

Development of Cloud Point Data Processing Program for 3D BIM and 2D Cross Section Needs

Muhammad Farhan Mufid Kusuma^a, Akhmad Aminullah^{b*}, dan Djoko Sulistyob^b

^a Postgraduate Program, Department of Civil and Environmental Engineering, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia

^b Department of Civil and Environmental Engineering, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia

Keywords:
LiDAR Scanner
Point Cloud
Surface 3D
Library Python
Script Code

ABSTRACT

The need for technological developments is needed to facilitate performance, accuracy, and effectiveness of work, especially in the field of civil engineering, is needed. With the emergence of innovative LiDAR (Light Detection and Ranging) technology scanners that are popularly used for 3D printing, developed into LiDAR Scanners for real field scanning. The result of using a LiDAR Scanner is in the form of point cloud data in a certain format, with a large enough memory. The purpose of this research is to use field point cloud data as 3D BIM data and then form a cross-section of the object. For this purpose, a special program is needed that functions to process cloud point data complexly and is easy to use to change the shape of cloud point data to 3D data surface and 2D cross sections. The method used in this study is by creating a special program to process data point clouds using script code with the python language and several data point cloud processing libraries. In the program, 2 sub-menus will be created with certain functions: 1) Point Cloud (voxel downsampling, outlier reduction, normalize); 2) 3D model (ball pivoting/poisson surface, reduce vertex, slice mesh, transform mesh). In each data processing, the created program can only process on a specific file format; for point cloud processing in .xyz, .xyzn, .xyzrgb, .pts, .ply, .pcd formats; while for 3D data processing models are in .ply, .stl, .obj, .off, .gltf/glb format. The result of data processing using the created program can be a 3D surface with .ply / .obj format, and for cross-section generated 2D data with .jpg / .png format and can be in the form of .dxf data for Autocad software. 3D surface data can be used as BIM data, while 2D cross-section data can be used as built 2D.



This is an open access article under the [CC-BY](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In the development of technology, especially in terms of field review, technology has been developed that can perform real field scanning. This technology is known as a LiDAR (Light Detection and Ranging) Scanner or known as a Laser Scanner. The development of the use of Laser Scanners is an application with the same principles as scanners for 3D printing, only on a different scale of breadth.

BIM (Building Information Modelling) is one of the needs used mainly in the field of civil engineering. BIM contains system technology that includes various information such as design, construction, management, etc. BIM is based on the design of construction, financing, and scheduling, to the implementation and maintenance of buildings. With the existence of a LiDAR scanner, it is hoped that it can be used as a BIM in development.

The use of Lidar Scanner in the world of civil engineering is commonly used for field surveys, the results of the Lidar scanner used are in the form of data point clouds. Several developers and researchers are competing to create programs and research on-point cloud data processing for various purposes. For example, studies in an overview of detection from building point cloud data [1]; in BIM Geometry Creation from Point Cloud [2]; which focuses on identifying point cloud data classification [3] [4] as well as the position and perception of the use of Lidar Scanner in point cloud data retrieval was also further researched and carried out.[5]

2. Method

In this study, the method used was the form of creating a program and then testing the program. The steps carried out are the preparation of sample test data to be used, and other needs for the creation of programs. The sample test

*Corresponding author.
E-mail: akhmadaminullah@ugm.ac.id

<http://dx.doi.org/10.21831/inersia.v19i1.54210>

Received October 31st, 2022; Revised May 29th, 2023; Accepted May 29th, 2023
Available online May 31st 2023

data used is based on point cloud data obtained from the results of field reviews in the form of observations of toll roads along approximately 1 km. As for the need of creating programs needed, such as Visual Studio Code, Open3D libraries, Trimesh, Pymeshlab, etc.

In the data point cloud from the field, it needs to be processed first using the built-in software from the LiDAR scanner used, then adjustments are made to the form of the file format so that it can be processed with other software. In this case, the data point cloud is at home in the form of apply format.

For the creation of a point cloud data processing program, it will be determined what purposes or functions will be created in the program. In the program will be made 2 sub-menus: 1) point cloud (voxel downsampling, outlier downsampling, normalize); 2) 3D model (ball pivoting/Poisson surface, reduce vertex, transform mesh, slice mesh). Each function of the sub-menu has its purpose and use and is operated sequentially to achieve the 3D surface and the 2D cross-section. The entire program will be created with script code in python language. The process step can be seen in the [Figure 1](#).

First step, voxel downsampling is a process where the dense will be regarded as noise, and the program will data point cloud will be minimized or deducted the number of cloud points on an object under review. The purpose of this process is to shrink the number of point clouds and memory of a data point cloud file, without significantly reducing the information on the data. This needs to be done to simplify the next data point cloud process.

Filters using a voxel grid are a method to lower cloud sample points to reduce computation time [6]. Filtering is done by creating a 3D voxel grid, like a small box arranged and stacked. In each voxel, a point approach is carried out either by averaging or by approaching the middle point on each voxel. the illustration of the voxel downsampling process can be seen in [Figure 2](#).

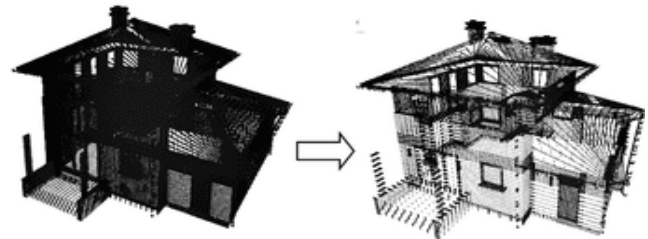


Figure 2. Voxel Downsampling Point Cloud [7]

The logic or algorithm of this downsampling voxel process, in general, is like doing point cloud compression where the cloud point data will be reduced by the number of points, without eliminating the information in it. The parameter used is the amount of voxel applied to the point cloud for compression.

Second step, outlier reduction aims to eliminate noise or cloud point data that interferes or is not used, as a result of inaccurate field reviews so that other point clouds are not included in the object of review. This function relies on the density and proximity between point clouds as its parameter. Cloud point data that are far apart or not dense will be regarded as noise, and the program will automatically remove it. This outlier will be applied to the object under review, after going through the previous downsampling voxel process.

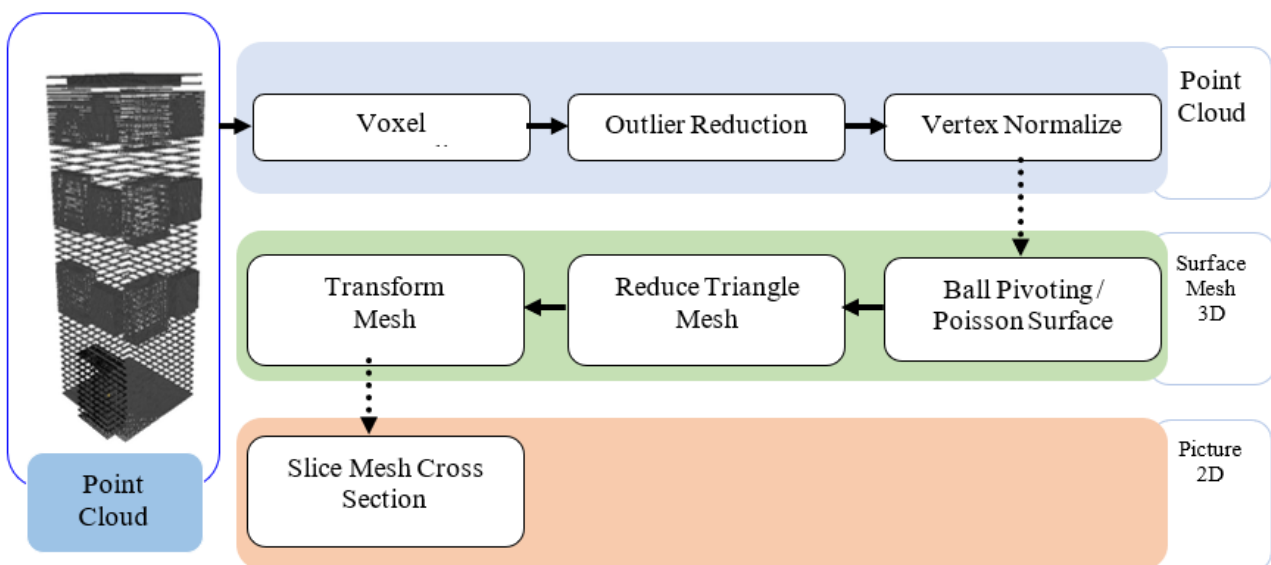


Figure 1. Steps in The Process of Changing Data from Point Cloud To 3D And 2D

The elimination of this statistical outlier is done with a point cloud library to calculate the average and standard deviation of the distance from each point in the environment [8]. Then, remove the points that are outside the specified range.

Using python language, the library used for this function is the Open3D library. The logic or algorithm in this function, if there is a point in the point cloud object data that is not included in the object under review, will be considered noise (disruptive data). This function aims to eliminate noise in the data point cloud under review. This detection is done by using an outlier script to remove or remove points that are not monitored or automatically observed objects. The illustration of the outlier reduction process can be seen in Figure 3.

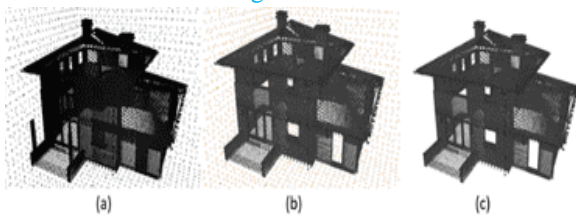


Figure 3. Outlier Reduction Process;

(a) Point Cloud Data; (b) Detect Noise Point Cloud; (c) Deleting Noise [7]

Third step, after going through the previous process (voxel downsampling, outlier reduction) the data point cloud has the possibility of experiencing damage, either in the loss of several point clouds on a certain side, or damage to the position of the shape of the object data. There needs to be a function to re-construct the point cloud data. This normalizes functions specifically to repair the point cloud data that has suffered damage to the data point cloud under review.

Normalization is a process related to relational data models that serve to organize datasets with high or close dependence and interrelationships [9]. In another sense, normalization is a parameter used to avoid duplication of a database table and is also the process of completing a data table that still has an anomaly of unnaturalness resulting in a simple and structured data table. The illustration of the outlier reduction process can be seen in Figure 4.

This function serves to reconstruct the point cloud data that has been damaged due to the editing process. After going through several processes such as voxel downsampling and outlier reduction, it is likely to

experience some changes in the data point cloud, such as point shifts, loss of some points in certain parts, etc. that make the shape of the data point cloud damaged. This function is created using python script code, with the Open3D library.

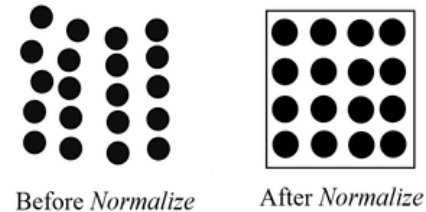


Figure 4. Illustration of Normalize Point Cloud Data

Fourth step point cloud data will be changed to another form of data, namely 3D data surface. There are 2 methods that can be used in the process of forming a data surface, namely ball pivoting, and Poisson surface. Ball pivoting is a method that is likened to a ball that crosses the surface of the point cloud, then it will fix the surface. Meanwhile, the Poisson surface method is likened to enveloping the data point cloud with fabric and forming a 3D surface. The shape of the perfection of the 3D surface is highly dependent on the density, accuracy, and position between point clouds. The more data point cloud, the more complex the data point cloud, and the larger the data will affect the formed surface and the length of formation required. The illustration of the ball pivoting method process can be seen in Figure 5, and the illustration of the poisson surface method process can be seen in Figure 6.

The Ball Pivoting Algorithm (BPA) is one way to implement alpha forms, which are necessary for linear approximation of the original form and then designed for shape reconstruction from an unorganized set of points. The main idea or main logic of this BPA is to put a circle on the set of points and roll it along the outside or surface of the applied points. If the set of points passed by the circle (sphere) does not intersect/touch other points again then the surface will be considered as an edge [10]. Poisson Surface Reconstruction (PSR) is a surface reconstruction technique where the point cloud is a supporting point for the Poisson equation. PSR affixes a collection of oriented points and relies on the function of the 3D indicator

Cloud point data that is ready to use (no noise, no point stacking, etc.), will be converted into 3D mesh data. Writing this script code in python language with Open3D library

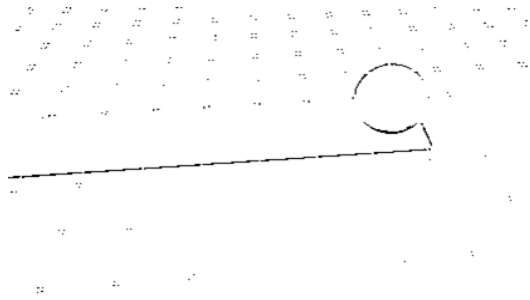


Figure 5. Illustration of Ball Pivoting Point Cloud Data

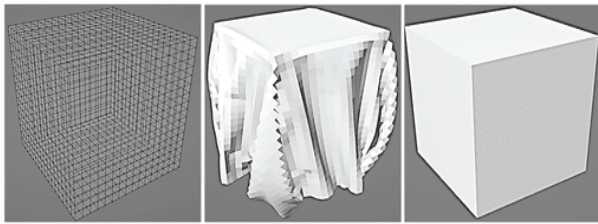


Figure 6. Illustration of Poisson Surface Point Cloud Data

Fifth step, on the 3D surface is a form of unification of each data point cloud that is close to each other and forms triangular lines or what is known as a triangle mesh. The more point clouds, the more triangle mesh is formed. On the surface, this triangle mesh line determines the shape of the surface object and the amount of memory in the surface data file.

The depletion/reduce algorithm aims to reduce the number of triangles in the triangle mesh, keep the original topology, and approach it with the original geometry [10]. A simple thinning algorithm, on all mesh nodes, is candidates or falls within the criteria to be eliminated but can also determine the specified depletion criteria. The resulting hole or distance on the mesh will be patched by forming a local triangulation.

The main function of reducing vertex is to reduce or minimize the triangle mesh line on a surface object. The parameter used in this function is to determine the number of triangles meshes to be applied or targeted to an object. Determining this target, if the smaller it will damage the shape of the object on the surface, it is necessary to experiment first until the maximum limit of reduced triangle mesh is applied. The script code is written using the python language and the pymeshlab library. The illustration of the reduce triangle mesh process can be seen in Figure 7.

Sixth step, in the data point cloud, it has its position coordinates, object rotation, and object scale. This function makes it possible to change these 3 things, according to the needs of the next data process. This change is usually used

when editing surface meshes in other editing applications, so it will make it easier to edit data.

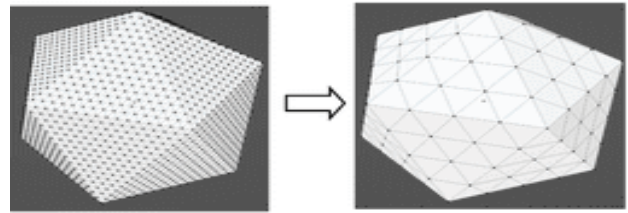


Figure 7. Illustration of Reduction Triangle Mesh Data

Surface mesh, both formed from data point clouds and other surface mesh data, is not completely in the center position with x, y, and z axes (0, 0, 0), and the object is not in a flat position (rotate), and the object is not at a scale that matches the shape of the actual object size (scale). The script code is written using the python language and the Open3D library. The illustration of transform mesh coordinate process shows in Figure 8.

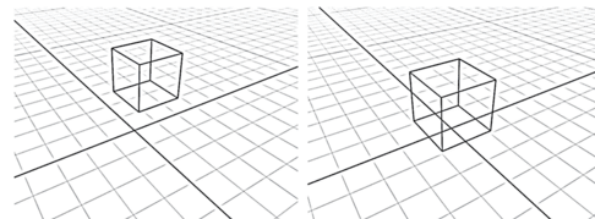


Figure 8. Illustration of Change Coordinates Mesh to The Center

Then, each data point cloud and mesh have a different position directed towards the x, y, and z axes. For example, there is a front face direction on the y-axis, the side on the x-axis, and the upper direction on the z-axis. Or in another example, the data file has a front face direction on the axis z, a side direction on the y-axis, or the upper direction on the x-axis. The illustration of mesh rotate process can be seen in Figure 9.

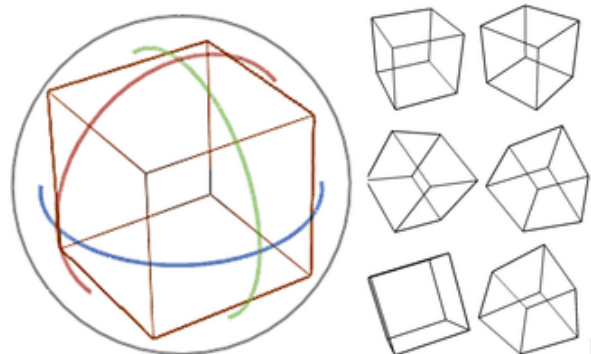


Figure 9. Illustration of Rotate Mesh to The Center

Point cloud data and mesh data, both taken by direct observation in the field, and obtained with other media, have different sizes of magnitude. The case that occurs is usually the size of the data does not match the original or even if 2 data are compared to having different sizes from each other. This is what is referred to as scale on the data. This scale determines the size of each object's data. The illustration of scale mesh process can be seen in [Figure 10](#).

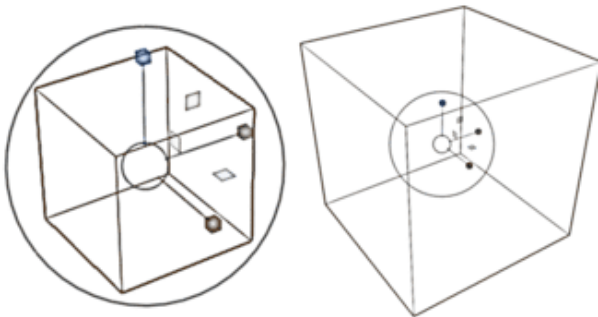


Figure 10. Illustration of Scale Mesh to The Center

Last step, for the change of data from a 3D surface to a 2D cross section or known as a cut on the surface, this slice mesh function is used for this. The cutting of this surface is parameterized on the x, y, and z axes as the axis of the direction of the cut to be formed. Then it is very necessary to place the coordinates on the surface object to the coordinate positions of the x, y, and z axes (0,0,0) or in the central position. This will facilitate the slice mesh process so that cutting can be done to the wishes.

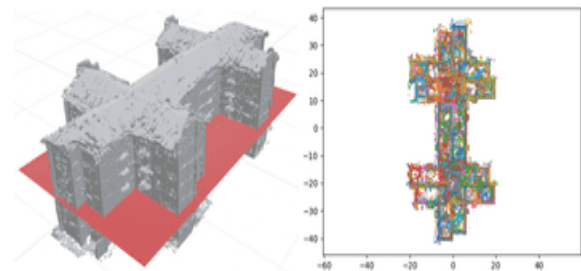
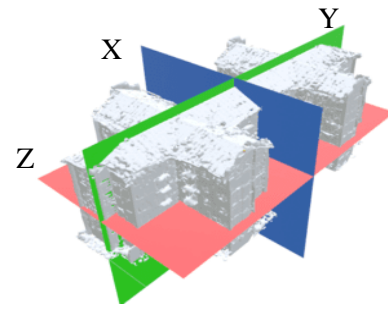


Figure 11. Illustration of Slice Mesh Data [11]

Another thing to note is the rotation or direction of the position of the object (front, back, up, down) against the x, y, and z axes, this direction of rotation will determine which pieces you want to make through the direction of that axis. The result of the slice mesh piece will be a 2D drawing .png/ .jpg or in the form of a .dxf file so that it can be operated in Autocad. The writing of this script code is done with the python language, and with the trimesh library. The illustration of slice mesh process can be seen in [Figure 11](#) and the overview in [Figure 12](#).

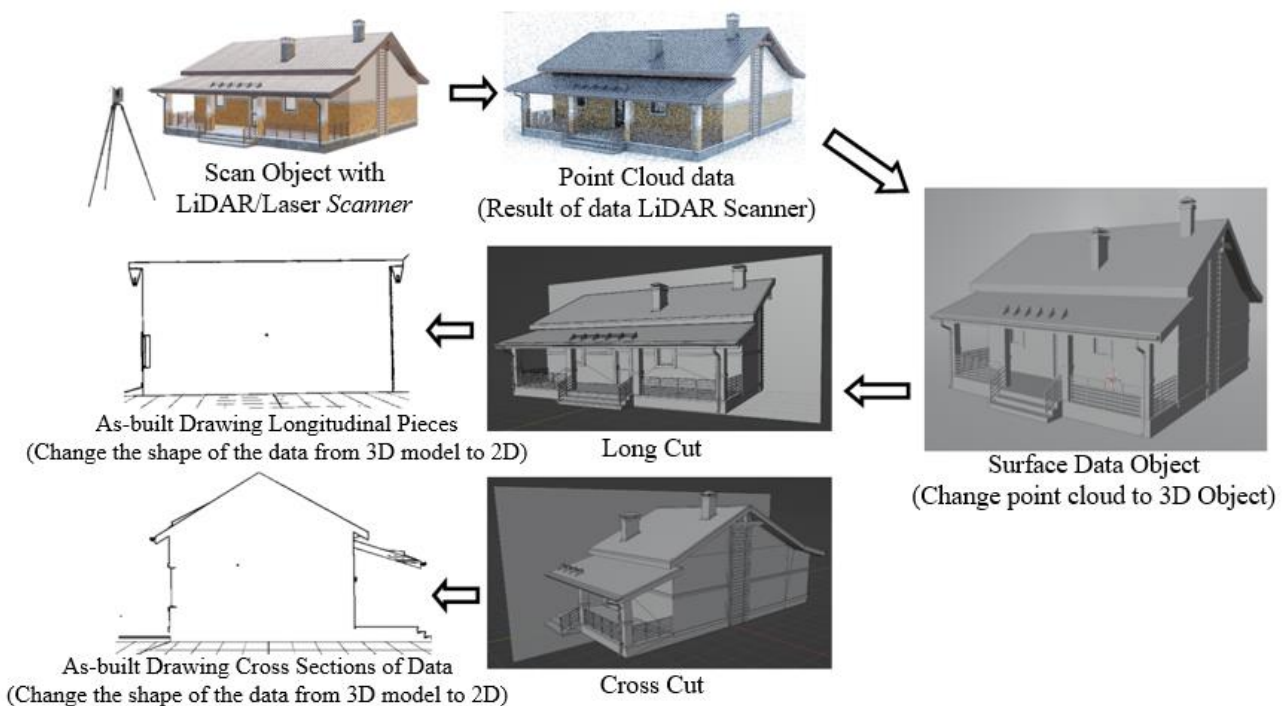


Figure 12. Overview of The Whole Process of Changing Data [12]

3. Results

The point cloud data used comes from direct field observations, in the form of toll road point cloud data in the Sumatra, Sigli - Aceh location. This data was taken by PT Hutama Karya (PT HK) staff using the Leica Pegasus Two Ultimate Mobile Laser Scanner on the Sumatra toll road project. Road observation data has a length of 1 kilometer, with a memory size of 1,703,714 KB, and a total point cloud of 44,733,394 points. The collection of cloud point data is intended to facilitate the work of Building Information Modelling (BIM), and work reports As-Built 2D effectively, and efficiently, save time and minimize the number of fields surveyors. Programs that have been created and have been tested with various other construction data point clouds, will be carried out with data processing with the data point cloud reviewed. This data processing process has several steps, and each step has its function to create a 3D surface and a 2D cross-section.

In this process, the parameter used is the amount of voxel specified, then it will be applied to the field cloud data point object. This test will include the large voxel applied are 0.2, 0.5, 0.8, and 1. The larger the size of the voxel applied, the less the number of point clouds in the data. The results of the data change process can be seen in Figure 13.

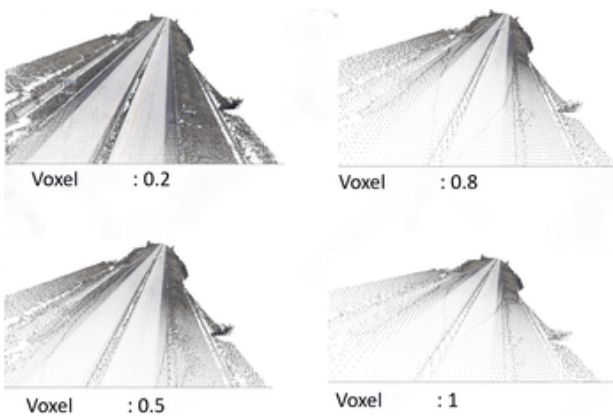


Figure 13. Voxel Downsampling Point Cloud Data

There are 2 methods of removing noise outlier reduction in this test, namely outlier reduction statistical, and outlier reduction radius. In the first method (statistical outlier), the data processing process uses the proximity of neighbors (neighbors) and standard ratios as its parameters. In this test, variations were carried out using neighbors' values: 5, and 10, while for the standard ratio: 1, and 2.

The results obtained can be seen in Figure 14. In the selection of parameters used, the smaller the coverage or standard ratio that is the coverage area, the more detailed

the outlier reduction process that works. However, it will become more and more the process of the work of the program being executed.

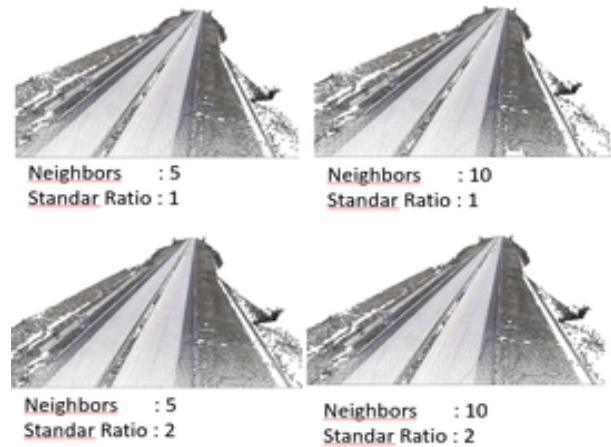


Figure 14. Outlier Reduction Statistical Point Cloud Data

Furthermore, in the outlier reduction method, the ratio of the parameters used is a point that shows the number of points in an area, and a radius size which is the area in question as a limit for calculating the number of points determined. In this test, the parameter values used are points: 5, and 10, while for radius sizes: 1, and 2. The results of the test can be seen in Figure 15.

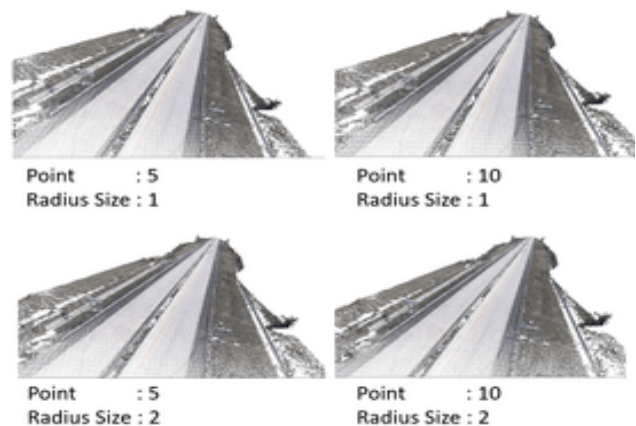


Figure 15. Outlier Reduction Radius

To reconstruct or repair the data point cloud, it is necessary to carry out the normalization process. This is done after going through the previous process, due to the reduction of the amount of data point cloud or the elimination of cloud point noise. This function is used as an anticipation in the event of a stretch or hole in the data point cloud. The parameters used in this normalization are radius size (the region to be reviewed), and max nearest neighbor (the maximum number of points in the region under review). In this test, the radius size and max nearest neighbor values entered have no effect (producing the same result), so the amount of value applied is not in question. Figure 16 show the normalizing process results on the data point cloud.



Figure 16. Vertex Normal Data Point Cloud; (a) x, y, and z axes of each Point Cloud; (b) Review After Normal Vertex

The formation of surfaces from data point clouds is carried out with several methods of application. Ball Pivoting is one of the widely used surface formation methods. This method is usually used for surfaces whose scope is quite small or objects that are quite small. The parameter used in this method is the size of the ball that will be the intermediary for the formation of the surface. In this test, the ball size parameters used were 1, 3, 5, and 7. The results of the formation can be seen in Figure 17. From the results of the variation in the size of the ball applied, the smaller the ball size used, the more obvious the surface result will become. Vice versa, the larger the size of the ball used, the surface results are not very clear.

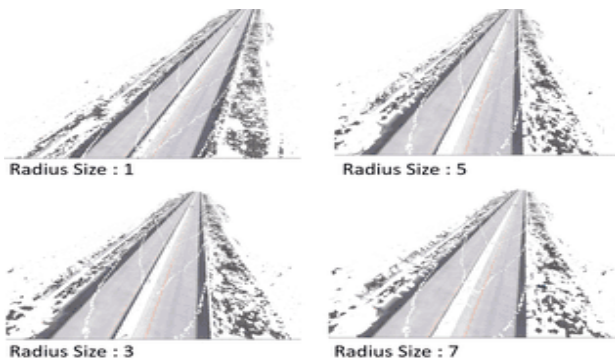


Figure 17. Ball Pivoting Data Point Cloud

One of the other methods that are often used for surface formation is with a Poisson surface. This method can be used for outside and large cloud point data coverage. This is because this method is likened to enveloping the surface of the data point cloud thoroughly and then wrapping it. The parameters used in this method are depth size, and density (densities). Depth aims for the level of detail to be given, while densities are used for the density of surface wrapping after depth wrapping is performed. In this test, the parameters applied; (1) dept size: 10, densities: 0.2; (2) dept size: 12, densities: 0.2; (3) dept size: 14, densities: 0.2; (4) dept size: 16, densities: 0.2. The greater the depth entered, the more detailed it is, and the more density it enters, the more eroded the edges produced.

The presence of triangle mesh on the surface can cause the

length of the program process to run on the surface, as well as large memory size. This is due to many triangles, and it is very tight. Then it is necessary to reduce the number of triangles so that the subsequent programming process is not heavy and does not take up a large enough memory. The parameter used in this process is the target triangle to be achieved. The smaller the target entered, the more damage it will cause damage to the shape of the data surface. In this test, the target triangles included are 5,000,000, 1,000,000, 800,000, and 500,000 triangles on the surface, as can be seen in Figure 19.

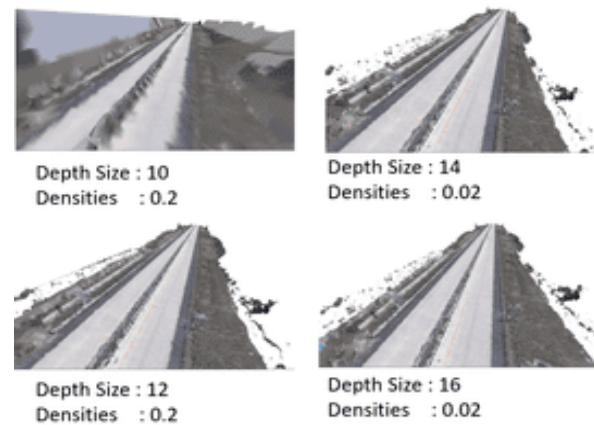


Figure 18. Poisson Surface Data Point Cloud

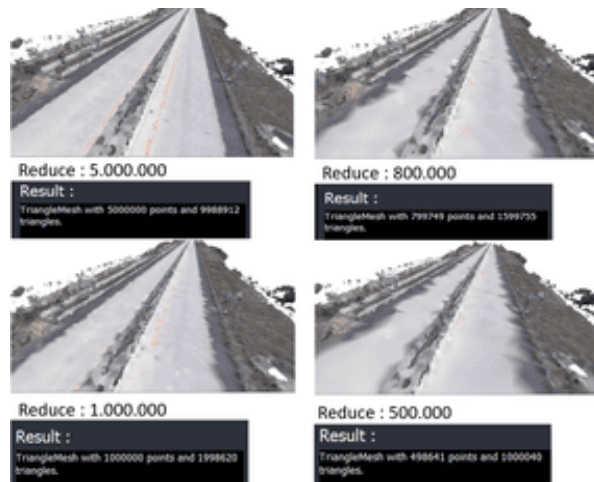


Figure 19. Reduce Vertex Data Point Cloud

In the process of forming a surface, each data point cloud has coordinates against the x, y, and z axes. These coordinates come from where the point cloud data is taken or from field observations, the axis shows the direction of the location or elevation point at the original field size. But in this test, it is necessary to convert these x, y, and z-axis coordinate points into the central axis of 0,0,0. It is necessary for the subsequent process of cutting or slicing mesh. The parameters used are new coordinate points, that is, the x-axis becomes 0, the y-axis becomes 0, and the z-axis becomes 0.

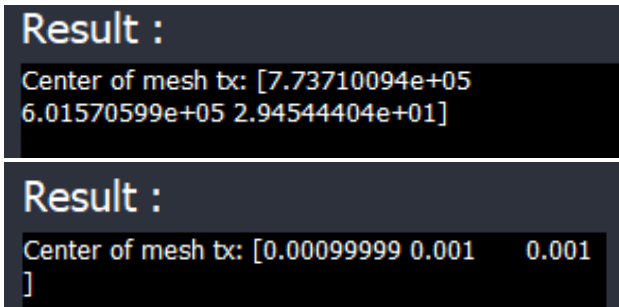


Figure 20. Transform Coordinate Mesh Data Point Cloud; (up) First Coordinate; (down) Coordinate After Changing

The surface will be made into a 2D shape for cross sections or pieces on objects. The process of changing this shape by cutting the surface by relying on the x, y, and z axes as a function of cutting the surface. The parameter in this process is the direction of the axis of the cut to be formed, it will be sliced on the x-axis, the y-axis, or the z-axis. The result of the cutting will be saved in 2D form with .jpg / .png format, or .dxf. Cropping on the surface can be seen in [Figure 21](#) for the cut in the x-axis direction, [Figure 22](#) for the cut on the y-axis, and [Figure 23](#) for the cut on the

z-axis.

Slicing is the process of changing a 3D model into a series of a report containing the boundaries of the 2D perimeter where there is a closed loop of connected points [7]. There are 2 operational steps in the slicing process, namely calculating the intersection between the triangle and the wedge plane, then the next step is to sort the intersection into contiguous line contours.

After going through various processes from voxel downsampling, outlier reduction, normalize, then the formation of a 3D surface, the reduction triangle mesh, and transform, it is necessary to ascertain whether there is a change in the shape of the data from the initial data point cloud with the surface mesh created. To ensure this, it is necessary to compare the two data, then assess the difference. This process is carried out using CloudCompare software, to compare point cloud and mesh. The results of the comparison can be seen in [Figure 24](#), it can be seen that there is no significant difference between the point cloud and the mesh.

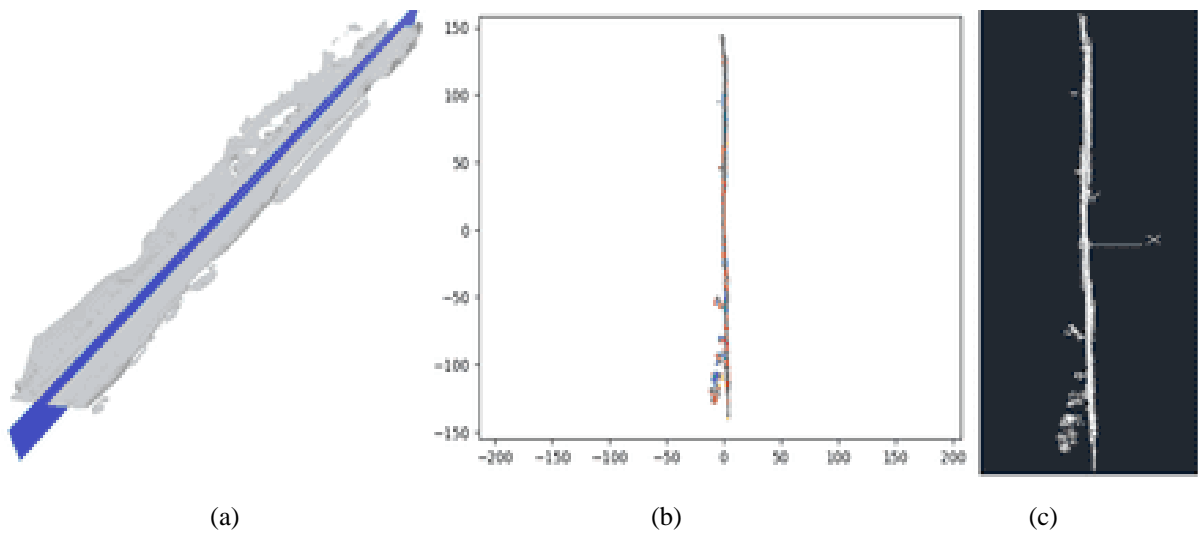


Figure 21. Cutting Results / Slice Surface Mesh Against x Axis On Field Cloud Data Point; (a) Cross Section on BIM; (b) Cross Section with png/jpg Format; (c) Cross Section with .dxf Format

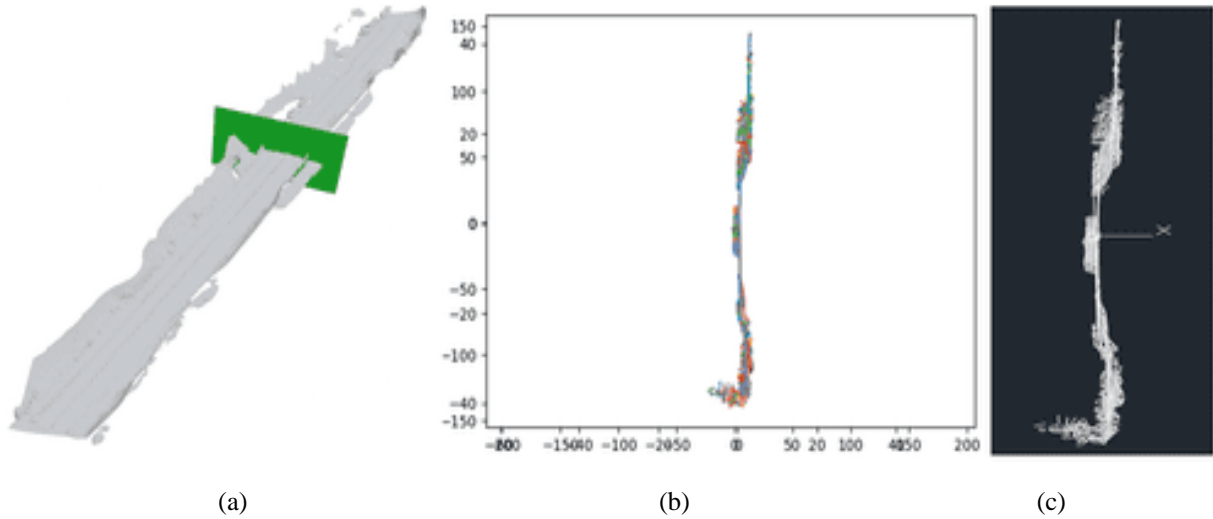


Figure 22. Cutting Results / Slice Surface Mesh Against y Axis On Field Cloud Data Point; (a) Cross Section on BIM; (b) Cross Section with png/jpg Format; (c) Cross Section with .dxf Format

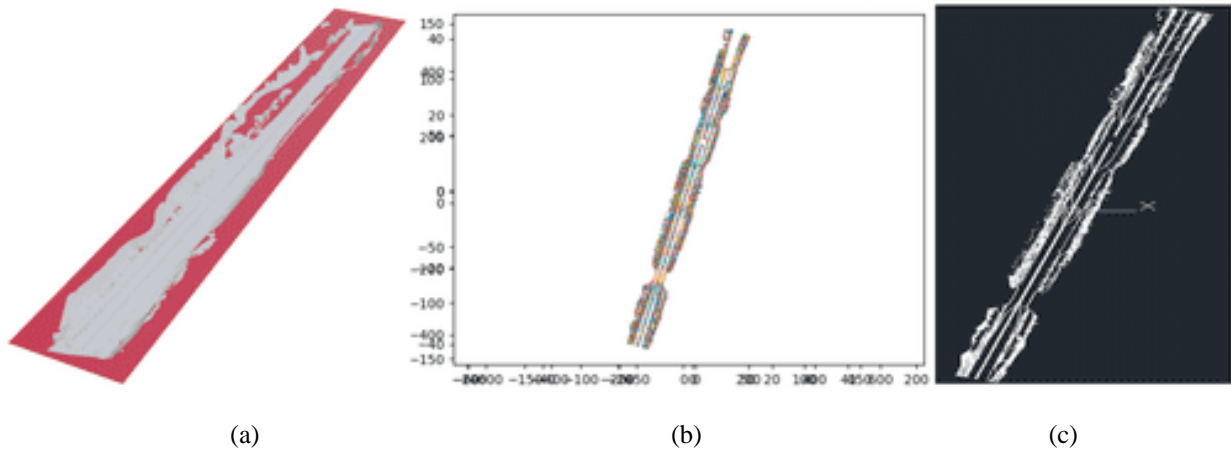


Figure 23. Cutting Results / Slice Surface Mesh Against y Axis On Field Cloud Data Point; (a) Cross Section on BIM; (b) Cross Section with png/jpg Format; (c) Cross Section with .dxf Format

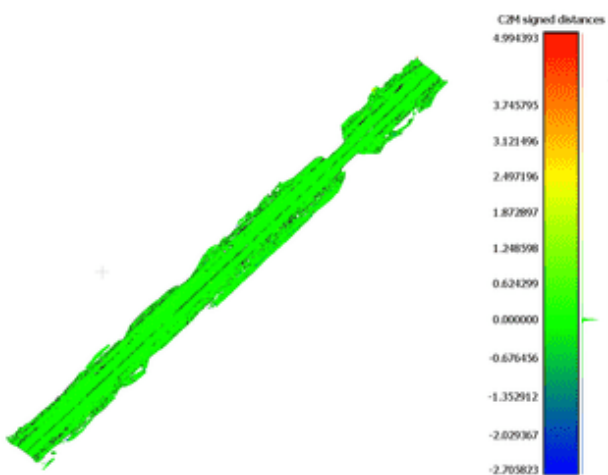


Figure 24. Comparison Between Initial Data Point Cloud And 3D Surface Mesh in CloudCompare

The comparison results on CloudCompare show that the comparison display, the green color indicates there is no difference in the distance between the two data or is worth 0.0000 m. By looking at the results of the comparison between the two data, it can be concluded that there is no difference in changes in the shape of the data, in terms of coordinates, data volume size, and data elevation so that the data processing program that has been made can be used properly.

4. Conclusions

From the research conducted regarding the processing of cloud and surface point data, it can be concluded: (1) Creating programs used for data point clouds, using python script code with Open3D, Trimesh, and Pymesh

libraries. From the script code written, special software was created that aims to facilitate the process of use in processing data point clouds. (2) Processing point cloud data in the field with the software made, it is necessary to adjust the form of the file format so that it can be processed on the software.

For data point cloud, the software created can only process in the format: .xyz, .xyzn, .xyzrgb, .pts, .ply, .pcd, while for surface processing 3D processing in formats: .ply, .stl, .obj, .off, .gltf/glb; (3) In the software, several processes are carried out which are divided into 2, namely; (a) Point cloud (Processing of point cloud data), with the process of voxel downsampling (reduction of the number of point clouds), outlier reduction (removal of disruptive cloud point noise), and normalise (reconstruction of point cloud data); (b) 3D Model (3D surface data processing), with ball pivoting / poisson surface process (formation of point cloud to 3D surface), reduce triangle (reduction of triangle mesh lines on the surface), slice mesh (cutting / cross section on the surface to 2D .jpg / .png / .dxf), and transform mesh (change of center coordinate, rotate, and scale); (4) The results of field cloud point data processing with the software created, in the form of 3D data surfaces with .ply/.obj format, and cross section 2D data with .jpg/.png/.dxf formats.

References

- [1] H. Mansor, S. A. A. Shukor, and R. Wong, "An overview of object detection from building point cloud data," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jun. 2021. doi: 10.1088/1742-6596/1878/1/012058.
- [2] K. Daria, "Kosavchenko Daria BIM Geometry Creation from Point Clouds, 64 pages."
- [3] R. Jayakumari, R. R. Nidamanuri, and A. M. Ramiya, "Object-level classification of vegetable crops in 3D LiDAR point cloud using deep learning convolutional neural networks," *Precis Agric*, vol. 22, no. 5, pp. 1617–1633, Oct. 2021, doi: 10.1007/S11119-021-09803-0.
- [4] M. Arastounia and D. D. Lichti, "Simultaneous identification, modeling and registration refinement of poles using laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 181, pp. 327–344, Nov. 2021, doi: 10.1016/J.ISPRSJPRS.2021.09.017.
- [5] C. Benedek, A. Majdik, B. Nagy, Z. Rozsa, and T. Sziranyi, "Positioning and perception in LIDAR point clouds," *Digit Signal Process*, vol. 119, p. 103193, Dec. 2021, doi: 10.1016/J.DSP.2021.103193.
- [6] J. Jonas, "3D Smart Sensor," 2017.
- [7] "Bambo House Free 3D Model - .obj .blend .fbx - Free3D." <https://free3d.com/3d-model/bambo-house-47896.html> (accessed May 31, 2023).
- [8] H. Liu, Y. Zhang, L. Lei, H. Xie, Y. Li, and S. Sun, "Hierarchical Optimization of 3D Point Cloud Registration," *Sensors 2020, Vol. 20, Page 6999*, vol. 20, no. 23, p. 6999, Dec. 2020, doi: 10.3390/S20236999.
- [9] Z. Efendy, "NORMALISASI DALAM DESAIN DATABASE," *Jurnal CoreIT*, vol. 4, no. 1, 2018.
- [10] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," pp. 65–70, Jul. 1992, doi: 10.1145/133994.134010.
- [11] "DEMO DATA | Drupal." <http://www.merko.lv/en/demo-data> (accessed May 31, 2023).
- [12] "GameReady Cottage FREE - Download Free 3D model by Gerhald (@Gerhald) [9203e8f]." <https://sketchfab.com/3d-models/gameready-cottage-free-9203e8f4b67b4d9e8b32c10b2d5e6a57> (accessed May 31, 2023).