of existing triangles until all feasible triangles are produced. The BPA exhibits sensitivity to variations in the ball radius, which plays a crucial role in determining the quantity of reconstructed faces. A smaller radius renders the model susceptible to noise in the input data, while a larger radius may lead to the missing of intricate details, resulting in the formation of holes within the generated surface. BPA is suitable for handling point clouds with non-uniform densities and noise; however, careful selection of the ball radius is crucial to prevent gaps or overlaps in the reconstructed surface.

Poisson reconstruction is a widely recognized approach for generating smooth surfaces from point clouds using a volumetric strategy. It leverages oriented point samples obtained from 3D range scanners to create watertight surfaces. The method assumes that the point cloud serves as a sample of an underlying surface's indicator function and solves for an implicit function whose gradient best aligns with the estimated normals of the point cloud. The surface is subsequently extracted as an iso-surface of the implicit function utilizing marching cubes. Poisson reconstruction excels in producing high-quality surfaces with well-defined features, but it requires oriented normals as input and may introduce undesired intricacies in flat regions. Nevertheless, this technique exhibits resilience to noisy data and artifacts arising from misregistration.

III. Results and Discussion

All surface reconstruction techniques were evaluated and tested on a computer system comprising an Intel i7 12700h processor with 32GB of RAM and a GTX 3050ti graphics card. All methods were implemented using the Open3D library in python. Fig. 6 shows the clean point cloud data of the chosen building that will be used for the surface reconstruction algorithms upon preprocessing and Fig. 7 shows the close-up view for the point cloud for better understanding to show its complexity and unstructured of the data. Table I shows the comparison of all algorithms with the parameters used, time taken for each algorithm needs to be completed, together with their resulting images from two views: the whole building and close-up view from the top corner.

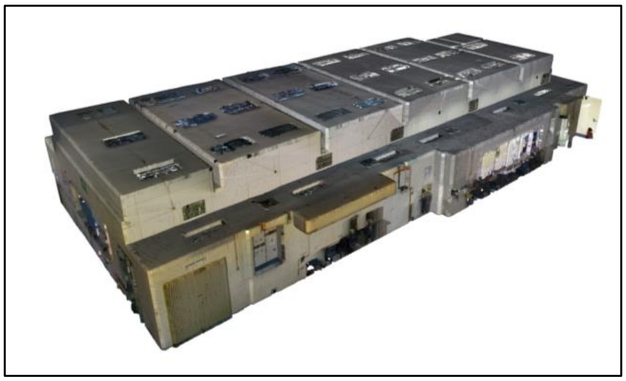
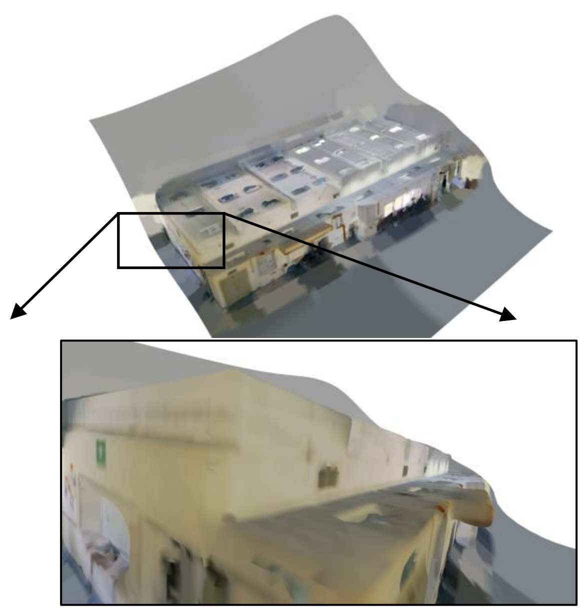


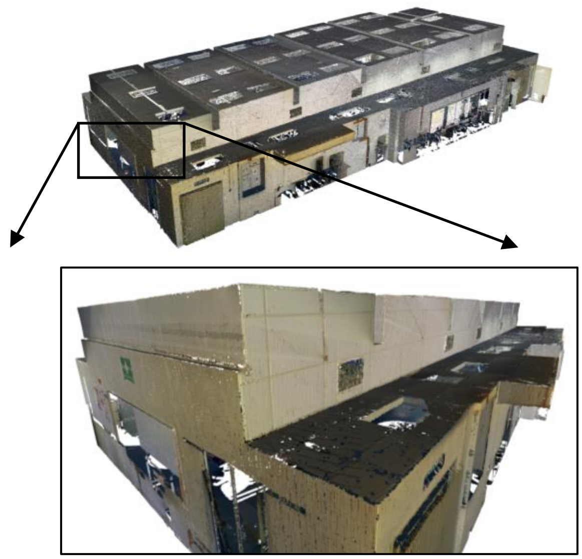
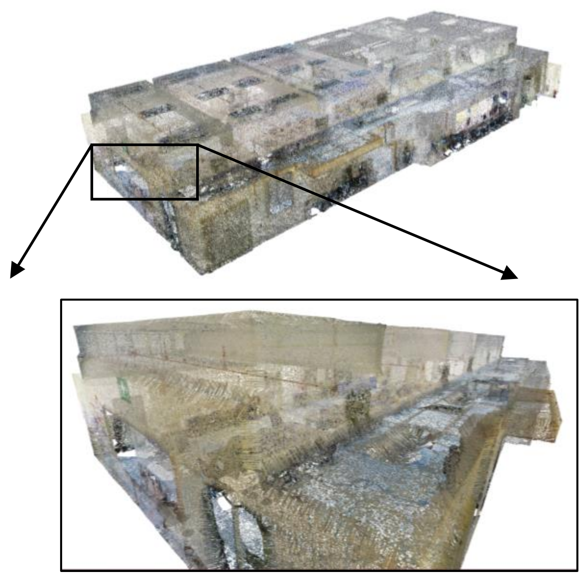
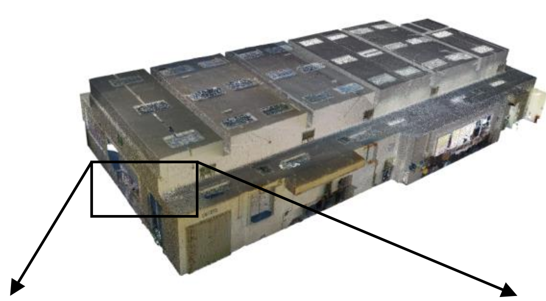
Fig. 6. Point cloud data of the building.

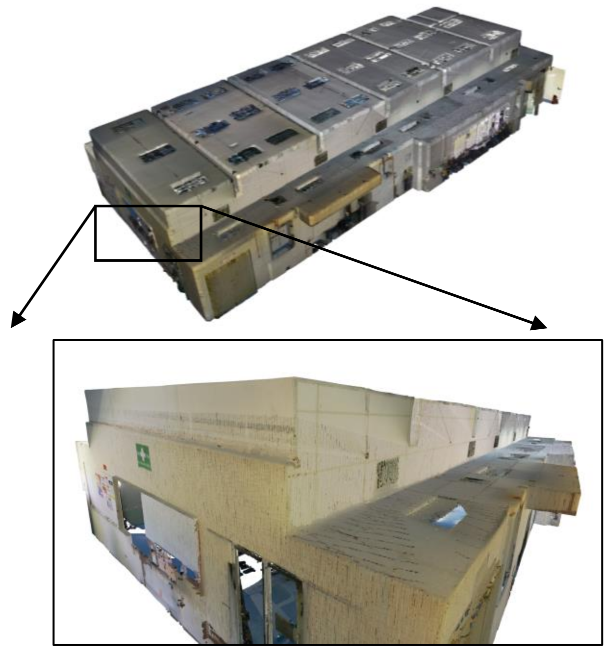
Fig. 7. Close-up view of the point cloud.

Poisson surface reconstruction necessitated the prior prediction of normals for surface reconstruction, employing it automatically by the maximum nearest neighbour search. The depth of reconstruction indicated the resolution of the resulting triangle mesh, determined by the octree's depth which being set to 8. The minimum number of sample points within an octree node was established to adapt the octree construction to the sampling process, with lower numbers chosen to mitigate noise interference. The scale factor denoted the ratio between the reconstruction cube's diameter and the sample bounding box's diameter. Nevertheless, Poisson surface reconstruction yielded outcomes without any mesh holes but it leaves unnecessary mesh triangulations on planar point as shown in Table I.

TABLE I. Summary of Result Comparison for the Selected Algorithms

Algorithms	Parameters	Processing Time (s)	Results
Poisson Reconstruction	**Octree depth=8 Scale factor=1**	**163.92**	

Ball Pivoting	**Ball radius=0.02**	**288.34**	
Alpha Shape	**α = 0.13**	**303.19**	
Alpha Shape + Ball Pivoting	**α = 0.13 Ball radius=0.02**	**566.38**	

			
Alpha Shape + Ball Pivoting + Mesh Filtering	**α = 0.13** **Ball radius=0.02** **Number of iterations (Simple Smoothing)=30** **Number of iterations (Laplacian)=30** **Lambda filter=0.5**	**1092.07**	

On the other hand, Ball Pivoting surface reconstruction required prior normal estimation before commencing surface reconstruction. Vertices falling within a given percentage of the clustering radius were merged to prevent excessive small triangle creation. The ball rolling process would cease if it encountered a significant crease angle surpassing the threshold angle. The ball's rolling radius over the point cloud was set to 0.02. Table I exhibited also the outcomes of ball pivoting surface reconstruction, where holes were observed in regions with lower point cloud density. However, unlike Poisson surface reconstruction, this method avoided unnecessary mesh creation that required subsequent cropping since it failed to represent facial depth accurately.

Meanwhile, the result of meshing using the Alpha Shape method with a parameter value of 0.13 involves generating a surface mesh that captures the shape of a point cloud with holes and concavities. The Alpha Shape method is a generalization of the convex hull and is particularly useful when dealing with point clouds that have irregular or non-convex shapes. Unlike Poisson and Ball Pivoting, Alpha Shape algorithm does not require normal estimation. The parameter "α" in the Alpha Shape method determines the level of detail in the reconstructed surface. A smaller value of "α" (0.13) produces a more detailed mesh with a higher resolution where smaller features and intricate details present in the point cloud can be captured in the resulting mesh as shown in Table I. The mesh will closely follow the shape of the input point cloud and adapt to its local density and curvature. Therefore, the choice of the "α" parameter is decided after running several tests with different values. This gives the best result as it still can capturing details and maintaining the overall shape fidelity in the resulting mesh.

Due to the limitations of each individual algorithm, the combination of Ball Pivoting and Alpha Shape methods in meshing involves leveraging the strengths of both approaches to achieve a more robust and accurate representation of the underlying point cloud. By combining Ball Pivoting and Alpha Shape methods, the strengths of both techniques can be fully utilized. Ball Pivoting handle the local details and irregularities of the point cloud, while Alpha Shape capture the overall shape and handle concavities. This combined approach produced a mesh that preserves the important features and characteristics of the original point cloud, while also providing a more accurate and visually appealing representation as shown. However, the Alpha Shape results in not so precise reconstruction when it comes to sharpness of angles of the building as shown in Table I.

The integration of Ball Pivoting, Alpha Shape and mesh refinement through Laplacian smoothing and Simple Smoothing however constitutes a comprehensive strategy for the task of surface reconstruction and subsequent mesh enhancement. In terms of mesh refinement, two distinct smoothing techniques are employed. Laplacian smoothing involves iterative adjustments of vertex positions, effectively attenuating high-frequency noise and enhancing mesh smoothness while preserving sharp geometric features. Conversely, Simple Smoothing employs a rudimentary moving average filter to vertices, delivering a general refinement to the mesh's geometric properties. This two-fold smoothing methodology serves the purpose of mitigating undesired irregularities and augmenting the overall visual fidelity of the mesh representation. Through the combination of these methodologies, the resultant mesh stands to gain advantages in terms of heightened accuracy, diminished noise, and improved surface regularity. The amalgamation of Ball Pivoting, Alpha Shape, and the aforementioned dual smoothing techniques yields a potent solution applicable across diverse domains, encompassing computer graphics, 3D modeling, virtual reality, and computational geometry. Furthermore, the adaptability inherent in this combined approach empowers users to fine-tune parameters and iteration counts, thereby achieving optimal outcomes tailored to the distinctive attributes of their datasets and the desired level of mesh refinement. Consequently, this combination of methodologies emerges as an efficacious and versatile technique for addressing surface reconstruction and mesh refinement tasks.

In summary, each of the methodologies introduced for surface reconstruction has its respective advantages and limitations. The Poisson surface reconstruction method consistently produces hole-free results but can also introduce unnecessary triangulations. On the other hand, the Ball Pivoting method excels in avoiding unnecessary mesh production yet lacks in representing facial depth precision. The Alpha Shape method, adaptable to non-convex and irregular shapes, offers a versatile solution to meshing, though it may occasionally compromise the precision of angles. However, it becomes evident that the approach emerges from the integration of Ball Pivoting, Alpha Shape, and mesh refinement techniques, specifically through the incorporation of Laplacian and Simple Smoothing. This combination technique harnesses the unique strengths of each individual method, producing an optimal solution that balances accuracy, noise mitigation, and enhanced surface consistency. As a result, this integrated methodology stands out as the most recommended, providing unparalleled robustness and adaptability suitable for a wide range of surface reconstruction and mesh refinement applications.

Upon analyzing the results obtained, it becomes imperative to consider the scalability of the proposed methodologies. While the techniques have shown promising outcomes on the datasets in this study, there is an important area of evaluation to consider. Their performance across datasets that represent distinct architectural styles, varying scales, and diverse complexities needs to be examined. Future investigations should focus on evaluating these methods across a wider range of datasets. This approach will help validate the adaptability and universality of the proposed techniques. In scenarios where the datasets are vast and depict intricate urban landscapes, it is vital to assess metrics such as computational efficiency, memory requirements, and the fidelity of the resultant mesh.

Additionally, despite the promising outcomes, there exist potential limitations and areas for further exploration. One foreseeable challenge pertains to highly intricate and ornate architectural designs, where capturing every minute detail could strain the computational resources and necessitate further algorithmic optimizations. Moreover, while the combination of Ball Pivoting and Alpha Shape methods harnesses their collective strengths, there remains room for improvement in handling sharp angles, as evidenced in Table I. Future research could just focus on refining these combinations, possibly integrating other meshing algorithms or advanced smoothing techniques to better address such challenges. Furthermore, with the rapid evolution of hardware and software capabilities, exploring the integration of machine learning or AI-driven approaches in the surface reconstruction pipeline could offer innovative solutions, enhancing the accuracy and efficiency of the process.

IV. Conclusion

In conclusion, the successful implementation of specific surface reconstruction methods is contingent upon satisfying the input requirements, such as having normal orientation information available for the point cloud. In cases where the input lacks normal orientation, an algorithm must be employed to estimate the orientation values for the point cloud. For instance, Poisson surface reconstruction necessitates the estimation of normal orientations prior to the reconstruction process. Certain methods exhibit robustness towards specific types of point cloud artifacts, including noisy data, nonuniform sampling, outliers, misaligned scans and missing or incomplete data. However, certain methods may be limited in their applicability to different shape classes. For instance, surface reconstruction cannot handle shape classes that do not result in watertight meshes. The output of the reconstruction process can manifest in various forms, such as producing watertight meshes, mesh triangulation on planar surfaces, generating cloth-like meshes that envelop the initial point cloud or voxelization. The result of Poisson surface reconstruction often presents unnecessary mesh triangulations on planar point clouds, resembling cloth-like structures, while effectively closing holes in sparsely sampled regions. On the other hand, Ball Pivoting surface reconstruction tends to yield accurate mesh edges but can leave numerous holes due to nonuniform sampling and missing data, particularly in areas like the angle’s region with less point cloud. By combining different methods, it can help to achieve smoother with higher accuracy of the surface of buildings.

Acknowledgment

The authors would like to acknowledge the support from the Fundamental Research Grant Scheme (FRGS) under the grant number of FRGS/1/2022/TK07/UNIMAP/02/23 from the Ministry of Higher Education Malaysia.

REFERENCES

[1] R. G. Kippers, M. Koeva, M. Van Keulen, and S. J. Oude Elberink, “Automatic 3D building model generation using deep learning methods based on CityJSON and 2D floor plans,” in International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, International Society for Photogrammetry and Remote Sensing, Oct. 2021, pp. 49–54. doi: 10.5194/isprs-archives-XLVI-4-W4-2021-49-2021.

[2] Q. Meng et al., “A review of integrated applications of BIM and related technologies in whole building life cycle,” Engineering, Construction and Architectural Management, vol. 27, no. 8. Emerald Group Holdings Ltd., pp. 1647–1677, Oct. 01, 2020. doi: 10.1108/ECAM-09-2019-0511.

[3] Y. P. Ma, “Extending 3D-GIS district models and bim-based building models into computer gaming environment for better workflow of cultural heritage conservation,” Applied Sciences (Switzerland), vol. 11, no. 5, pp. 1–23, Mar. 2021, doi: 10.3390/app11052101.

[4] B. Yang, “Developing a mobile mapping system for 3D-GIS and smart city planning,” Sustainability (Switzerland), vol. 11, no. 13, Jul. 2019, doi: 10.3390/su11133713.

[5] M. Awrangjeb, S. A. N. Gilani, and F. U. Siddiqui, “An effective data-driven method for 3D building roof reconstruction and robust change detection,” Remote Sens (Basel), vol. 10, no. 10, Oct. 2018, doi: 10.3390/rs10101512.

[6] M. Enders, M. R. Enders, and N. Hoßbach, “Dimensions of digital twin applications-A literature review,” 2019. [Online]. Available: https://www.researchgate.net/publication/359715537

[7] M. Liu, S. Fang, H. Dong, and C. Xu, “Review of digital twin about concepts, technologies, and industrial applications,” J Manuf Syst, vol. 58, pp. 346–361, Jan. 2021, doi: 10.1016/j.jmsy.2020.06.017.

[8] H. H. Hosamo, H. K. Nielsen, A. N. Alnmr, P. R. Svennevig, and K. Svidt, “A review of the digital twin technology for fault detection in buildings,” Frontiers in Built Environment, vol. 8. Frontiers Media S.A., Nov. 09, 2022. doi: 10.3389/fbuil.2022.1013196.

[9] Michael Grieves, Complex Systems Engineering: Theory and Practice. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2019. doi: 10.2514/4.105654.

[10] Michael Grieves, “Digital Twin: manufacturing excellence through virtual factory replication,” White paper, vol. 1, pp. 1–7, 2014.

[11] M. Rashidi, M. Mohammadi, S. S. Kivi, M. M. Abdolvand, L. Truong-Hong, and B. Samali, “A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions,” Remote Sensing, vol. 12, no. 22. MDPI AG, pp. 1–34, Nov. 02, 2020. doi: 10.3390/rs12223796.

[12] F. Wang et al., “Reconstruction of LoD-2 building models guided by Façade Structures from Oblique Photogrammetric Point Cloud,” Remote Sens (Basel), vol. 15, no. 2, Jan. 2023, doi: 10.3390/rs15020400.

[13] A. Jamali, P. Kumar, and A. Abdul Rahman, “Automated extraction of buildings from aerial lidar point clouds and digital imaging datasets,” in International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, International Society for Photogrammetry and Remote Sensing, Oct. 2019, pp. 303–308. doi: 10.5194/isprs-archives-XLII-4-W16-303-2019.

[14] S. Cao, D. Hu, W. Zhao, M. Du, Y. Mo, and S. Chen, “Integrating multiview optical point clouds and multispectral images from ZiYuan-3 satellite remote sensing data to generate an urban digital surface model,” J Appl Remote Sens, vol. 14, no. 01, p. 1, Jan. 2020, doi: 10.1117/1.jrs.14.014505.

[15] R. Sharma and P. Abrol, “Parameter extraction and performance analysis of 3D surface reconstruction techniques,” International Journal of Advanced Computer Science and Applications, vol. 14, no. 1, pp. 331–336, 2023, doi: 10.14569/IJACSA.2023.0140135.

[16] V. Stojanovic, M. Trapp, R. Richter, B. Hagedorn, and J. Döllner, “Towards the generation of digital twins for facility management based on 3D point clouds,” 2018.

[17] Y. Zhou, S. Shen, and Z. Hu, “Detail preserved surface reconstruction from point cloud,” Sensors (Switzerland), vol. 19, no. 6, Mar. 2019, doi: 10.3390/s19061278.

[18] Q. Wang and M. K. Kim, “Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018,” Advanced Engineering Informatics, vol. 39. Elsevier Ltd, pp. 306–319, Jan. 01, 2019. doi: 10.1016/j.aei.2019.02.007.

[19] M. A. Zuraimi, F. H. K. Zaman, and L. Mazalan, “3D human face surface reconstruction from unstructured point cloud face images,” in 2022 IEEE 13th Control and System Graduate Research Colloquium, ICSGRC 2022 - Conference Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 165–170. doi: 10.1109/ICSGRC55096.2022.9845175.