Persistent Homology: Hole Detection in LiDAR Point Clouds with Topological Data Analysis

by

Suen Wun Ki, Christine

Supervisor: Dr. Wai Yeung Yan

Moderator: Prof. Bo Wu

A dissertation

submitted to The Hong Kong Polytechnic University

in partial fulfilment of the requirement

for the degree of

Bachelor of Science (Honours)

In the Program of

Geomatics (Land Surveying)

Date

15th May 2021

Certificate of Originality

I hereby, declare that this dissertation is my own work and that, to the best of my knowledge and belief, it contains no material which has been accepted or submitted for the award of any other degree or diploma, except where due reference is made in the text of the text.

Abstract

Laser scanning has been widely used in various applications because of its efficiency and accuracy. However, a common problem found in laser scanning is that there are holes contained in the point clouds, caused by the weak or missing return signal of the laser scanner. Point clouds with hole can lead to an improper result. Hence, hole detection is needed before the use of laser scanning data. The existing hole detection methods mostly require sufficient user intervention or have strict restriction on the data type being detected. In this regard, this thesis attempts to fill the research gap by proposing a hole detection approach based on Persistent Homology, a foundational tool of Topological Data Analysis. Persistent Homology measures the topological features (e.g., holes) in the point clouds by the filtration across simplicial complexes. With the process of filtration, no threshold value is needed for the accurate result. The use of Persistent Entropy also significantly enhances the efficiency of the hole detection process. To assess the performance of Persistent Homology on hole detection, Persistent Homology is applied in both airborne point clouds and terrestrial point clouds with hole in regular shape, irregular shape and hole caused by occlusion. The result shows that Persistent Homology performs well in identifying and locating the hole in point cloud with less than 1% difference between the hole detected and the hole contain in the point cloud in terms of area. The detection result is further analyzed for more applications. In the second part of the thesis, hole classification is conducted using the Maximum Likelihood Estimation. The classification is carried out by applying the data of the hole detected from Persistent Homology, including the longest distance between any two points, 3D area and the Z's standard deviation of the hole. The model successfully classifies three types of hole (hole of window, occlusion of building and rooftop) with 86% of accuracy. The possibility of classifying the hole in the point cloud based on Persistent Homology by machine learning is demonstrated. The outcome of this thesis suggests a new direction of hole detection and enhances the development of point cloud processing in laser scanning.

Acknowledgement

From the design of the topic to the submission of the dissertation, I have received so much encouragement and support from many people, and I would like to express my sincere appreciation to them.

Dr. Wai Yeung Yan is my supervisor. Inviting him as the supervisor is the cleverest choice I have made. He is an extremely inspiring and encouraging person. Besides providing idea and suggestions about the dissertation, he teaches me a lot about the conduct of life. He concerns much more about my future and career pursuit than the progress of the dissertation. Whenever I finished the meeting with him, I became very confident because of his encouragement. I feel thankful for his teaching. He is a role model of life that I would like to be someone like him. Another person I highly appreciate is Leo, who is wise and helpful. He taught me the operation of laser scanner and the processing work step by step. He gave me the call when I need help or being frustrated. He listened to my confusion and provided me suggestions. Leo had helped me so much that it cannot have written them all.

I am also thankful to Ms. Tai Kai Wing, Ms. Sandy Wan and Mr. Anson Yip. They provided me lots of inspiration when I was designing the topic of the dissertation. They treated it as serious as their dissertation. I am also happy to have a buddy like Ms. Tai Kai Wing, she assisted my laser scanning survey even in her busy time. She is also the best study partner that I love to discuss the obstacles of the dissertation with her. New idea usually popped up after having an intense discussion with her.

I would like to express my highest gratitude to my family, especially my father. He supports every decision I made and encourages me to study hard. He always loves and tolerates his rebellious daughter, even he never speaks it out. Last but not least, I appreciate Mr Chong Chi Man for his patience and consideration. He knows me the best and supports me when I was stressed.

I feel blessed with all these kindnesses received.

Contents

Certificate of Originality
Abstract
Acknowledgement
Chapter 1 1
1. Introduction 1
1.1 Research motivation 1
1.2 Research objective 2
1.3 Organization of the dissertation 3
Chapter 2 4
2. Literature Review 4
2.1 Hole Detection 4
2.2 Persistent Homology 5
2.2.1 Simplicial Complex 6
2.2.2 Filtration 8
2.2.3 Persistence Diagram 9
2.3 Discussions 10
Chapter 3 11
3. Hole Detection based on Persistent Homology 11
3.1 Methodology 11
3.1.1 Overall workflow 11
3.1.2 Detection of Topological Features 13
3.1.2.1 Persistent Homology 13
3.1.2.2 Greedy Permutation Test 15
3.1.3 Extraction of Significant Holes 16
3.1.3.1 Persistent Entropy 16
3.1.3.2 Output of Significant Holes 19
3.1.3.3 The Efficiency of the Flow 20
3.2 Experimental Testing 21
3.2.1 Testing Dataset 21
3.2.2 Design of Experiment and Evaluation 22
3.3 Results and Analysis 23
3.3.1 Airborne Laser Scanning 24
3.3.1.1 Hole of Regular Shape Object 24
3.3.1.2 Hole of Irregular Shape Object (Occlusion) 29
3.3.2 Terrestrial Laser Scanning 33
3.2.2.1 Hole of Regular Shape Object 33
3.2.2.2 Hole of Irregular Shape Object 35
3.2.2.3 Occlusion 36
3.3.3 Analysis of Hole Detection Result 38
3.3.3.1 Identification of hole 38
3.3.3.2 Performance of Hole Detection between ALS and TLS 39
3.4 Chapter Summary 40

Chapter 4 41
4. Hole Classification based on Maximum Likelihood Estimation 41
4.1 Methodology 41
4.1.1 Overall Workflow 41
4.1.2 Data Training 43
4.1.2.1 Determination and Collection of Feature Sets 43
4.1.2.2 Computation of Mean Matrix and Variance-Covariance matrix.. 47
4.1.3 Classification by Likelihood Model 48
4.1.3.1 Calculation of Probability 48
4.1.3.2 Comparison and Classification 49
4.2 Experimental Testing 50
4.2.1 Design of Experiment and Evaluation 50
4.2.2 Training Data 51
4.2.2.1 Window 51
4.2.2.2 Occlusion of Building 53
4.2.2.3 Rooftop 55
4.3 Results and Analysis 58
4.4 Chapter Summary 62
Chapter 5 63
5. Conclusions and Future Works 63
5.1 Conclusions 63
5.2 Future Works 64
References 65

List of Tables

Table 1: Comparison between area of hole detected and area of hole in point cloud 38
Table 2: Information of training data 57
Table 3: Classification result of hole 62

List of Figures

Figure 1: (a) Hole; (b) Void 6
Figure 2: (a) A point (0-simplex); (b) Line Segment (1-simplex); 6
Figure 3: A simplicial complex 7
Figure 4: (a) Cech Complex, the triangle form when there is a point with triple intersection of three balls; (b) Vietoris-Rips Complex, the triangle form when three edges connect with each other. 7
Figure 5: (a) Simplicial Complex with a suitable $\in$; 8
Figure 6: Filtration 8
Figure 7: Persistence Diagram 9
Figure 8: Overall workflow of hole detection (Chapter 3) 12
Figure 9: (a) Plot of original point cloud (1000 points); (b) Persistence diagram of original point cloud 15
Figure 10: (a) Plot of sampled point cloud (500 points); (b) Persistence diagram of sampled point cloud 16
Figure 11: Computational performance of two workflows 21
Figure 12: Hole Detection result of Swimming Pool 24
Figure 13: Hole Detection result of Road 25
Figure 14: Hole Detection result of Rooftop 26
Figure 15: Hole Detection result of Water Tank 27
Figure 16: Hole Detection result of Dam 28
Figure 17: Occlusion detection result of Smoke 29
Figure 18: Occlusion detection result of Cloud 30
Figure 19: 3D view of building with occlusions (red arrow) 31
Figure 20 Occlusion detection result of building 31
Figure 21: Occlusion detection result of tree 32
Figure 22: Hole detection result of window 33
Figure 23: Hole detection result of door 34
Figure 24: Hole detection result of puddle 35
Figure 25: Detection result of occlusion caused by word on wall 36
Figure 26: Detection result of occlusion caused by roadblock pillar 37
Figure 27: Noise recognized as points of topological hole (red arrow) 40
Figure 28: (a) and (b) Point cloud with noise in the middle; (c) and (d) Point cloud without noise 40
Figure 29: Overall workflow of hole classification (Chapter 4) 42
Figure 30: Two holes with same area but different longest distance 43
Figure 31: (a) Original point cloud of Dam; (b) Meshed point cloud of Dam 45
Figure 32: (Training data) Hole of window 1 with 1.56m in longest distance, 1.14m^2

in 3D area and 0.27 in standard deviation of Z 51
Figure 33: (Training data) Hole of window 2 with 2.39m in longest distance, 1.13m^2 in 3D area and 0.80 in standard deviation of Z 52
Figure 34: (Training data) Hole of window 3 with 2.61m in longest distance, 2.80m^2 in 3D area and 0.74 in standard deviation of Z 52
Figure 35: (Training data) Hole of window 4 with 2.11m in longest distance, 3.21m^2 in 3D area and 0.60 in standard deviation of Z 52
Figure 36: (Training data) Hole of window 5 with 1.59m in longest distance, 1.80m^2 in 3D area and 0.58 in standard deviation of Z 53
Figure 37: (Training data) Hole of building's occlusion 1 with 31.92m in longest distance, 616.67m^2 in 3D area and 3.34 in standard deviation of Z 53
Figure 38: (Training data) Hole of building's occlusion 2 with 34.24m in longest distance, 819.09m^2 in 3D area and 2.91 in standard deviation of Z 54
Figure 39: (Training data) Hole of building's occlusion 3 with 33.77m in longest distance, 691.40m^2 in 3D area and 3.43 in standard deviation of Z 54
Figure 40: (Training data) Hole of building's occlusion 4 with 33.39m in longest distance, 879.82m^2 in 3D area and 2.97 in standard deviation of Z 54
Figure 41: (Training data) Hole of building's occlusion 5 with 20.95m in longest distance, 584.33m^2 in 3D area and 5.01 in standard deviation of Z 55
Figure 42: (Training data) Hole of rooftop 1 with 41.01m in longest distance, 877.54 m^2 in 3D area and 1.62 in standard deviation of Z 55
Figure 43: (Training data) Hole of rooftop 2 with 12.22m in longest distance,66.93m^2 in 3D area and 0.43 in standard deviation of Z 56
Figure 44: (Training data) Hole of rooftop 3 with 40.63m in longest distance, 986.64m^2 in 3D area and 1.68 in standard deviation of Z 56
Figure 45: (Training data) Hole of rooftop 4 with 80.52m in longest distance, 1724.64m^2 in 3D area and 1.94 in standard deviation of Z 56
Figure 46: (Training data) Hole of rooftop 5 with 90.63m in longest distance, 2823.98m^2 in 3D area and 1.02 in standard deviation of Z 57
Figure 47: **Hole of window classified as 'window'**, with 1.58m in longest distance, 1.19m^2 in 3D area and 0.53 in standard deviation of Z (Result 1)..... 59
Figure 48: **Hole of window classified as 'window'**, with 1.72m in longest distance, 1.31m^2 in 3D area and 0.52 in standard deviation of Z (Result 2)..... 60
Figure 49: **Hole of occlusion classified as 'occlusion'**, with 26m in longest distance, 314.21m^2in 3D area and 4.42 in standard deviation of Z (Result 3) 60
Figure 50: **Hole of occlusion classified as 'rooftop'**, with 22.06m in longest distance, 390.48m^2in 3D area and 3.43 in standard deviation of Z (Result 4) 60
Figure 51: **Hole of rooftop classified as 'rooftop'**, with 65.81m in longest distance,

1941.38m^2 in 3D area and 2.45 in standard deviation of Z (Result 5) 61
Figure 52: **Hole of rooftop classified as 'rooftop'**, with 58.73m in longest distance, 1484.07m^2 in 3D area and 3.56 in standard deviation of Z (Result 6) 61
Figure 53: **Hole of rooftop classified as 'rooftop'**, with 45m in longest distance, 1294.1m^2 in 3D area and 4.36 in standard deviation of Z (Result 7) 61

Chapter 1

1. Introduction

1.1 Research motivation

Laser Scanning (LiDAR) is a remote sensing method for determining the ranges by measuring the time between a pulse of light projects and reflects. It is flexible and efficient that different type of laser scanning can be adopted depends on the applications. There are three main types of laser scanning, including terrestrial laser scan, airborne laser scan and mobile laser scan. Terrestrial laser scanning (TLS) can measure the entire environment accurately from a fixed position on the ground using the laser scanner, but the setting up and transporting are relatively challenging (Stal, 2020). Mobile laser scanning (MLS) allows the scanning conducted on a moving platform, such as vehicle or operator function as a platform. The acquisition time is largely reduced but the data acquired mostly is not as detailed as TLS. Airborne laser scanning (ALS) utilizes the UAV as the platform, which can acquire data among large area in a short period of time, but the resolution is relatively lower.

Laser scanning is also accurate that the highest accuracy of data collected can reach to mm level. The data laser scanner captures and records is the geometry of visible surfaces of objects and sites (Vosselman, 2010). The data collected are represented in the form of points. The clustered points are called point cloud, which is delivered as list of X Y Z coordinates.

In recent years, with the advance of laser scanning, it has been applied in diverse applications, such as mapping, 3D scene/object reconstruction, spatial information obtaining and entertainment, for more efficient and simpler data acquisition. The importance and practicality of laser scanning can be demonstrated by the example of Smart City. In many well-developed cities, Smart City is proposing or already implementing. One of the foundational tools for Smart City is the digital city models, which means digitizing the whole city, such as buildings, streets and vegetation, into the 3D model. LiDAR is one of the typical and significant technologies to collect the data for 3D modelling because of its accuracy, efficiency and flexibility.

However, a common problem with laser scanning is the hole contained in the point clouds, regardless of the types of laser scan used. According to Liang et al. (2020), the point cloud data often contain areas that are scanned but fail to be collected because the pulse does not return. The missing area in the point cloud data can be a hole. Some hole can be called as occlusion, but both are empty area in the point cloud visually because of the unreceived light beam. They can be caused by several factors.

- Hole:
  - The object has poor reflective property/no reflectance (Fang et al., 2004) that the light refracts or absorbs by the object, rather than reflects.
    - Water surfaces, e.g., river, swimming pool, lakers.
    - Black surfaces as they absorb the light, e.g., asphalt road.
  - The object has high reflective property that the light scatters.
    - Shiny surfaces, e.g., mirror.
  - The object is transparent that the light pass through the surfaces
    - E.g., glasses, window.
- Occlusion:
  - The object being scanned is occluded by another object (Bendels, 2006) so part of the object is not in the scanner's line of sight.
    - Vertical building walls that block by the roof or other buildings.
    - Mountain Overhand that its steep part is invisible to the scanner.

The hole existed can affect the analysis and processing of 3D data point cloud that the applications mentioned above highly rely on. The problems caused by holes depend on the application of point cloud (Fiolka et al., 2017). For example, the hole can lead to missing information in mapping, improper model or surface in reconstruction works, wrong area measurement in spatial information obtaining, and unattractive presentation in entertainment. Imagine numerous empty rectangular spaces appear on the wall of buildings in the digital city because of the transparent glass window. It is a failure in 3D modeling. Therefore, the detection of hole in point cloud is significant that it is the primary work of the analysis and application of point cloud and hole filling cannot be conducted without hole detection. With the technique of hole detection, laser scanning can be applied in different aspects with higher accuracy and efficiency.

1.2 Research objective

In this research, persistent homology, a kind of topology data analysis, is presented to detect the hole in point clouds. The result of Persistent Homology has various applications,

so three application of Persistent Homology would be demonstrated too. Specifically, the objectives of the research are:

1. To detect the hole in point clouds using Persistent Homology
2. To extract and output the topological hole (positional information)
3. To compute the holes' features (area, longest distance between any two points and standard deviation of Z)
4. To classify the type of hole using Maximum Likelihood Estimation

The first objective is to apply Persistent Homology in hole detection of LiDAR point clouds. Further applications are conducted using the detection result, that is the next three objectives. The second objective is to extract the position of the topological hole that represent the objects. This application can be adopted to locate the specific object. For example, the location of water body can be easily extracted with the airborne laser scanning and Persistent Homology. The information of the hole extracted can be computed for analysis. The third objective is to compute the features of hole, including its area, longest distance between any two points and standard deviation of Z. Application like cost computation based on the glass window size can be conducted with the third objective. The fourth objective is to conduct the hole classification based on the result of the first three objective using Maximum Likelihood Estimation. The classification result has various applications. For example, with the classification of type of hole detected, the method of hole filling can be more specific, or the total number of glass window and their area can be automatically computed with the laser scanning of the building. Therefore, this research focus on Persistent Homology in hole detection of point clouds and the applications of the detection result.

1.3 Organization of the dissertation

This thesis is organized as follows: Chapter 2 presents a literature review of the current approach in hole detection and the concept of Persistent Homology. Chapter 3 presents the application of Persistent Homology in hole detection of point cloud with the methodology, experimental setting and the result with analysis. The experiment and result present the hole detection in various type of hole and laser scanning type. Chapter 4 presents the application of hole classification using Maximum Likelihood Estimation with the extraction of holes' information, including longest distance, 3D area and standard deviation of Z. Conclusion of the research are drawn in Chapter 5 with the directions for future work.

Chapter 2

2. Literature Review

2.1 Hole Detection

There is substantial amount of approaches proposed in the area of hole detection in point clouds. A common way is to detect the hole in triangle mesh. It is defined that the hole is a closed loop of boundary edges in meshes (Wei et al., 2010). In Branch (2006), if an edge of a triangle does not share with other triangles, or a triangle does not have any neighbouring triangle, the existence of hole can be identified and traced using the triangle, or using the edge as that is the boundary edge of hole. In Jun (2005), similar approach is adopted, which proposed that vertex does not belongs to two or more triangles should be on the boundary edge of the hole and thus the hole can be found. In Bendels et al. (2006), the probability of each point locates on or near a hole is calculated using a weighted sum by combining three criterias: the angle criterion, the halfdisc criterion and the shape criterion. Where the angle criterion is sharper and better exposes thin lines of boundary points; the halfdisc criterion is better capable of detecting small holes; the shape criterion has the best performance when there is noisy data.

However, using triangulated point cloud to detect hole require manually setting the threshold value for detecting the neighbourhood points to construct the triangular mesh. In Bendels 's method, threshold for parameters, like angle criterion, ellipsoid criterion, weighted average criterion, half-disc criterion and shape criterion, also needed. The hole detection result highly depends on the threshold set, for example, small holes are ignored if the threshold for triangulation is too large. Since each set of point cloud contains different holes, and have different density and distribution of points, specific threshold value has to be set to each set of point cloud. The user intervention in the above methods required is thus considerable.

In some algorithm, threshold for detecting the neighbourhood points also required even the hole is not identified in the triangulated mesh. In Nguyen (2015), the points on boundary of point cloud are extracted and defined as exterior boundary points, the remaining points are called inter points. If the point is an inter point and any of its neighbors, in 4-connectivty, is empty, it is determined as a boundary point of hole. Yet, the detection of the existence of fours neighbors results in tedious and expensive computation.

Some hole detection algorithm require user to select the hole manually. In Yang (2017), the hole boundary is determined by selecting the feature points along the boundary manually first. The segment is then created between two feature points. The new boundary point is determined by choosing the nearest neighbor from input point cloud to the middle point of segment and new segment form with new boundary point detected. The process continues until no corresponding new boundary point is found. Yet, the detection of hole relies on the user intervention. When it comes to situations like there is tiny hole with low visibility to human eyes, the point cloud contains numerous holes or the point cloud is complicated and large, methods required manual selection are ineffective and could not ensure all holes are accurately detected.

Another approach proposed utilize the properties of airborne LiDAR data. For airborne with GPS mounted, the collected data include GPS time of recording along with the XYZ of each point. In Rupnik (2015), it states that the time frames between the consecutive scanned points are expected to be nearly identical with minimal variations because the scanning usually performed at the constant frequency. If there is large deviation between the returning time and the predicted time, hole can be identified. Whereas, it can only apply in scanning the flat area because the sudden change of geometry disrupt the linear distribution of the consecutively scanned points. More importantly, this approach limits to the airborne LiDAR with GPS, so its compatibility and flexibility is relatively low.

Some other methods proposed hole detection by deep learning with the advancement in deep learning. In Tabib (2020), the point cloud data is modelled towards learning hole boundary points and the architecture is trained to use the modelled data in order to model the deep learning framework towards hole boundary detection. Hole detection by deep learning is a fully automatic approach, yet, its interpretability is low that the result highly depends on the trained data and is hard to give the derivation of the result.

2.2 Persistent Homology

Topological Data Analysis (TDA) is a field studying the qualitative features of data by geometry and topology (Otter et al., 2017), with the advantages of robustness to noise and ability to investigate the complex and high dimensional data in low dimensional approximations, while the structure and connectivity information are preserved (Sheffar & Dayten, 2020).

To achieve the above goals, a tool call Persistent Homology (PH) in TDA was developed, which can discern topological features of data using algebraic. From the view of

topological, in PH, topological features (Homology) can be identified include connected components, holes and voids (Figure 1); types of data can be studied include set of discrete points with metric, digital images and network.

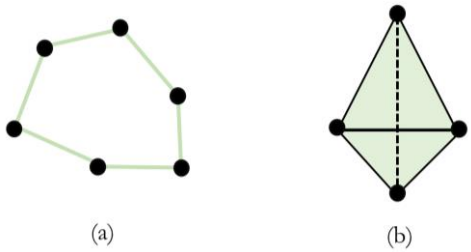


Figure 1: (a) Hole; (b) Void

Discrete points with metric refer to point cloud data. To identify the topological features from sets of point cloud data, there are three steps generally:

1. Apply the **Simplicial Complex (Ch.2.2.1)**
2. Obtain the **Filtration (Ch.2.2.2)**
3. Produce the **Persistence Diagram (Ch.2.2.3)**

2.2.1 Simplicial Complex

In topological, point cloud data do not contain any interesting information because there is no connection between points. Therefore, simplicial complex is first employed to the point cloud as it connects the points for approximating more complicated shapes and is mathematically and computationally tractable. It gives relationship among various points.

The idea of simplicial complexes starts from simplices. A k dimension simplex is the convex hull of k+1 affinely independent points (Fugacci et al., 2016). A 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, a 3-simplex is a tetrahedron and so on.

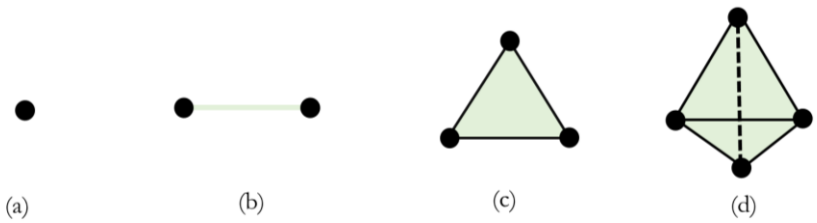


Figure 2: (a) A point (0-simplex); (b) Line Segment (1-simplex);
(c) Triangle (2-simplex); (d) Tetrahedron (3-complex)

A simplicial complex is a finite set of simplices (Fugacci et al., 2016), which is formed by different simplices. In Figure 3, a simplicial complex is formed by the combination of a 2-simplex and a 1-simplex.

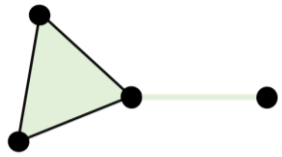


Figure 3: A simplicial complex

There are various construction methods for simplicial complex, such as Cech Complex and Vietoris-Rips Complex. Cech Complex can capture more topological information since it can contain more information than its skeleton graph, while its computational cost is relatively higher so it is infeasible in practical (Dantchev et al., 2012). Because of the infeasibility of Cech Complex, its approximation, Vietoris-Rips Complex (VR Complex), would be used for efficient computation. Its computational cost is significantly lower, even the size of point cloud is enormous (Zomorodian& Afra, 2010).

(a) (b)

Figure 4: (a) Cech Complex, the triangle form when there is a point with triple intersection of three balls; (b) Vietoris-Rips Complex, the triangle form when three edges connect with each other.

To construct the VR complex, balls are surrounding and centered in each points of point cloud P with $\in$ radius. A Point is connected with other points when its ball touches others' balls. If there are two balls intersect with each other, line segment (1-simplex) would be appeared; three balls intersect with each other, triangle (2-simplex) would be appeared; four balls intersect with each other, tetrahedron (3-simplex) would be appeared. Similar to VR complex, to construct Cech Complex, what need more is the point with double intersection when creating line segment, triple intersection when creating the triangle, and quadruple intersection when creating the tetrahedron.

2.2.2 Filtration

In Figure 5, the presence of hole can be visualized with the simplicial complex in $\in$. It shows the simplicial complex with a suitable $\in$. However, when the chosen $\in$ is too small, the connected points are not enough to reveal the hole; when the chosen $\in$ is too large, the hole will be filled up by the dense connection.

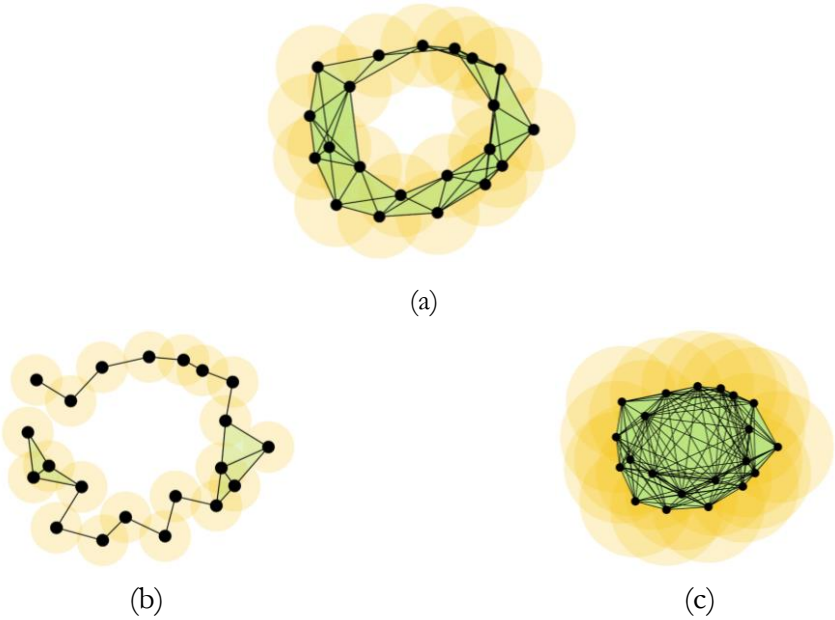


Figure 5: (a) Simplicial Complex with a suitable $\in$;
(b) Simplicial Complex with too small $\in$; (b) Simplicial Complex with too large $\in$

To handle this problem, filtration, another technique in PH involved. Filtration solves the limitations of classical homology by describing the shape in multiple scale (Fugacci et al., 2016). It is the process of revealing the topological features appeared in simplicial complex with each level of $\in$. In Figure 6 , it shows that filtration is achieved starting from the point cloud data, and the balls surrounding the points increase their radius $\in$ to form different levels of simplicial complex. In different levels of simplicial complex, the change of topological features can be detected. To conclude, the problem of choosing s suitable $\in$ is overcome by considering all the $\in$, from 0 to value that all topological features are filled up.

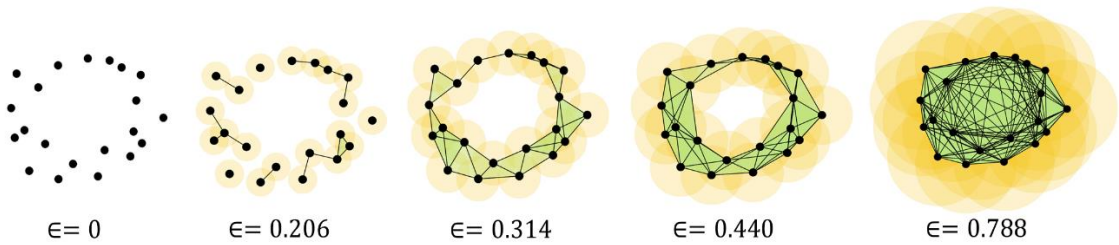


Figure 6: Filtration

2.2.3 Persistence Diagram

For the change of a topological feature, that is its appearance and disappearance, can be recorded as birth and death (persistence). The collection of all topological features' persistence can be visualized in the persistence diagram.

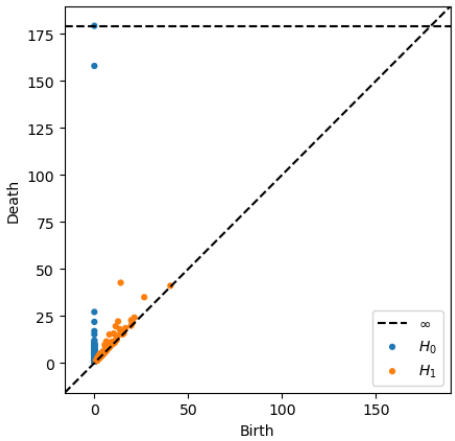


Figure 7: Persistence Diagram

In Figure 7, a persistence diagram is demonstrated. Its x-axis represents the birth time, that is the value of $\in$ when the topological features appear. Its y-axis represents the death time, that is the value of $\in$ when the topological features disappear. Two kind of topological features are shown here, including components (H0 in blue dot) and holes (H1 in orange dot). The diagonal dashed line represents the start of a topological feature, so the distance between each dots to the dashed diagonal line represents their lifetime (Time of Death – Time of Birth). The larger the distance, the longest the lifetime the topological feature has. The horizontal dashed line represents the value of $\in$ that all topological features are filled up, so no dots plotted after this line.

From the persistence diagram, all topological features can be revealed. Regardless the size, shape and location of the holes in set of point cloud, all can be identified by Persistent Homology with the construction of simplicial complex and the process of filtration.

2.3 Discussions

As reviewed in Ch. 2.1 Hole Detection, so far methods related to hole detection have limitations, including the require of threshold value, the require of user intervention, restriction in input data and lack of interpretability, while Persistent Homology in hole detection can overcome these obstacles. When detecting hole in point clouds by Persistent Homology, the following advantages can be obtained:

1. When connecting the points, persistent Homology require neither the input of threshold nor user intervention. This is because filtration consider all $\in$ to construct the simplicial complex and all holes in the point clouds can be revealed, even the hole is tiny or have low visibility to naked eye. Moreover, compared with triangulated mesh, such as Delaunay Triangulation, simplicial complex can create larger number of simplices. Therefore, the accuracy of identify all holes using Persistent Homology is higher than the current methods.

2. The data type available in Persistent Homology has fewer restriction. The data types able in Persistent Homology include point clouds and even digital images. For point cloud data, only positional information (XYZ) is needed, no GPS information is needed. The flexibility of detecting hole by Persistent Homology is much higher.

3.Persistent Homology is interpretable. Persistent Homology gives a theoretical framework towards hole detection as it is based on the algebraic topology. The formalized method enables the explicit definition of shape.

4. Persistent Homology is stable under perturbation of data. That means when there is adjustment on the position of points in the point cloud, the result executed by Persistent Homology will not have big change. It is essential in detecting hole in point clouds since the small error occurs during the LiDAR data collection inevitably, but Persistent Homology accepts tolerate these errors.

Persistent Homology provides a new perspective in hole detection with the ability to compensate the current limitations of hole detection. In this regard, this thesis research attempts to fill the current gap by proposing an approach to detect the hole in point clouds by Persistent Homology and studying the classification of hole in point clouds by Maximum Likelihood Estimation.

Chapter 3

3. Hole Detection based on Persistent Homology

This chapter presents the approach for identifying the hole in point clouds by Persistent Homology. The topological features in point clouds are tracked by Risper.Py and the significant holes are extracted by Persistent Entropy. The proposed approach is applied to the real LiDAR datasets, including terrestrial LiDAR point clouds and airborne LiDAR point clouds, for experimental testing. The whole approach is achieved by Python. The assessment of accuracy on the result would also be conducted. Finally, this chapter ends with a short summary of the proposed approach and result.

3.1 Methodology

3.1.1 Overall workflow

Figure 8 illustrates the overall workflow for hole detection in point clouds. Since most of the point clouds are large in data size and leads to the low computational speed, sampling is conducted by greedy permutation. The effect of greedy permutation is thus tested first (refer to Section 3.1.2.2). After confirming the effect of greedy permutation, detection of topological features can be conducted (refer to Section 3.1.2). Point cloud is input to Ripser.Py and the output of Ripser.Py is the persistence diagram with both components and holes detected (refer to Section 3.1.2.1), so extraction of significant holes has to be conducted (Refer to Section 3.1.3). First, only holes are extracted from the result of Ripser.Py(Refer to Section 3.1.3.1). Then, the significant and representative holes is further extracted by Persistent Entropy (refer to Section 3.1.3.2). The final output is the positional information of the significant and representative holes in the point clouds (refer to Section 3.1.3.3).

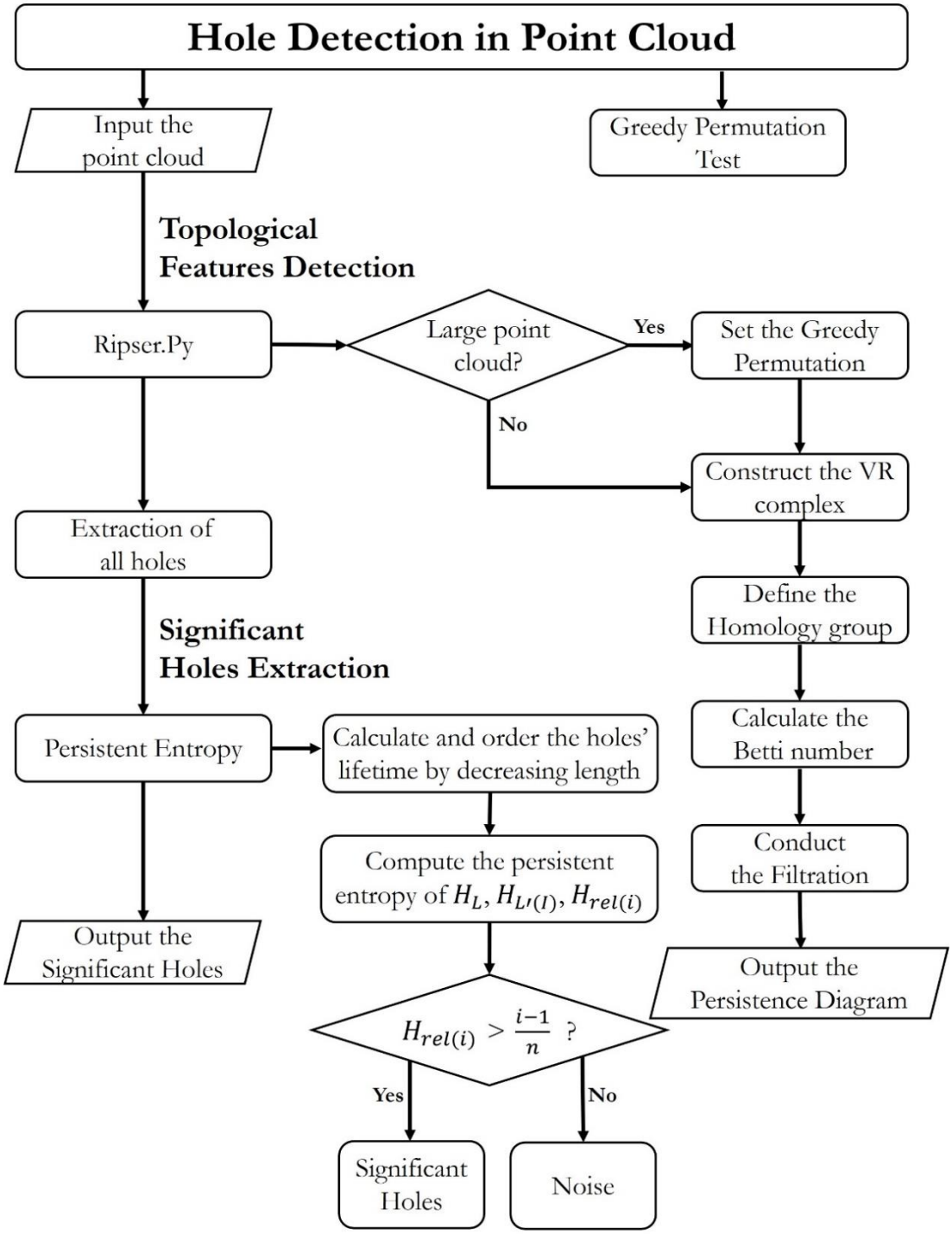


Figure 8: Overall workflow of hole detection (Chapter 3)

3.1.2 Detection of Topological Features

3.1.2.1 Persistent Homology

To apply Persistent Homology, Ripser.Py, a package for python that build on the blazing fast C++ Ripser as the core computational engine, is used to detect the topological features in point clouds. The performance of Ripser was assessed in Otter (2017) and it has the best performance among various open-source PH libraries. For Ripser.Py, more advancement has been made by making some incredibly useful features in PH easy to use. The most important one used in this thesis is Sparse Filtrations. It speeds up the computation of filtration by applying the greedy permutation to subsample the point clouds and calculating the sparse distance matrix to reduce the unuse data.

The general flow of apply Persistent Homology is as follow:

Step 1. Input the point cloud P.

Step 2. Sampling the point cloud when the point cloud is too large to handle by greedy permutation (n_perm).

Step 3. Construct the Vietoris-Rips Complex $\sum_P$ by building the neighborhood graph. The neighborhood graph contains the nodes (points) and edges (the two connected points). The simplicial complex is then built by defining the intersection.

Step 4. Define the homology group of $\sum_P$ to reveal the presence of topological features as:

$$H_k(\sum_P) \coloneqq H_k\big(C_*(\sum_P)\big) = \frac{Z_k(\sum_P)}{B_k(\sum_P)} \tag{1}$$

where, the chain complex associated with $\sum_P$ is denoted as:

$$C_*(\sum_P) \coloneqq (C_k(\sum_P), \partial_k)_{k \in Z} \tag{2}$$

and $C_k(\sum_P$ is the free abelian group generated by the k-simplices of $\sum_P$

The boundary map encodes the boundary relations between the k-simplices and the (k-1) simplices of $\sum_P$ such that $\partial^2 = 0$ is:

$$\partial_k = C_k(\sum_P) \to C_{k-1}(\sum_P) \tag{3}$$

The group of k-cycles of $\sum_P$ is:

$$Z_k(\sum_P) \coloneqq \ker(\partial_k) \tag{4}$$

The group of k-boundaries of $\sum_P$ is:

$$B_k(\sum_P) := Im(\partial_{k+1}) \tag{5}$$

Step 5. Calculate the Betti number to reveal the topological features in the homology group. β_0 is the number of connected components, β_1 is the number of holes and β_2 is the number of voids.

Where the kth Betti number is calculated as:

$$\beta_k = \dim(H_k(\sum_P)) \tag{6}$$

Step 6. Conduct the filtration, which is defined as:

$$H_k^{p,q}(\sum^f) := Im(i_k^{p,q}) \tag{7}$$

Step 7. Produce the persistence diagram by plotting the components and holes with their death and birth.

After applying Persistent Homology to the input point cloud, the result of Persistent Homology is stored with a list of cycles in each dimension, the birth and death of the cycles. The default value 1 is used here so the extracted topological features include the components (0-simplex) and holes(1-simplex). Then, all holes have to be extracted before identifying the significant holes. The list of cycles includes the components and holes. In the list of holes, each hole is represented by its edge, and the edge is represented by the point number of the point that compose of the edge. The persistent diagram is also output.

The Python script for achieving them is as follow:

Python Script 1: Detection for topological features

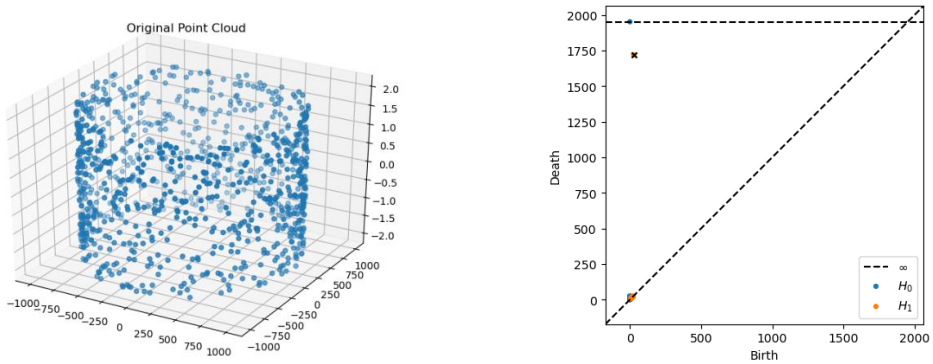
```
input_path=r"D:\door1.txt"
point_cloud= np.loadtxt(input_path)
result = ripser(point_cloud,coeff=17, do_cocycles=True, n_perm=10000)
plot_diagrams(diagrams, show = True)
cocycles = result['cocycles']
hole=cocycles[1]
```

3.1.2.2 Greedy Permutation Test

Greedy permutation is used for sampling the point cloud for faster computation. Moreover, for some extreme large point clouds, it is unable to run Ripser.Py, so sampling of point cloud is needed. In this regard, test on greedy permutation is performed before performing Ripser.Py. Greedy permutation test is carried out by setting the sampling value to the point cloud and comparing the detection results between the sampled point cloud and original point cloud.

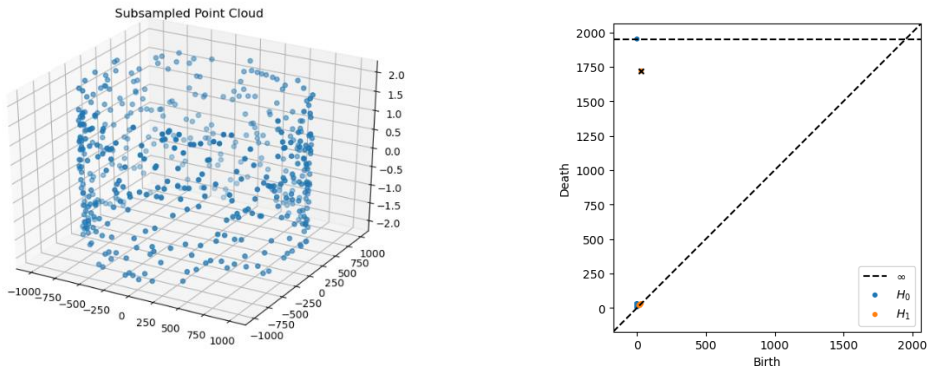
In the original point cloud, there are totally 1000 points contained with only one single big hole. The point cloud is plotted and the detection result is showed in persistence diagram in Figure 9. The significant hole detected, which is the big hole of the point cloud, is marked with 'x' in the persistence diagram. For sampled point cloud, it is sampled to 500 points by the greedy permutation function of Ripser.Py. The sampled point cloud is plotted and the detection result is showed in the persistence diagram in Figure 10. Form the plotting of point cloud, it can be found that the big hole of the point cloud still existed. From the persistence diagram showing the detection result of the sampled point cloud, the significant hole representing the big hole is detected.

To conclude, the sampling do not affect much of the result since the significant hole can be detected in both original point cloud and the sampled point cloud. Greedy permutation can be performed to sample the point cloud and lower the computational cost without affecting the detection result.



(a) (b)

Figure 9: (a) Plot of original point cloud (1000 points); (b) Persistence diagram of original point cloud



(a) (b)

Figure 10: (a) Plot of sampled point cloud (500 points); (b) Persistence diagram of sampled point cloud

3.1.3 Extraction of Significant Holes

3.1.3.1 Persistent Entropy

The holes extracted from Section 3.1.3.1 can be topological noise or significant holes (topological features). Generally, holes with long lifetime are considers as topological features because of their stability to small changes in the filtration. Yet, the hole with the longest lifetime may not be the most representative topological feature. Therefore, Persistent Entropy can be used for separating the topological noise from features and identifying the significant holes by differentiating long from short lifetime.

The general flow of Persistent Entropy is as follow (Atienza et al., 2016):

Step 1. Extract the birth and death of each holes from the detection result and calculate the lifetime by:

$$Lifetime = Time\ \ of\ \ Death - Time\ \ of\ \ Birth \tag{8}$$

Step 2. Order the interval (lifetime l) of all holes by decreasing order.

Step 3. Compute the persistent entropy H_L of the filtration:

$$H_L = -\sum_{i=l}^{n} \frac{l_i}{S_L} log \frac{l_i}{S_L} \tag{9}$$

where,

$$L = \{l_i = d_i - b_i : 1 \leq i \leq n\} \tag{10}$$

$$S_L = \sum_{i \in I} l_i \tag{11}$$

Step 4. Compute $H_{L'(i)}$:

$$H_{L'(i)} = -\sum_{i=l}^{n} \frac{l'_k}{S_L} log \frac{l'_k}{S_L} \tag{12}$$

where,

$$l'_k = \frac{S_i}{e^{H_i}} \tag{13}$$

Step 5. Compute $H_{rel(i)}$:

$$H_{rel(i)} = (H_{L'(i)} - H_{L'(i-1)})/(\log(n) - H_L) \tag{14}$$

Step 6. If $H_{rel(i)} > \frac{i-1}{n}$, the interval represents the topological feature, otherwise, it represents the topological noise. The result identifies whether each hole belongs to topological features or topological noise.

The Python script for achieving Persistent Entropy is as follow:

Python Script 2: Identify the significant holes by Persistent Entropy

```
#Compute the lifetime of each hole
dgm=[]
diagrams = result['dgms']
dgms=diagrams[1]
dgm.append(dgms)
for dgm in dgm:
        l = dgm[:, 1] - dgm[:, 0]
#2. Order the intervals by decreasing length
index=sorted(range(len(l)), key=lambda k: l[k])
index=index[::-1]
l=np.sort(l)
l=l[::-1]
```

```
#3. Compute the persistent entropy HL
if all(l > 0):
    L = np.sum(l)
    p = l / L
    Hmax = -np.sum(p * np.log(p))#HL
#4. Compute the persistent entropy HL'(i)
#4.1 Compute l'k
#4.1.1 Compute e^(Hi)
Hi=[]
for a in range(len(l)):
    b=0
    temp_Hi=0
    while b<=a:
        temp_Hi=temp_Hi+((-l[b]/L)* np.log(l[b]/L))
        b=b+1
    Hi.append(temp_Hi)
EHi=[]
for item in Hi:
    EHi.append(math.exp(item))
#4.1.2 Compute Si
lk=[]
for a in range(len(l)):
    Sum=0
    b=0
    while b<=a:
        Sum=Sum+l[b]
        b=b+1
    lk.append(Sum/EHi[a])#l'k
    H2i=-(lk/L)* np.log(lk/L) #H'i
#Compute Hrel(i)
Hrel=[] #Hrel(i)
feature=[]
for a in range(len(l)):
    Hrel.append((H2i[a]-H2i[a-1])/(np.log(len(l))-Hmax))
    compare=((a-1)/len(l))
    if Hrel[a]>compare:
        feature.append(1)
    else :
```

```
            feature.append(0)
feature=np.array(feature)
```

3.1.3.2 Output of Significant Holes

After the identification of significant holes by Persistent Entropy in the previous section, the coordinates of the points that composed of the holes have to be output. Because the detection result only lists the point number over the point cloud input, the coordinates of the specific points have to search in the point cloud data file. The Python script to achieve it is as follow:

Python Script 3: Output the coordinates of significant holes

```
x_hole=[]
y_hole=[]
z_hole=[]
index=np.array(index)
Find_hole=np.empty((len(feature), 2))
vertix_all_hole=[]
for a in range(len(feature)):
        Find_hole[a,0]=feature[a]
        Find_hole[a,1]=index[a]
hole_feature=[]
for a in range(len(Find_hole)):
      x_hole_feature=[]
      y_hole_feature=[]
      z_hole_feature=[]
      if Find_hole[a,0]==1:
            position=int(Find_hole[a,1])
            hole_feature.append(position)
            temp=hole[position]
            vertix_hole=[]
            for b in temp:
                  for c in b:
                     if all(vertix_hole!=c):
                           vertix_hole.append(c)
                     if all(vertix_all_hole!=c):
                           vertix_all_hole.append(c)
```

```
# vertix_all_hole=Union(vertix_all_hole,vertix_hole)
for d in range(len(vertix_hole)):
    vertix_index=vertix_hole[d]
    x_hole_feature.append(point_cloud[vertix_index,0])
    y_hole_feature.append(point_cloud[vertix_index,1])
    z_hole_feature.append(point_cloud[vertix_index,2])
```

3.1.3.3 The Efficiency of the Flow

In the current workflow, persistent homology is applied. The main advantage of its application is the increase of efficiency in hole identification. Before the adoption of persistent entropy, the significant holes are identified by naked eyes, that is to output all holes (coordinate of points that compose of each hole) detected from Persistent Homology, including both the topological noise and topological features, and identify whether the hole is significant and representative or not by the visualization.

To reveal the computational performance of two flow, workflow that output all hole and workflow with Persistent Entropy, a comparison of the executed time between them are conducted. The comparison is done using the same data set with different sampling value.

The result is showed in Figure 11. It shows that the computation time of the former workflow required is double of the latter workflow required. Moreover, the maximum number of points that the former one can be handled is 1400, compared with 14000 points that the latter one can run, it is significantly fewer by 10000. (The number of points can be handled depends on the data set input. The larger the total amount of homology in the point cloud, the smaller the number of points can be run)

To conclude, with the use of persistent homology, the thesis successfully enhanced the efficiency of hole identification and the ability of handling large point clouds by lowering the computational cost.

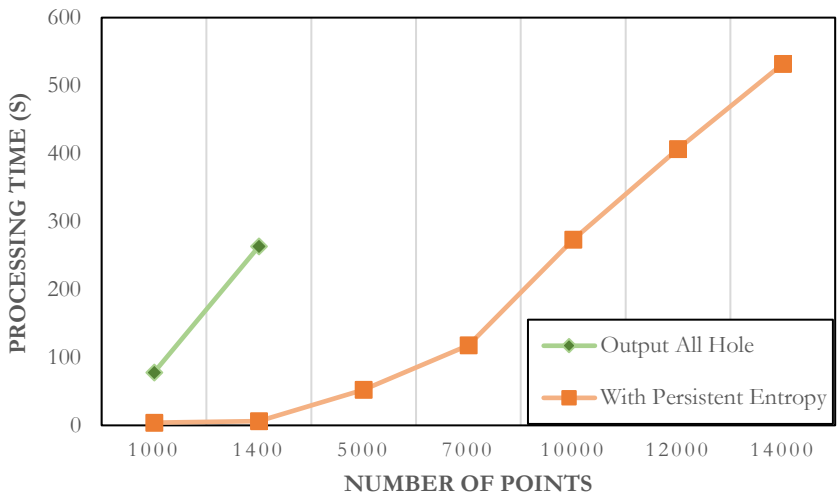


Figure 11: Computational performance of two workflows

3.2 Experimental Testing

3.2.1 Testing Dataset

The study covers both Airborne Laser Scanning and Terrestrial Laser Scanning. These two types of laser scanning are studied respectively because of the apparent differences between their LiDAR data collected. ALS utilizes the airborne as the platform, so the data collected mostly enable the top view of the object scanned only, like the rooftop of houses. While TLS is ground-based remote sensing that features are scanned from the ground. The object can be visualized with more detail from the data collected, but the top view or higher part of the objects are usually not available. For MLS, the biggest difference between it and TLS is the mobility, but the characteristics of their LiDAR data are similar. Therefore, the study focuses on Airborne Laser Scanning and Terrestrial Laser Scanning.

To assess the hole detection, various point clouds that contain hole, which is regular, irregular or hole caused by occlusion are tested. The airborne datasets used including Vaihingen_DMC (DGPF Projekt, 2008), 2012 Aluto LiDAR data (Hutchison et al., 2014), EU12_12_Santorini (Pyle, 2012) and 2013 Vancouver LiDAR data (Yan, 2013). The terrestrial point cloud data are collected by field work. Three sets of point cloud data are used. Their survey location is W core, M core and X core of the Hong Kong Polytechnic University respectively. The laser scanner uses for the former two is Leica RTC 360 with 1550nm laser wavelength. The laser scanner uses for the latest one is Leica BLK360 with 830nm laser wavelength that no return would be received from the water body.

3.2.2 Design of Experiment and Evaluation

Following the method in Section 3.1, Persistent Homology is applied into the point cloud. The separation of topological noise and features are done by Persistent Entropy and its result is displayed ArcScene 10.4.1 or ArcMap 10.4.1 (for result with the overlapping of satellite image). The persistent diagram is then output. To evaluate the performance of hole detection result, accuracy assessment is conducted by compared the area of hole detected and area of hole in the point cloud. Both of the area are obtained by OpenRoads Designer CONNECT Edition - 2019 Release 2. The entire procedure of the experiment is as follow:

1. Select the study area and reduce the size of point cloud.
2. Determine the maximum value of greedy permutation for sampling the point cloud if necessary.
3. Apply Persistent Homology and Persistent Entropy to extract the significant hole.
4. Output the persistent diagram with the indication of significant hole.
5. Input the original point cloud and the detection result into the OpenrRoads Designer and measure the area of hole by 'Measure Area' tool.
6. Compare the computed value and measured value for quality control of the detection result.

For the quality control part, the accuracy of hole detection result can be analyzed by comparing the visualization of hole detected and the original point cloud, and also the comparison between area of hole detected and area of hole in the point cloud. The comparison reveals the similarity of the size of the detected hole and the hole in potin cloud. The smaller the difference, the higher the accuracy of the detection result. The true area, that is the area of hole in the point cloud, is obtained by measuring the point cloud but not measuring the object in reality. This is because the area of hole detected contain the error occurred during the process of data collection. If the true area is obtained by measuring the object in reality, these errors are not contained, so leads to bigger difference between the comparison of two area but the large difference does not mean the low accuracy in this case. The error occurred during the process of data collection has no relationship with the performance of hole detection result. Therefore, the comparison between area of hole detected and area of hole in the point cloud is conducted to reveal the performance of hole detection in terms of size of hole.

The comparison of the visualization of the hole detected and the original point cloud is done by overlapping them. Because it is hard to visualize if the complicated simplicial complex is drawn, only points composed of the detected hole are output and displayed in

the Section 3.3 Results and Analysis. Generally, points composed of the detected hole is the boundary of the detected hole and points are connected to their points nearby to form the simplicial complex. From the overlapping of the detected hole and original point cloud, if the detected points constitute the boundary of the hole on the point cloud, the first and second objective of this research is achieved that the proposed method successfully detected and located the topological hole.

3.3 Results and Analysis

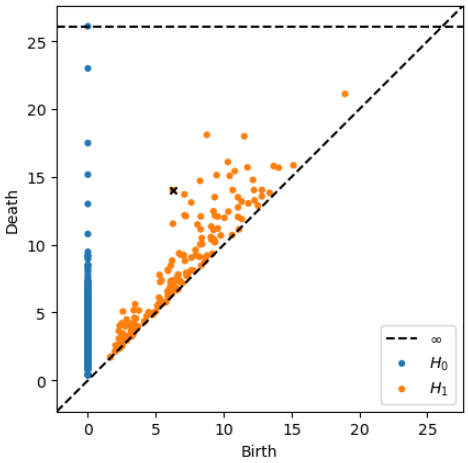
In this part, the hole detection results of Persistent Homology applying in point cloud datasets from airborne and terrestrial laser scanning are displayed. Various type of holes is tested, which can be classified in regular shape, irregular shape and hole caused by occlusion. In Section 3.3.1 Airborne Laser Scanning, hole in regular shape including swimming pool, road, rooftop, water tank and dam, hole in irregular shape including occlusion caused by smoke and cloud, occlusion from building and occlusion from tree, are displayed and analysis. In Section 3.3.2 Terrestrial Laser Scanning, hole in regular shape including window and door, hole in irregular shape including puddle, and hole belongs to occlusion including occlusion caused by words on wall and occlusion caused by roadblock, are displayed and analysis. For the occlusion, they are also hole in irregular shape. In Section 3.3.3 Analysis of Hole Detection Result, analysis is conducted based on the detection result.

3.3.1 Airborne Laser Scanning

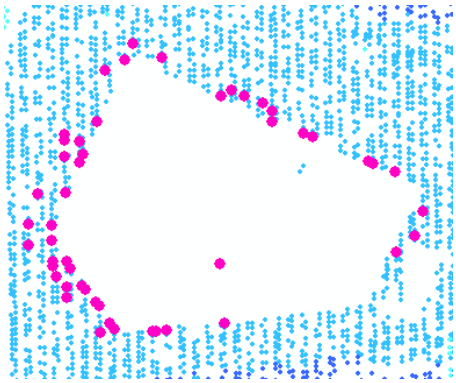
3.3.1.1 Hole of Regular Shape Object

1. Swimming Pool

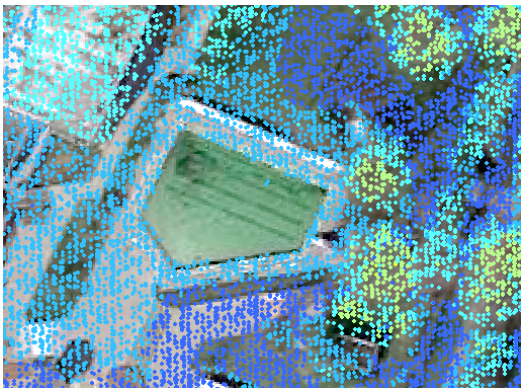
In point cloud Vaihingen_DMC, the swimming pool in the shape of hexagon is empty due to the absorption of signal by pool water. The use of Persistent Homology successfully detects the hole of swimming pool in the point cloud. The significant hole, which indicates as 'x' in the persistent diagram, is extracted and visualized in Figure 12, the point surrounding the empty hexagon is the hole boundary.



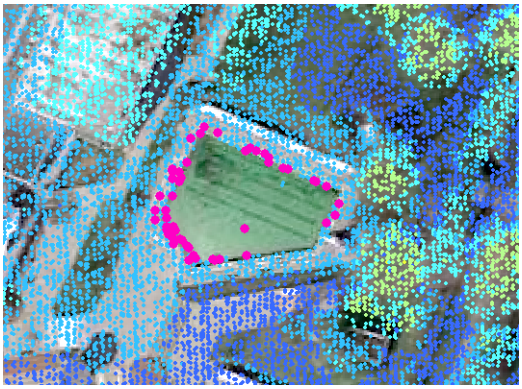
(a) Persistent Diagram with 'x' in significant hole



(b) Points (pink) that composed of the significant hole



(c) Overlapping of satellite image with the original point cloud

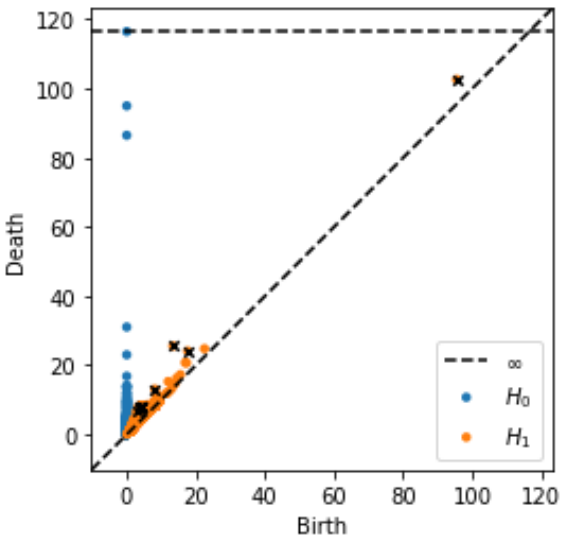


(d) Overlapping of satellite image with the original point cloud and hole detection result

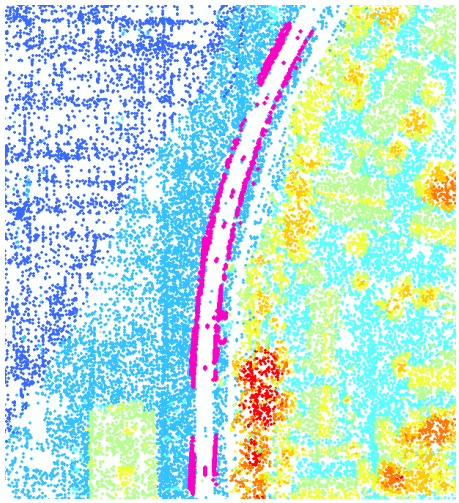
Figure 12: Hole Detection result of Swimming Pool

2. Road

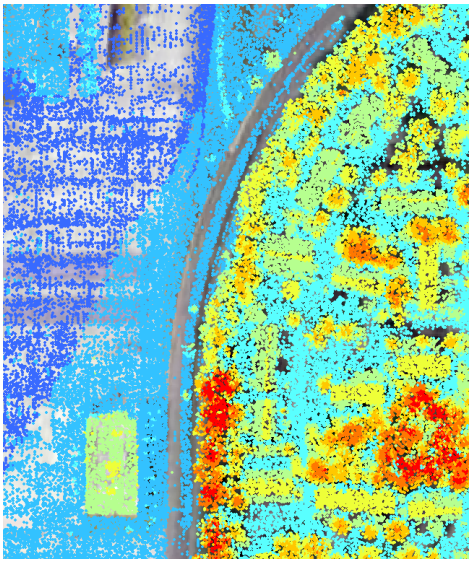
In 2013 Vancouver LiDAR data, the road alignment in spiral curve caused the missing area in the point cloud. This is because the road is made of asphalt, which is black in color and most of the return signal is lost. After applying the Persistent Homology, the significant holes are detected and formed the hole boundary, which is points in pink showed in Figure 13.



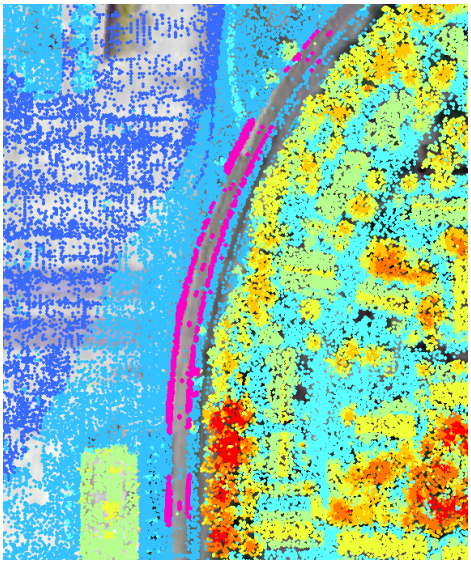
(a) Persistent Diagram with 'x' in significant hole



(b) Points (pink) that composed of the significant hole



(c) Overlapping of satellite image with the original point cloud

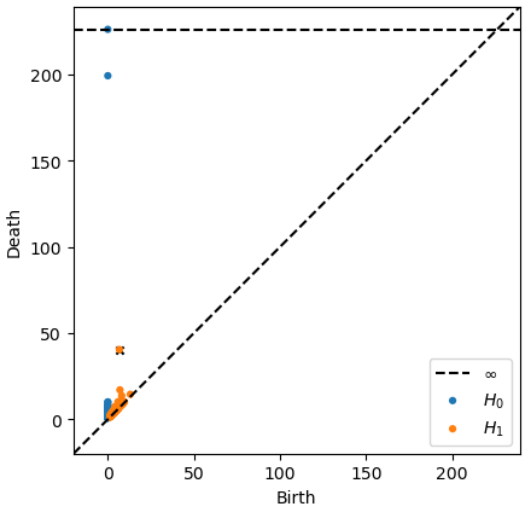


(d) Overlapping of satellite image with the original point cloud and hole detection result

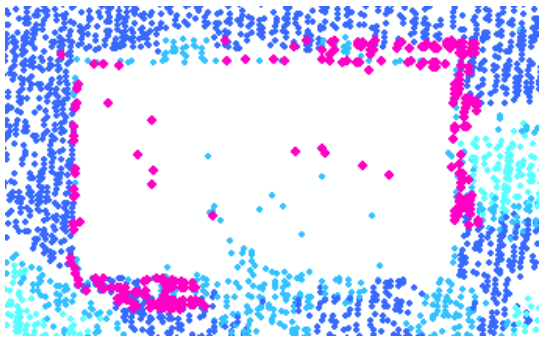
Figure 13: Hole Detection result of Road

3. Rooftop

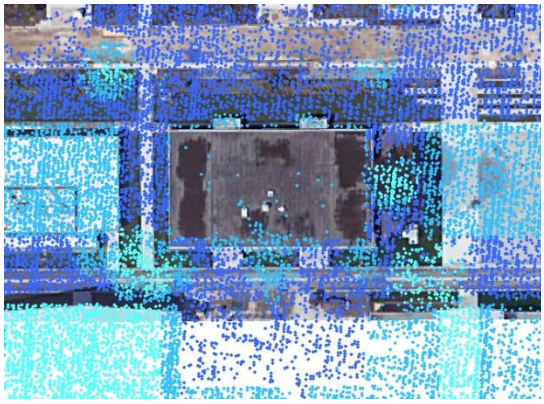
In 2013 Vancouver LiDAR data, the rooftop in rectangular shape and black color results in a hole in the point cloud. To detect the hole, Persistent Homology is applied into the point cloud. The result is presented in Figure 14. The persistence diagram shows the significant hole detected. The points composed of the hole detected locate on the boundary of hole in the point cloud.



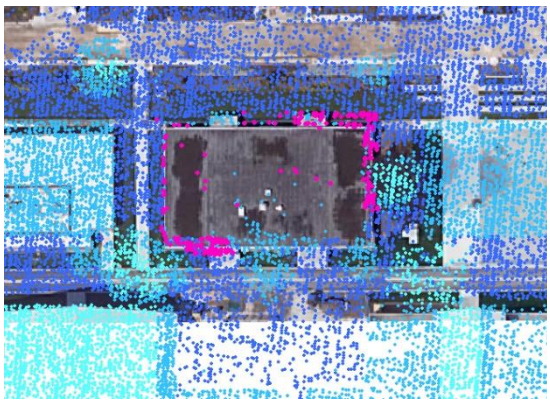
(a) Persistent Diagram with 'x' in significant hole



(b) Points (pink) that composed of the significant hole



(c) Overlapping of satellite image with the original point cloud

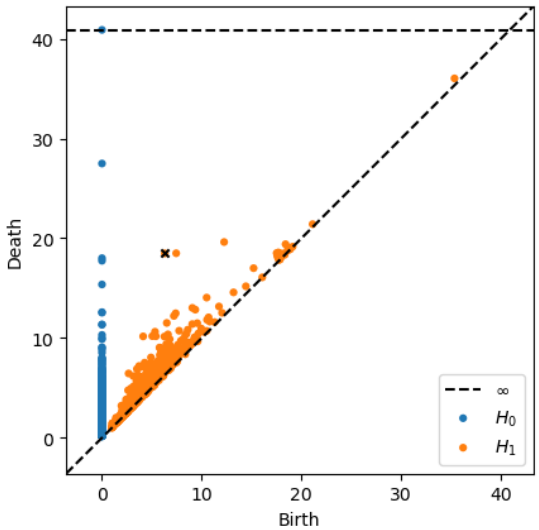


(d) Overlapping of satellite image with the original point cloud and hole detection result

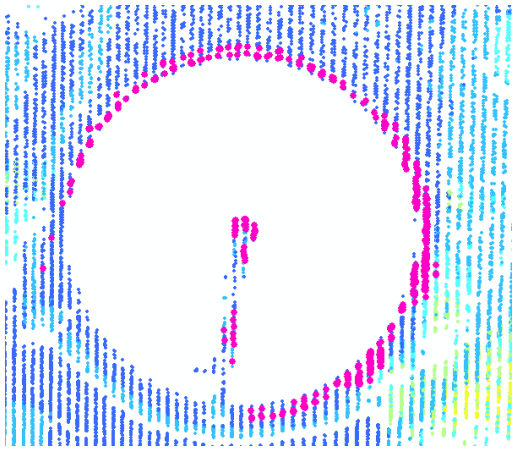
Figure 14: Hole Detection result of Rooftop

4. Water tank

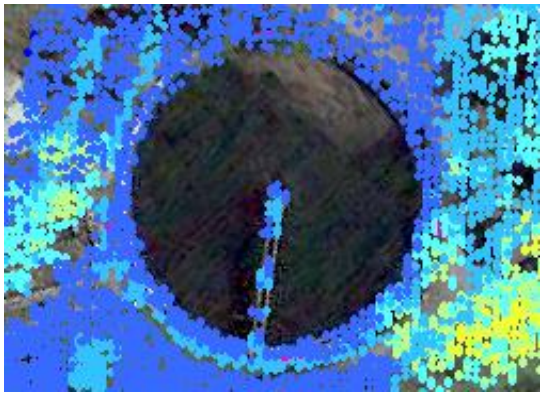
In Vaihingen_DMC point cloud data, the water tank in shape of circle and fills with water leads to an empty area without point scanned, as water surface may absorb or refract the signal projected by the laser scanner. Processing the point cloud with Persistent Homology helps detecting the empty area and the detection result is showed in Figure 15. The topological hole detected is the points surrounding the hole in circle in the point cloud.



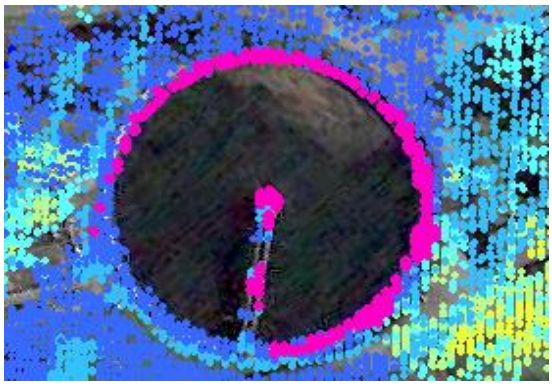
(a) Persistent Diagram with 'x' in significant hole



(b) Points (pink) that composed of the significant hole



(c) Overlapping of satellite image with the original point cloud

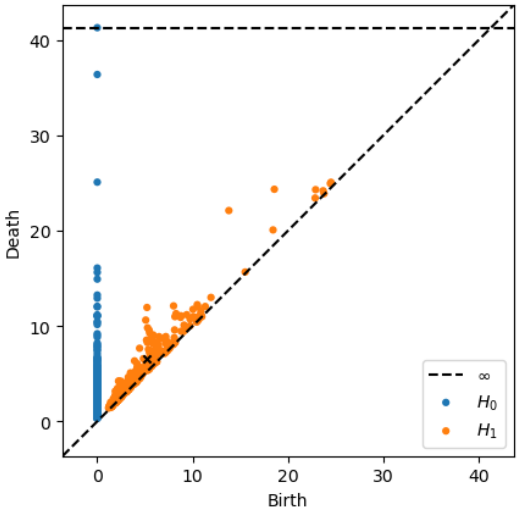


(d) Overlapping of satellite image with the original point cloud and hole detection result

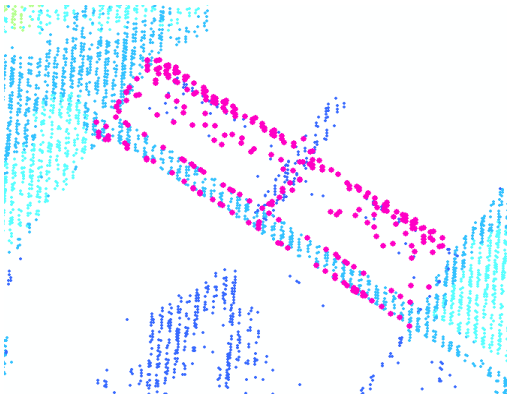
Figure 15: Hole Detection result of Water Tank

5. Dam

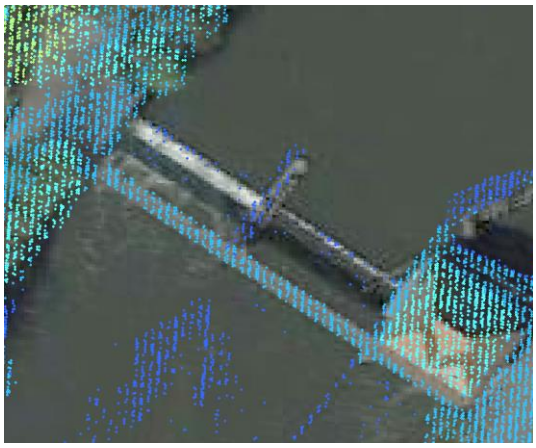
In the point cloud of Vaihingen_DMC, the river caused the large hole in the point cloud. Due to the high computational cost of running a point cloud in huge size, only part of the river is detected and applied into the algorithm of Persistent Homology. The dam constructed across the river appears with the hole in the point cloud because part of it is covered with water. Persistent Homology successfully detect the hole caused by the water and the points composed of the hole detected is the boundary of hole in the point cloud, which is showed in Figure 16.



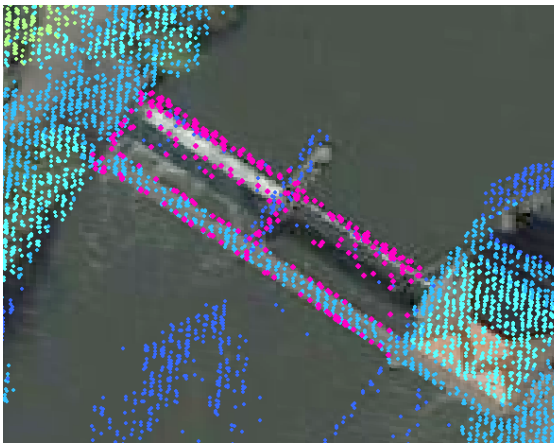
(a) Persistent Diagram with 'x' in significant hole



(b) Points (pink) that composed of the significant hole



(c) Overlapping of satellite image with the original point cloud



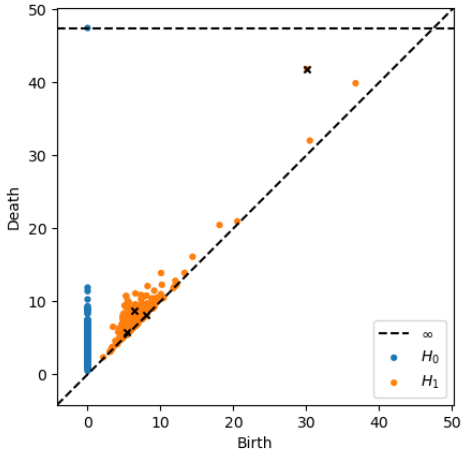
(d) Overlapping of satellite image with the original point cloud and hole detection result

Figure 16: Hole Detection result of Dam

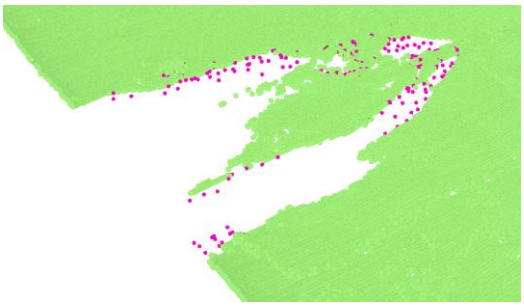
3.3.1.2 Hole of Irregular Shape Object (Occlusion)

1. Occlusion caused by Smoke

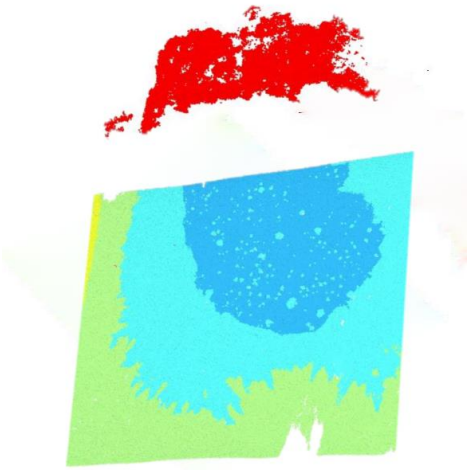
In 2012 Aluto LiDAR data, there is thick smoke when conducting the airborne survey. From the LiDAR data (Figure 17(c)), the smoke can be visualized in red color on the sky. The smoke also occluded part of the light emitted from the laser scanner so hole occurred in the point cloud. The hole is detected by Persistent Homology successfully. The results in Figure 17 shows that the boundary of hole in the point cloud is detected and located well.



(a) Persistent Diagram with 'x' in significant hole



(b) Points (pink) that composed of the significant hole



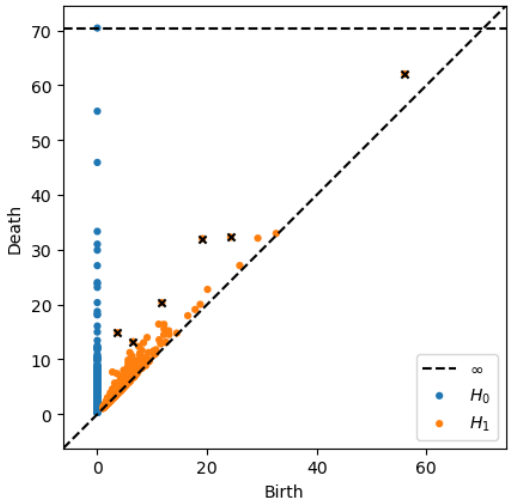
(c) Overlapping of satellite image with the original point cloud

(d) Overlapping of satellite image with the original point cloud and hole detection result

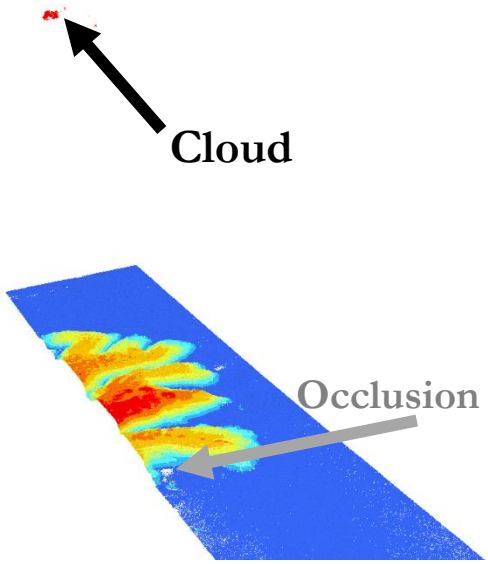
Figure 17: Occlusion detection result of Smoke

2. Occlusion caused by Cloud

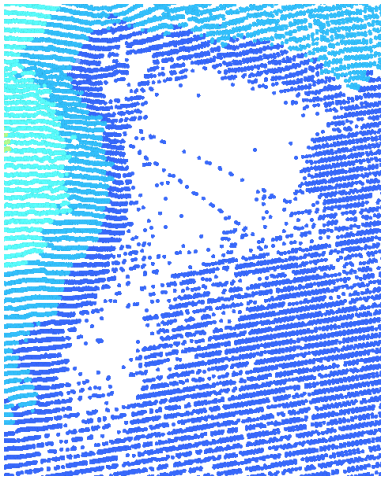
In the point cloud of EU12_12_Santorini, when conducting the airborne laser scanning survey, the flying height of the airborne is higher than some of the thick cloud, so the cloud is scanned and can be found in the point cloud. The cloud occluded some of the signal and the occlusion appeared in the point cloud. After applying Persistent Homology, the significant holes are successfully detected. They are points surrounding the hole in the point cloud. The results are present in Figure 18.



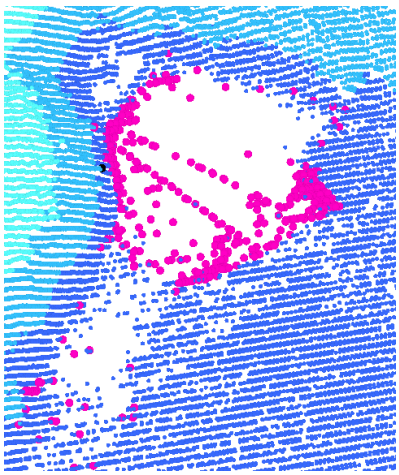
(a) Persistent Diagram with 'x' in significant hole



(b) Original point cloud with cloud (grey arrow) scanned and hole on the ground (black arrow)



(c) Overlapping of satellite image with the original point cloud

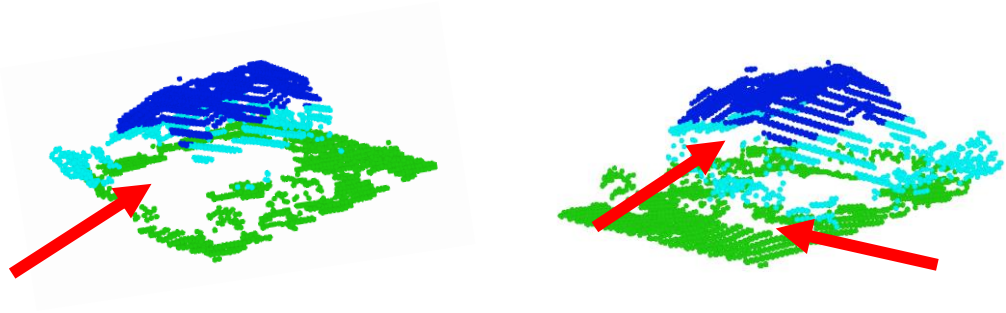


(d) Overlapping of satellite image with the original point cloud and hole detection result

Figure 18: Occlusion detection result of Cloud

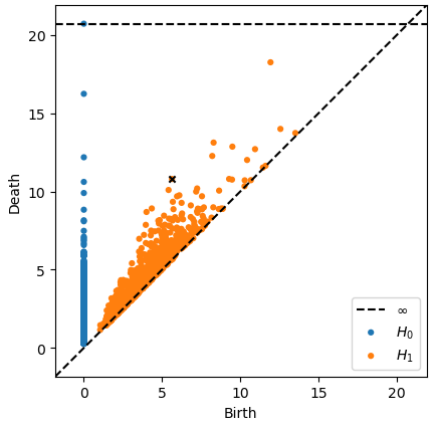
3. Occlusion of building

In 2013 Vancouver LiDAR data, there are many houses contain the occlusions because of the block of signal by other houses in the cluster of buildings. One of the occlusios detected from the building is illustrated as an example here. In Figure 19 , the 3D view of the house with occlusion is showed. The red arrow points out the occlusion. In Figure 20, the detection result from Persistent Homology is showed. The occlusion is detected that the points form the boundary of the occlusion.

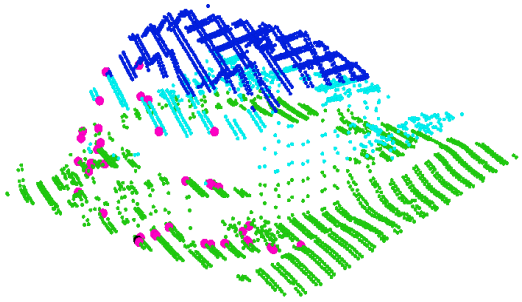


(a) (b)

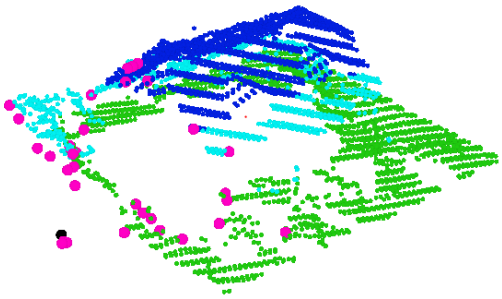
Figure 19: 3D view of building with occlusions (red arrow)



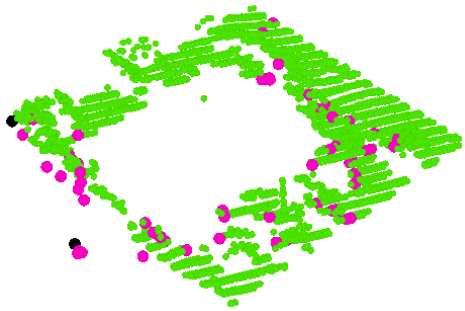
(a) Persistent Diagram with 'x' in significant hole



(b) Occlusion 1-Overlapping of original point cloud and hole detected



(c) Occlusion 2-Overlapping of original point cloud and hole detected

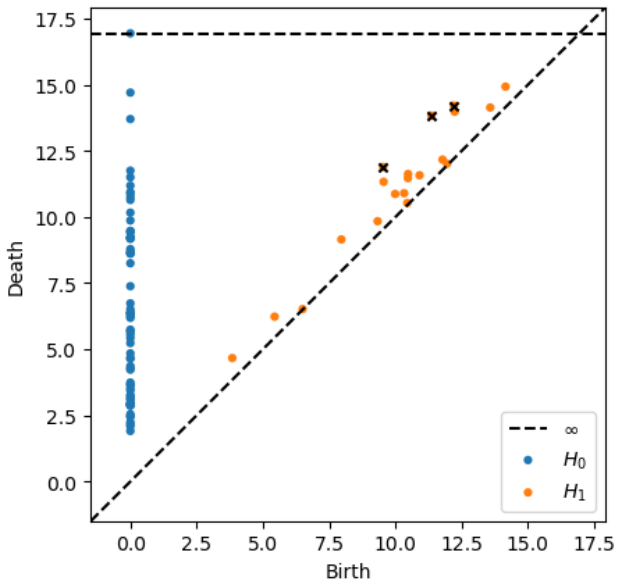


(d) Occlusion 3-Overlapping of original point cloud and hole detected

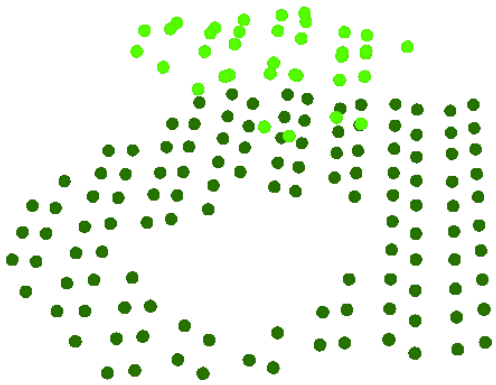
Figure 20 Occlusion detection result of building

2. Occlusion of tree

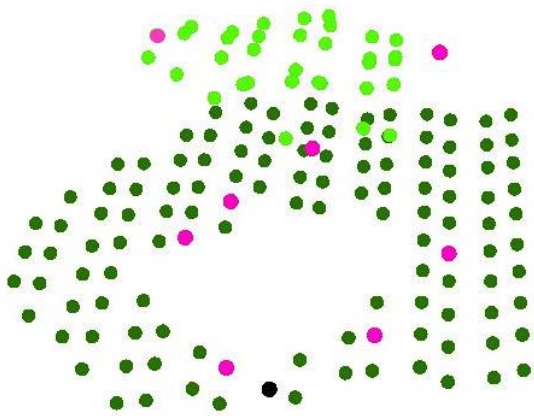
In the point cloud of EU12_12_Santorini, plenty of trees is scanned with occlusion because the signal is occluded by other trees. So the point cloud is input into the algorithm of Persistent Homology and the detection result is showed in Figure 21. The points composed of the significant holes detected successfully locate and identify the occlusion in the point cloud.



(a) Persistent Diagram with 'x' in significant holes



(b) Original point cloud with occlusion on it



(c) Overlapping of original point cloud and detected hole

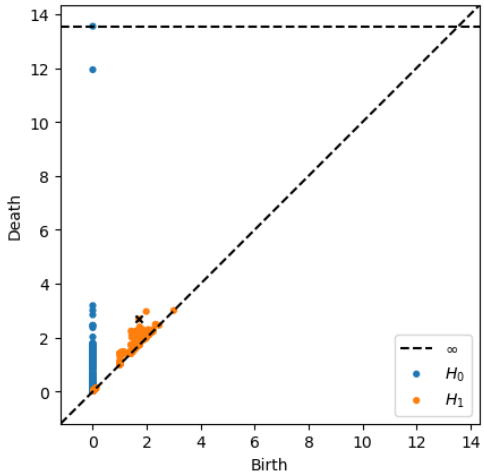
Figure 21: Occlusion detection result of tree

3.3.2 Terrestrial Laser Scanning

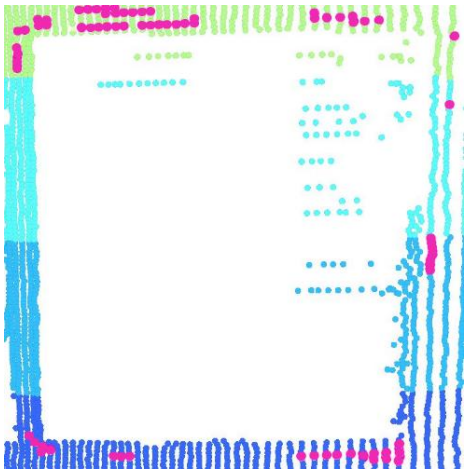
3.2.2.1 Hole of Regular Shape Object

1. Window

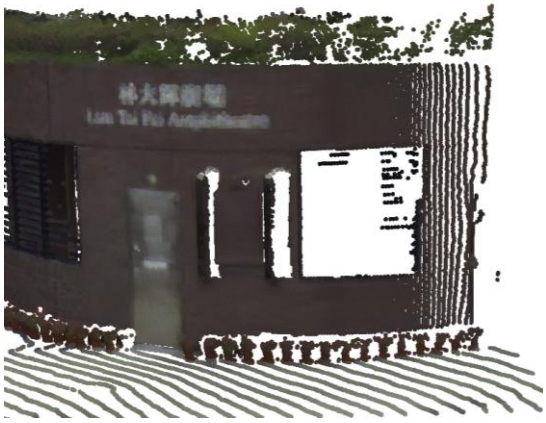
In the terrestrial point cloud data surveyed at W core of the Hong Kong Polytechnic University, hole in rectangular shape is found because of the transparency of the glass window. With the use of Persistent Homology, the significant hole that can represent the window is detected. The result is showed in Figure 22.



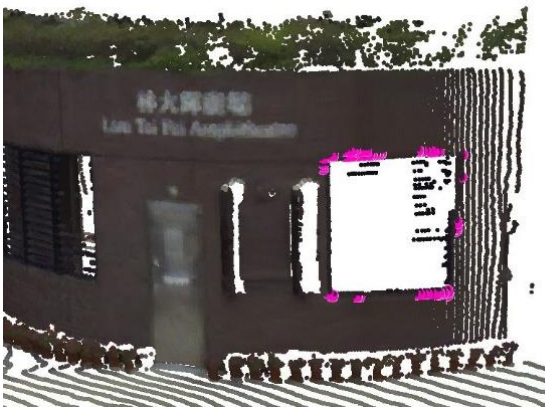
(a) Persistent Diagram with 'x' in significant hole



(b) Overlapping of original point cloud and detected hole



(c) Original point cloud in RGB with hole on it

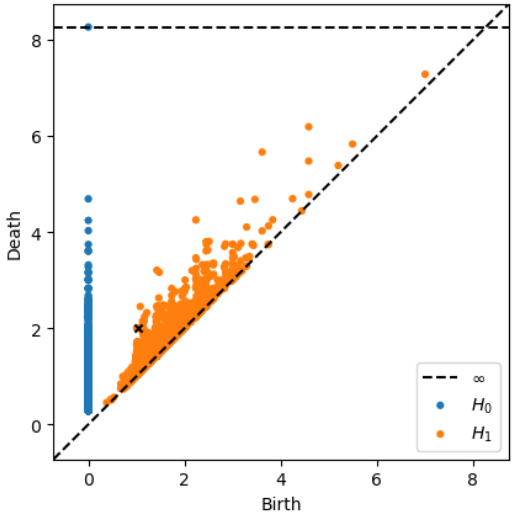


(d) Overlapping of original point cloud in RGB with detected hole

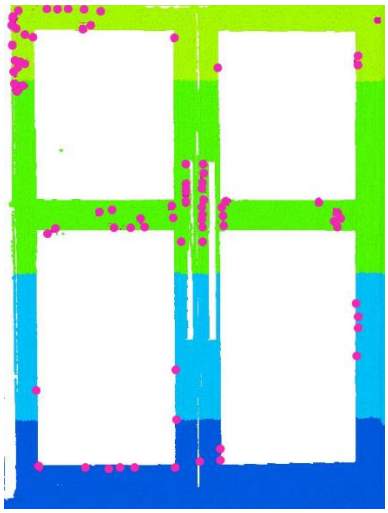
Figure 22: Hole detection result of window

2. Door

The outer wall of M core at the Hong Kong Polytechnic University is formed by glass window and glass door. Hence, the terrestrial point cloud data surveyed there contains many holes. The hole of glass door is illustrated as an example here. After applying the Persistent Homology, the hole caused by glass door is detected. The points composed of the hole detected are located on the metal frame of the door. The result is presented in Figure 23.



(a) Persistent Diagram with 'x' in significant hole



(b) Overlapping of original point cloud and detected hole

(c) Original point cloud in RGB with hole on it

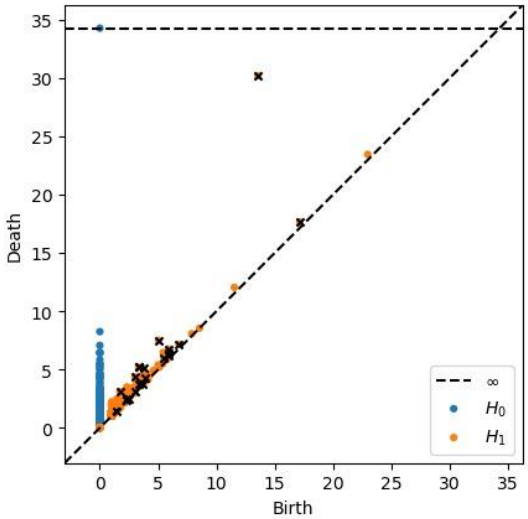
(d) Overlapping of original point cloud in RGB with detected hole

Figure 23: Hole detection result of door

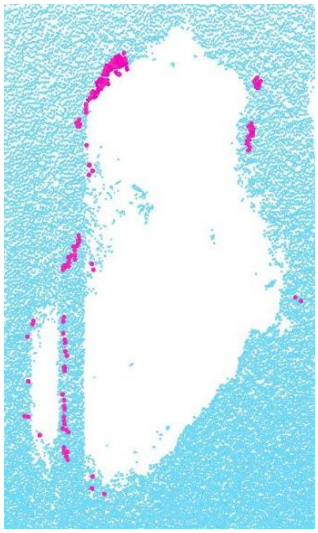
3.2.2.2 Hole of Irregular Shape Object

1. Puddle

The terrestrial laser scanning was conducted after the rain at X core of the Hong Kong Polytechnic University, so there is puddle on the road and it causes the hole in irregular shape on the point cloud. Therefore, the point cloud is processed with Persistent Homology and the hole od puddle is detected. Figure 24 shows the detection result and the boundary of hole is detected and surrounded by points.



(a) Persistent Diagram with 'x' in significant hole



(b) Overlapping of original point cloud and detected hole

(c) Original point cloud in RGB with hole on it

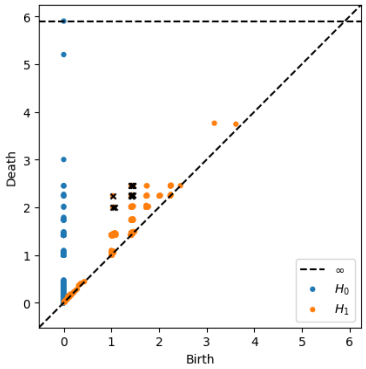
(d) Overlapping of original point cloud in RGB with detected hole

Figure 24: Hole detection result of puddle

3.2.2.3 Occlusion

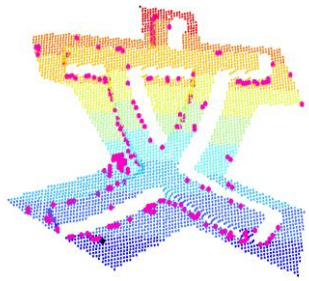
1. Occlusion caused by words on wall

In the terrestrial point cloud data surveyed at W core, there are word sculptures on the wall. Because of the scanning angle, the vertical side of the sculptures is not scanned and occlusion occurred in the point cloud. Therefore, Persistent Homology is used to detect the hole and the result is showed in Figure 25. The significant holes detected constitute the boundary of the occlusion and form the word '文', which represents the successful detection by Persistent Homology.



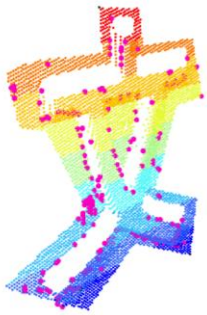
(a)

Persistent Diagram with 'x' in significant holes



(b) View 1- Overlapping of original point cloud and detected occlusion

(c) View 2 - Overlapping of original point cloud and detected occlusion



(d) View 3 - Overlapping of original point cloud and detected occlusion

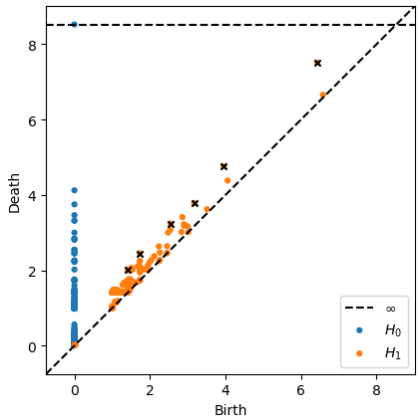
(e) Original point cloud in RGB with occlusion on it

(f) Overlapping of original point cloud in RGB with detected occlusion

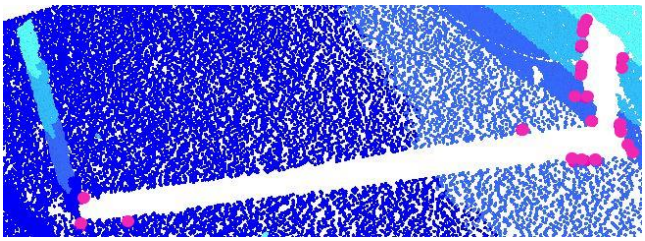
Figure 25: Detection result of occlusion caused by word on wall

2. Occlusion caused by roadblock pillar

In the terrestrial point cloud data surveyed at X core, there are roadblock pillar along the road. The roadblock pillar block the signal of the laser scanner and caused occlusion in the point cloud. With Persistent Homology, the occlusions are detected and one of the occlusion is displayed in Figure 26 as example. The significant holes detected show the location of hole in the point cloud by surrounding it.



(a) Persistent Diagram with 'x' in significant holes



(b) Overlapping of original point cloud and occulsion detected

(c) Original point cloud in RGB with occlusion on it

(d) Overlapping of original point cloud in RGB with detected occlusion

Figure 26: Detection result of occlusion caused by roadblock pillar

3.3.3 Analysis of Hole Detection Result

3.3.3.1 Identification of hole

The performance of detection and extraction of hole is assessed by overlapping the original point cloud and the detected hole. For the hole in regular shape, a single topological hole separated from the topological noise can represent the object. For the hole in irregular shape or complicated shape that have many sides, the union of all topological holes extracted is needed to represent the object. From the overlapping, all of the results show that the points composed of the hole detected overlap with boundary of hole on the point cloud. It shows the hole detection result can identify and locate the hole in the point cloud.

The performance of the detection result is also assessed by the comparison between area of hole detected and area of hole in point cloud. From Table 1, no percentage difference has exceeded 1%. The difference between the detection result and the true hole is small regardless of the types of hole, which means the accuracy of hole detected in terms of area is also high. It also reveals the size of hole detected is appropriate.

Overall, the proposed method successfully identifies the hole and is able to locate the hole with accurate size, regardless of the types of holes.

Table 1: Comparison between area of hole detected and area of hole in point cloud

		Area of Hole Detected (m^2)	**Area of Hole in Original Point Cloud (m^2)**	**Differences (m^2)**	**Percentage Difference (%)**
Airborne Laser Scanning					
Regular shape	Pool	419	421	2	0.004
	Road	1807	1804	3	0.001
	Rooftop	875	877	2	0.002
	Water tank	584	591	7	0.01
	Dam	904	900	4	0.004
Irregular shape (Occlusion)	Smoke	20509	20518	9	0.0004
	Cloud	1339	1336	3	0.002
	Building	617	609	8	0.01
	Tree	63	65	2	0.03

Terrestrial Laser Scanning					
Regular shape	Window	2.88	2.83	0.05	0.01
	Door	3.63	3.58	0.05	0.01
Irregular shape	Puddle	1.11	1.12	0.01	0.008
Occlusion	Words	0.74	0.74	0	0
	Roadblock	0.58	0.56	0.02	0.03

3.3.3.2 Performance of Hole Detection between ALS and TLS

There is no big difference between the overall performance of hole detection using Persistent Homology on ALS and TLS. However, sometimes noise is inevitably occurred in the ALS LiDAR data because of the high flying height and lower resolution relatively. In some of the hole detection result, the noise is also recognized as points composed of the hole boundary mistakenly. Figure 27 illustrated an example that the detection result of swimming pool from an ALS point cloud includes a point which is noise in the middle of the swimming pool (It could also be the return of water body if the water from the pool contains other materials that can reflect the signal). Persistent Homology regards it as the part of hole because during the process of filtration, the point in the middle also connects with other points on hole boundary and becomes a part of the simplicial complex.

Yet, the existence of noise would not affect the detection result too much. The same hole and similar lifetime would still be extracted and identified, but the simplicial complex formed would be a little bit different because of the additional point in the middle. Figure 28 shows the detection results of point cloud with noise and without noise. In the point cloud with noise, the hole is appeared and filled up when the radius of ball is 0.178 and 0.762 respectively. In the point cloud without noise, the hole appeared and filled up at the same value of radius with the point cloud with noise. This proves that even Figure 27 includes the noise as hole, the detection result would not change a lot because of the characteristic of Persistent Homology.

Therefore, although there is not much noise nearby or contain in the hole of TLS data, since the setting up of laser scanner is closed to the study area and no interruption appeared during data collection, the proposed method have similar performance on ALS and TLS. Persistent Homology successfully extract the hole boundary in point cloud.

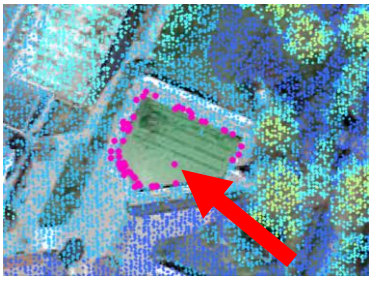
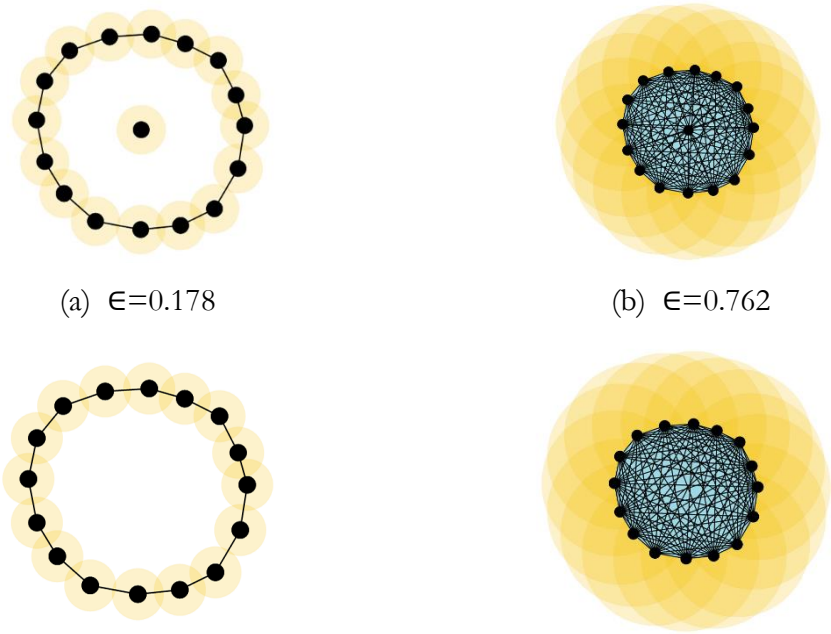


Figure 27: Noise recognized as points of topological hole (red arrow)



(a) ∈=0.178 (b) ∈=0.762

(c) ∈=0.178 (d) ∈=0.762

Figure 28: (a) and (b) Point cloud with noise in the middle; (c) and (d) Point cloud without noise

3.4 Chapter Summary

In this chapter, a hole detection approach in point cloud using Persistent Homology was presented as the existing methods has been reported with limitations. The proposed method includes the application of Persistent Homology and Persistent Entropy for identifying and extracting the topological hole from a set of point cloud. The method is tested in both airborne and terrestrial LiDAR data. Various types of hole, including regular shape, irregular shape and occlusion, are tested. The experiment result has good performance that all of the difference between area of detected holes and holes in point cloud is less than 1%. The proposed method can perform hole detection in point cloud regardless of the types of data and types of hole. Moreover, the detection result is stable that it is not affected by the small noise. To conclude, the proposed approach successfully detect the hole in point cloud with the merit of stable and flexible with different kind of

data.

Chapter 4

4. Hole Classification based on Maximum Likelihood Estimation

This chapter presents the approach for hole classification by Maximum Likelihood Estimation. Three types of holes would be classified, include glass window, occlusion of building, and rooftop. The feature sets use for data training are the 3D area, longest distance between any two points and the standard deviation of the hole. For experimental testing, holes refer to window, occlusion of building, rooftop and other types are applied, which are new data different from the training data. The whole approach is achieved by Python. The assessment of accuracy on the classification result is also conducted to evaluate the performance of the Maximum Likelihood Estimation model. Finally, the chapter ends with a short summary of the proposed method and its result.

4.1 Methodology

4.1.1 Overall Workflow

To carry the hole classification in point cloud, the data has to be trained first (refer to Section 4.1.2) through the determination of feature sets and the data collection of feature sets (refer to Section 4.1.2.1). With the data collected, the computation of mean matrix and variance-covariance matrix of the feature sets are conducted (refer to Section 4.1.2.2). After collecting and computing the training data, classification can be conducted (refer to Section 4.1.3). The unknown detected hole is input and the probability of it belonging to glass window, occlusion of building, or rooftop are computed respectively using matrix calculated in Section 4.1.2.2 and the likelihood model. The classification result is produced by the comparison of the largest probability.

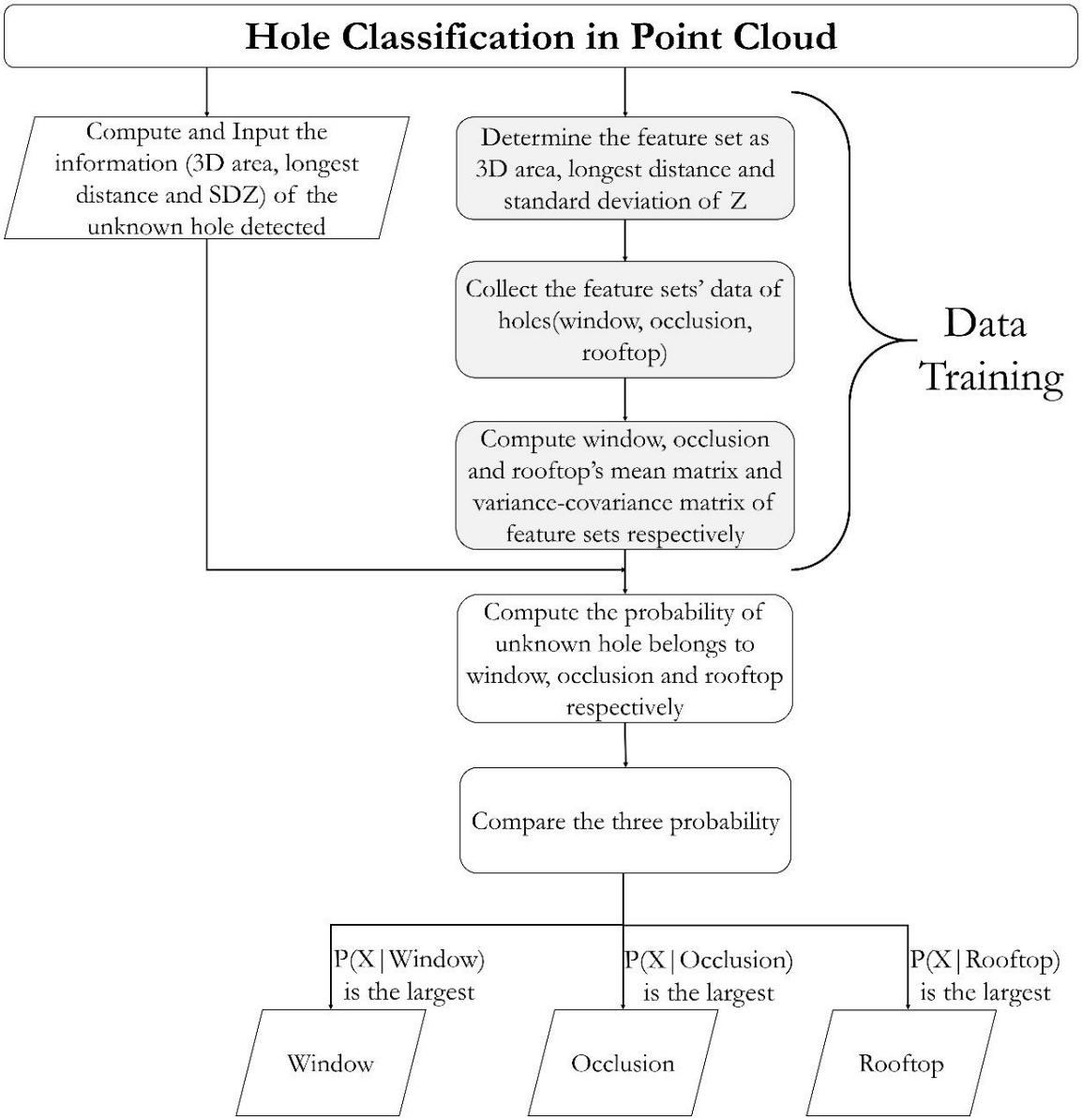


Figure 29: Overall workflow of hole classification (Chapter 4)

4.1.2 Data Training

4.1.2.1 Determination and Collection of Feature Sets

To train the data, feature selection is needed. The features selected help distinguishing different kinds of object. Neither too many nor too few features should be selected, otherwise, the processing time is long or the accuracy of the classification is slow.

In this thesis, the 3D area and the longest distance between any two points of the hole are selected to evaluate the size and shape. In some cases, a hole with large 3D area would have long longest distance, yet, under some situations, although two holes have the same area, they may have different value of longest distance. Figure 30 shows an extreme example. Moreover, standard deviation of the Z coordinates of points that composed of the holes is selected to identify whether the hole is flat or not.

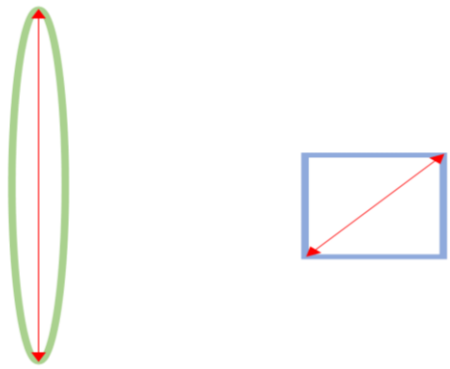


Figure 30: Two holes with same area but different longest distance

Longest Distance

To obtain the longest distance between any two points of the hole, the 3D distance between all pairs of points in the hole are calculated by the below formula and the largest distance is extracted.

$$Distance = \sqrt{(x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2} \tag{15}$$

It is achieved by the following Python Script:

Python Script 4: Compute the longest distance between any two points of the hole

```
def caldistance(x1, y1, z1, x2, y2, z2):
        d = math.sqrt(math.pow(x2-x1,2)+math.pow(y2-y1,2) +math.pow(z2 - z1, 2)* 1.0)
        return d
distance=[]
        length_hole_feature=len(x_hole_feature)
        for h in range(length_hole_feature):
            for i in range(length_hole_feature-h):
                if h!=i+h:
                    distance.append(caldistance(x_hole_feature[h], y_hole_feature[h],z_hole_feature[h],x_hole_feature[i+h], y_hole_feature[i+h], z_hole_feature[i+h]))
        LDistance=max(distance)
```

3D Area

To obtain the 3D area of significant holes detected in the point cloud, the following method is used:

Step 1. Build the triangulated mesh. In this thesis, ball pivoting algorithm (BPA) is used. The radii of the ball designed are the longest distance, longest distance/6, longest distance/3 and longest distance*3.

Step 2. Extract the vertices of each triangles that formed the mesh and calculate their area respectively by:

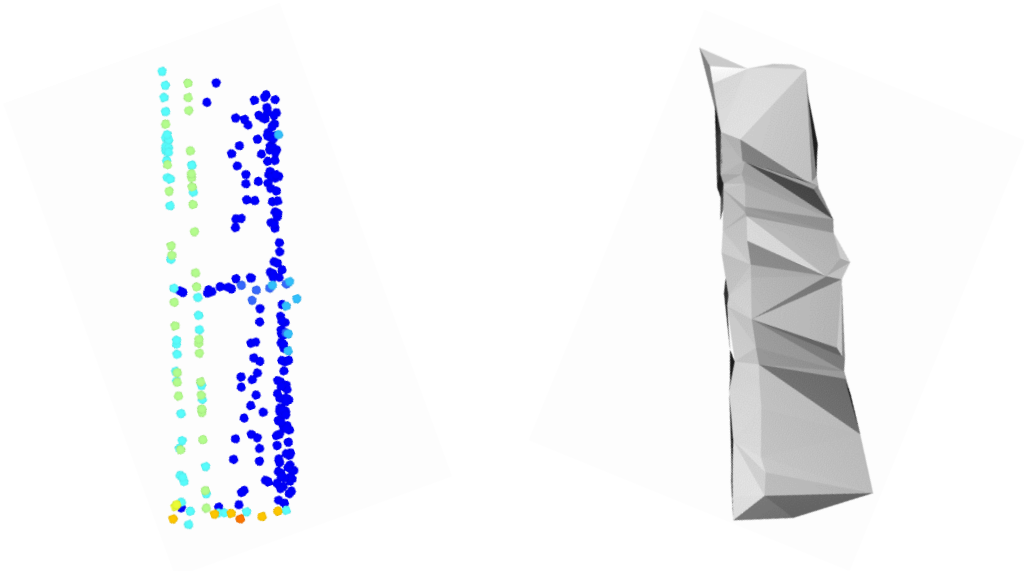
$$s = \frac{(a + b + c)}{2} \tag{16}$$

$$Area = \sqrt{s * (s - a) * (s - b) * (s - c)} \tag{17}$$

where,

a, b and c are the three sides of the triangle.

Step 3. Sum up the area of all triangle, which equals the total area of the hole detected.



(a) (b)

Figure 31: (a) Original point cloud of Dam; (b) Meshed point cloud of Dam

The Python Script is as follow:

Python Script 5: Obtain the 3D area of holes

```
def cal_tri_area(hole_xyz):
    x1=hole_xyz[0,0]
    y1=hole_xyz[0,1]
    z1=hole_xyz[0,2]
    x2=hole_xyz[1,0]
    y2=hole_xyz[1,1]
    z2=hole_xyz[1,2]
    x3=hole_xyz[2,0]
    y3=hole_xyz[2,1]
    z3=hole_xyz[2,2]
    a = math.sqrt(math.pow(x2 - x1, 2) +math.pow(y2 - y1, 2) +math.pow(z2
```

```
- z1, 2)* 1.0)
        b = math.sqrt(math.pow(x2 - x3, 2) +math.pow(y2 - y3, 2) +math.pow(z2 - z3, 2)* 1.0)
        c = math.sqrt(math.pow(x3 - x1, 2) +math.pow(y3 - y1, 2) +math.pow(z3 - z1, 2)* 1.0)
        s = (a+b+c)/2
        area = math.sqrt((s*(s-a)*(s-b)*(s-c)))
        return area
pcd = o3d.io.read_point_cloud(input_path,format='xyz')
pcd.estimate_normals(o3d.geometry.KDTreeSearchParamHybrid(radius=0.1, max_nn=30))
radius = LDistance
radii = [radius/6,radius/3,radius, radius*3]
bpa_mesh=o3d.geometry.TriangleMesh.create_from_point_cloud_ball_pivoting(pcd, o3d.utility.DoubleVector(radii))
triver=np.asarray(bpa_mesh.triangles)
sum_area=0
for count in range(len(triver)):
        tri=triver[count]
        hole_xyz=np.empty([3,3], dtype=float)
        hole_xyz[0,0]=point_cloud[tri[0],0]
        hole_xyz[0,1]=point_cloud[tri[0],1]
        hole_xyz[0,2]=point_cloud[tri[0],2]
        hole_xyz[1,0]=point_cloud[tri[1],0]
        hole_xyz[1,1]=point_cloud[tri[1],1]
        hole_xyz[1,2]=point_cloud[tri[1],2]
        hole_xyz[2,0]=point_cloud[tri[2],0]
        hole_xyz[2,1]=point_cloud[tri[2],1]
        hole_xyz[2,2]=point_cloud[tri[2],2]
        sum_area=sum_area+cal_tri_area(hole_xyz)
```

Standard Deviation of Z

To obtain the standard deviation of Z, the equation of standard deviation has to be followed:

$$\sigma_z = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{18}$$

It can be achieved by the Python Script below:

Python Script 6: Obtain the standard deviation of Z

```
z_hole2=np.array(z_hole_feature)
std_Z=np.std(z_hole2)
```

4.1.2.2 Computation of Mean Matrix and Variance-Covariance matrix

After obtaining the feature sets of each type of hole, the mean matrix and the variance-covariance matrix of the feature sets.

To compute the mean matrix of each feature set, the following step is:

Step 1. Compute the mean of each feature sets by:

$$\bar{x} = \frac{\sum_{i=1}^{n} x}{n} \tag{19}$$

where n=number of data

Step 2. Compute the mean matrix M of feature set. M is a 'n' by 1 matrix. As there are three feature sets (n=3), including 3D area, longest distance and standard deviation of Z, the mean matrix M is a 3 by 1 matrix.

$$M = \begin{bmatrix} \bar{x}_{Area} \\ \bar{x}_{LD} \\ \bar{x}_{\sigma_z} \end{bmatrix} \tag{20}$$

To compute the variance-covariance matrix, the following step is:

Step 1. Compute the variance of each feature sets:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \tag{21}$$

Step 2. Compute the variance-covariance matrix V. As V is a 'n' by 'n' matrix, here a 3 by 3 matrix is computed.

$$V = \begin{bmatrix} V_{Area} & C_{Area,LD} & C_{Area,\sigma_z} \\ C_{LD,Area} & V_{LD} & C_{LD,\sigma_z} \\ C_{\sigma_z,Area} & C_{\sigma_z,LD} & V_{\sigma_z} \end{bmatrix} \tag{22}$$

The above procedures are obtained by the following Python script (occlusion of building as an example):

Python Script 7: Computation of mean matrix and variance-covariance matrix

```
#Calculate the mean
mean_occlusion_3dArea=np.mean(occlusion_3dArea)
mean_occlusion_LDis=np.mean(occlusion_LDis)
mean_occlusion_STDZ=np.mean(occlusion_STDZ)
#Calculate the variance
var_occlusion_3dArea=np.var(occlusion_3dArea,ddof=1)
var_occlusion_LDis=np.var(occlusion_LDis,ddof=1)
var_occlusion_STDZ=np.var(occlusion_STDZ,ddof=1)
#Calculate the mean matrix
mean_matrix_occlusion=np.array([[mean_occlusion_3dArea],[mean_occlusio
n_LDis],[mean_occlusion_STDZ]])
#Calculate the variance-covariance matrix
vcv_matrix_occlusion=np.cov(np.array([occlusion_3dArea,occlusion_LDis,oc
clusion_STDZ]))
```

4.1.3 Classification by Likelihood Model

4.1.3.1 Calculation of Probability

After calculating the mean and variance covariance matrix of the feature sets, classification is conducted by calculating the probability of the unknown hole belongs to each type of hole first, that are P(X|Window), P(X|Occlusion), P(X|Rooftop). They are calculated by the likelihood model:

$$P(X|A) = \frac{1}{(2pi)^{\frac{n}{2}}|V|^{\frac{1}{2}}} exp\left[-\frac{1}{2}[X-M]^T V^{-1}[X-M]\right] \tag{23}$$

In practice, log-likelihood would be applied for easier maximization by taking log of the

likelihood model.

$$P(X|A) = -\frac{1}{2} log_e|V| - \left[\frac{1}{2}[X-M]^T V^{-1}[X-M]\right] \tag{24}$$

In the following Python Script, the probability of the unknown hole belongs to the occlusion of building is demonstrated as an example.

Python Script 8: Calculation of the probability of unknown hole belongs to occlusion of building

```
#Calculate the probability
unknown_data=np.array([[unknown_3darea],[unknown_LDis],[unknown_STDZ]])
#[X-M]^T
transposeX_M_occlusion=(unknown_data-mean_matrix_occlusion).transpose()
#V^-1
inverse_vcv_occlusion=inv(vcv_matrix_occlusion)
#[X-M]^T*V^-1
Times_occlusion=(transposeX_M_occlusion).dot(inverse_vcv_occlusion)
#[X-M]^T*V^-1*[X-M]
Times_data_occlusion=Times_occlusion.dot(unknown_data-mean_matrix_occlusion)
Times_data_occlusion = Times_data_occlusion[0]
#(1/2)*[X-M]^T*V^-1*[X-M]
tempPX_occlusion=(1/2)*Times_data_occlusion
#|V|
det_vcv_occlusion=np.linalg.det(vcv_matrix_occlusion)
#loge|V|
log_det_vcv=math.log(det_vcv_occlusion)
#(-1/2)*loge|V|
log_times_e=(-1/2)*log_det_vcv
PX_occlusion=log_times_e-tempPX_occlusion#(-1/2)*loge|V|-(1/2)*[X-M]^T*V^-1*[X-M]
```

4.1.3.2 Comparison and Classification

After the calculation of the probability, the unknown hole is classified by comparing which

probability is the largest. The type with largest probability shows the classification result. The comparison is conducted through the Python script below.

Python Script 9: Comparison of the three probability and Output the classification result

```
#Classification
classification=np.array(['Occlusion','Window','Rooftop'])
probability=np.array([PX_occlusion,PX_window,PX_roof])
index_max = np.argmax(probability)
classify=classification[index_max]
print('The hole detected is %s\n'%classify)
```

4.2 Experimental Testing

4.2.1 Design of Experiment and Evaluation

The hole classification aims to classify three types of hole, including window, occlusion of building and rooftop. Following the method in Section 4.1, the longest distance, 3D area and standard deviation of Z of unknown hole being classified and holes belongs to window, occlusion of building and rooftop are computed. Their information is then applied into the likelihood model to perform the classification. To evaluate the performance of classification model, totally 7 unknown holes would be tested, and the accuracy of the model is measured by:

$$accuracy = \frac{correct\ \ predictions}{all\ \ predictions} \tag{25}$$

The experiment is carried out by Python. The entire procedure is stated as below:

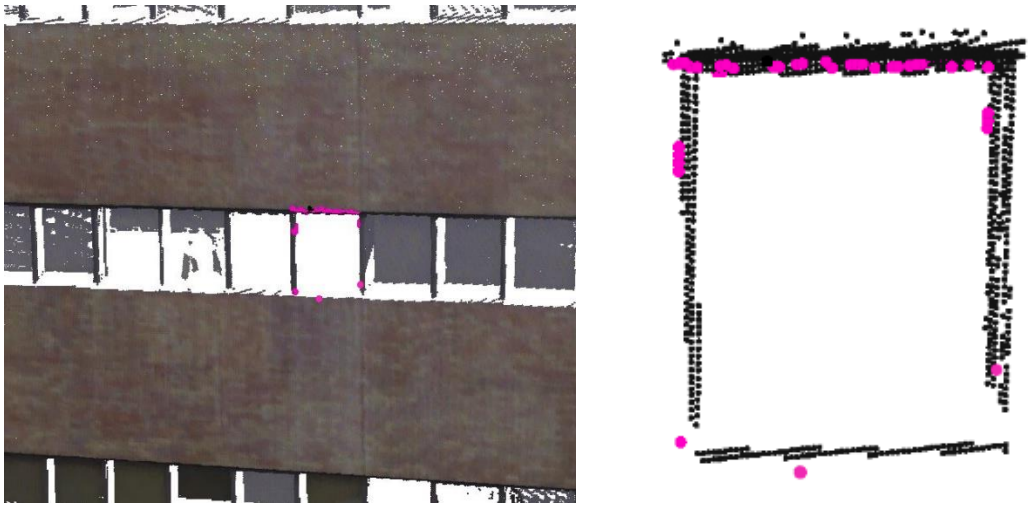
1. Select the training data and compute their feature sets' data
2. Compute the mean matrix and variance-covariance matrix of the training data
3. Input the point cloud with hole to perform hole detection by Persistent Homology
4. Compute the feature sets' information of the detected hole
5. Apply the computed information into the likelihood model to conduct hole classification
6. Verify the result by input the detected hole into ArcScene and check its type of hole
7. Calculate the accuracy of the machine learning model

4.2.2 Training Data

There are five data selected in 'window', 'rooftop' and 'occlusion of building' as the training data respectively. The data are independent that window from the same building would not be extracted twice as training data, the five holes of window selected are come from different buildings. The unknown hole for tested are also independent and different from the trained data. The information of all training data is also presented in Table 2: Information of training data.

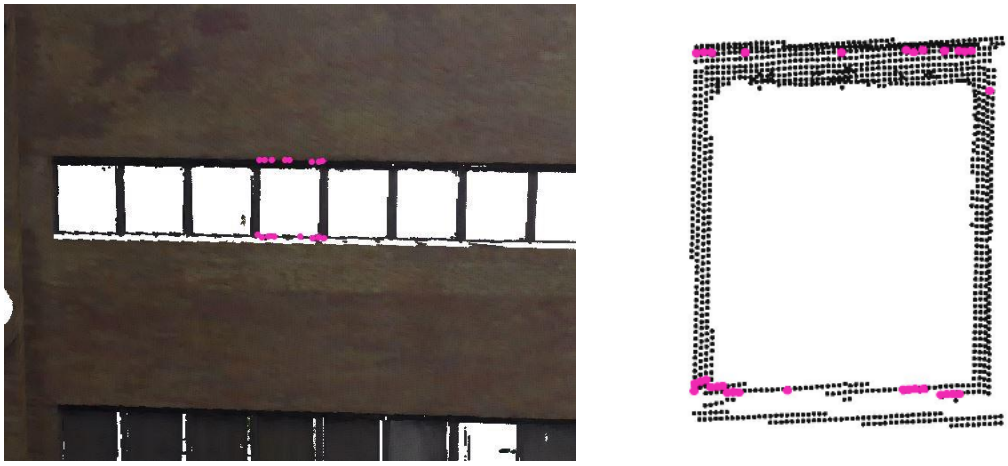
4.2.2.1 Window

In this part, the training data of window are displayed with diagram (a) shows the overlapping of point cloud and the hole detected (pink) and diagram (b) shows the overlapping of the hole in point cloud and the hole detected (pink).



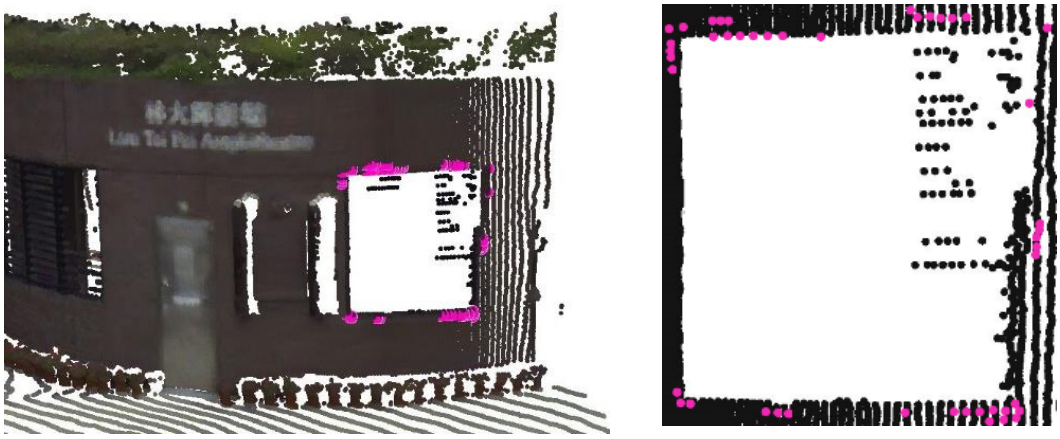
(a) (b)

Figure 32: (Training data) Hole of window 1 with 1.56m in longest distance, 1.14m^2 in 3D area and 0.27 in standard deviation of Z



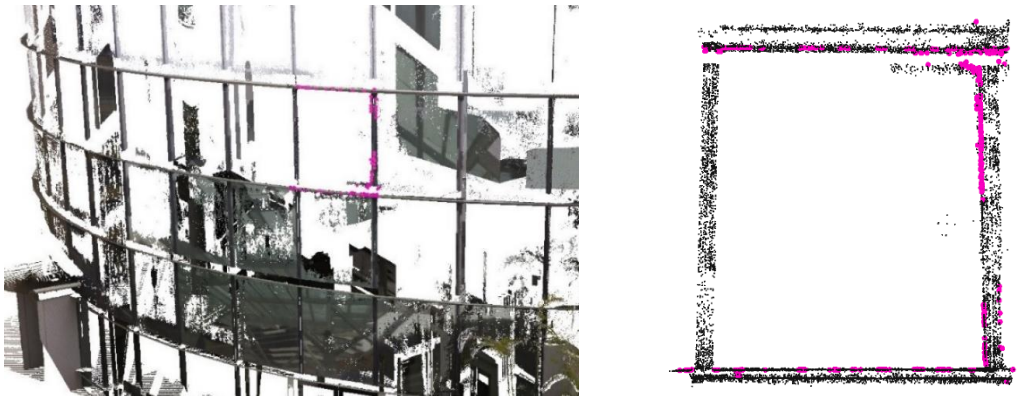
(a) (b)

Figure 33: (Training data) Hole of window 2 with 2.39m in longest distance, 1.13m^2 in 3D area and 0.80 in standard deviation of Z



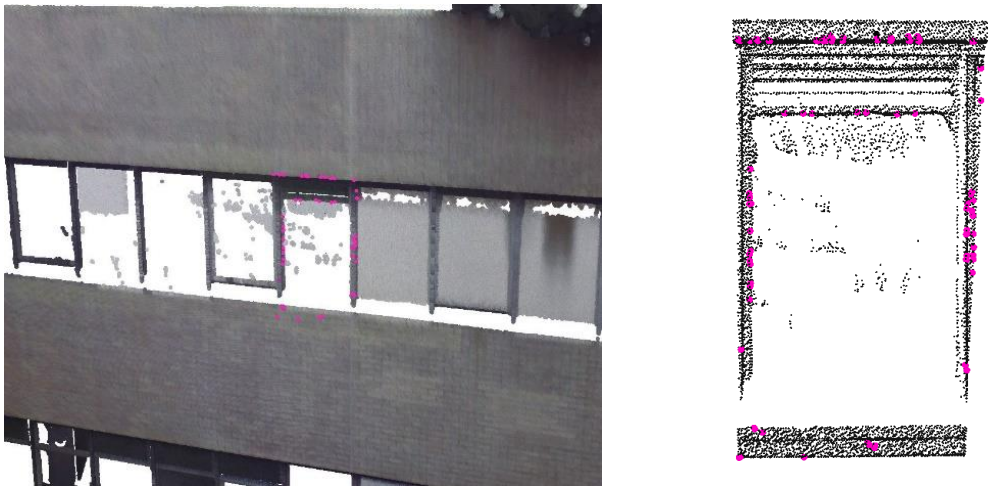
(a) (b)

Figure 34: (Training data) Hole of window 3 with 2.61m in longest distance, 2.80m^2 in 3D area and 0.74 in standard deviation of Z



(a) (b)

Figure 35:: (Training data) Hole of window 4 with 2.11m in longest distance, 3.21m^2 in 3D area and 0.60 in standard deviation of Z

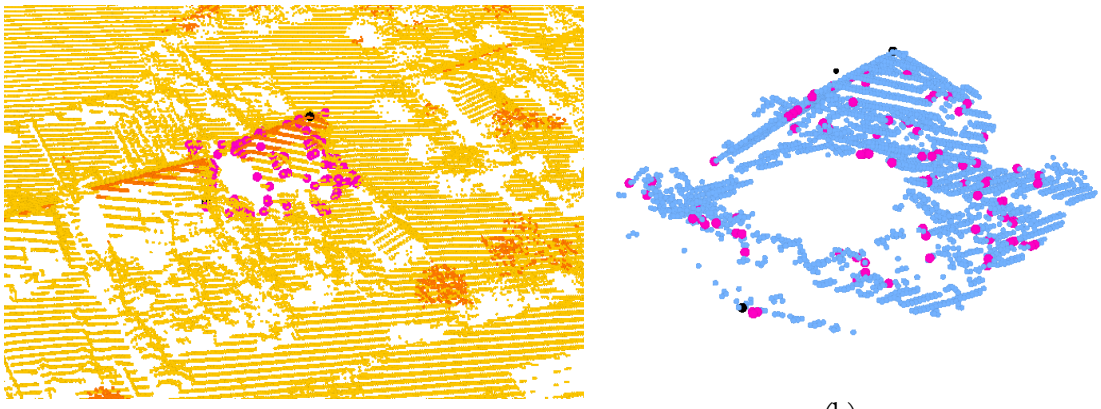


(a) (b)

Figure 36: (Training data) Hole of window 5 with 1.59m in longest distance, 1.80m^2 in 3D area and 0.58 in standard deviation of Z

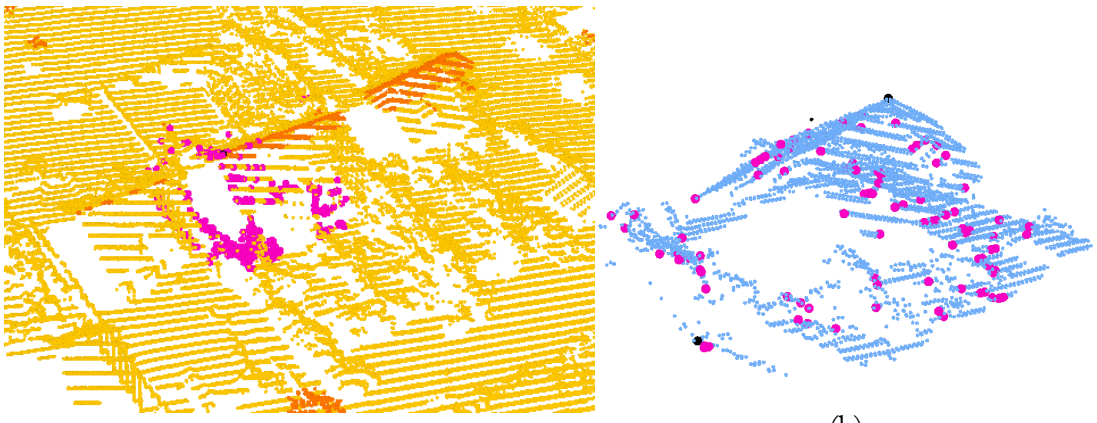
4.2.2.2 Occlusion of Building

In this part, the training data belongs to occlusion of building are displayed with diagram (a) showing the overlapping of point cloud and hole detected, while diagram (b) shows the overlapping of the hole in point cloud and the hole detected (pink).



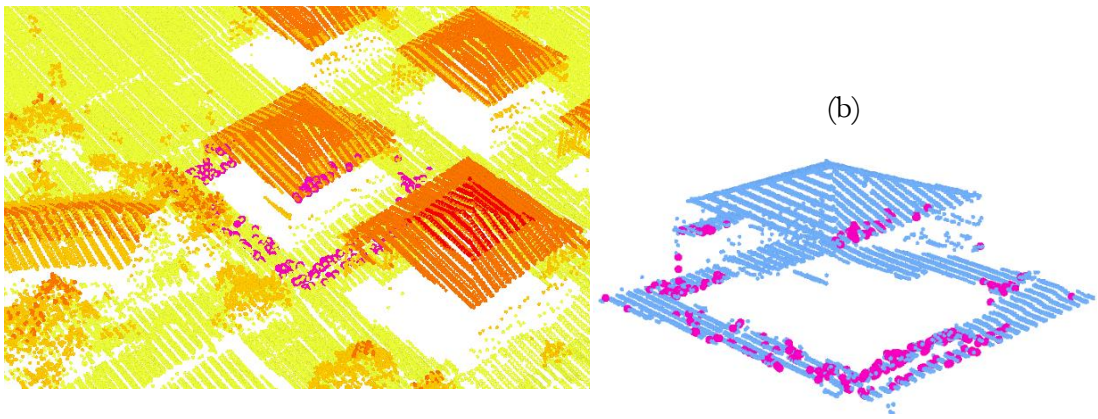
(a) (b)

Figure 37: (Training data) Hole of building's occlusion 1 with 31.92m in longest distance, 616.67m^2 in 3D area and 3.34 in standard deviation of Z



(a) (b)

Figure 38: (Training data) Hole of building's occlusion 2 with 34.24m in longest distance, 819.09m^2 in 3D area and 2.91 in standard deviation of Z



(b)

(a)

Figure 39: (Training data) Hole of building's occlusion 3 with 33.77m in longest distance, 691.40m^2 in 3D area and 3.43 in standard deviation of Z

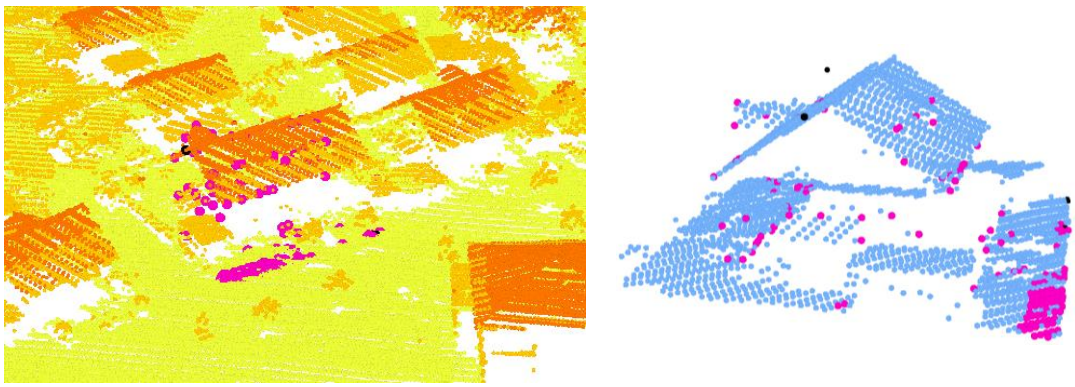


Figure 40: (Training data) Hole of building's occlusion 4 with 33.39m in longest distance, 879.82m^2 in 3D area and 2.97 in standard deviation of Z

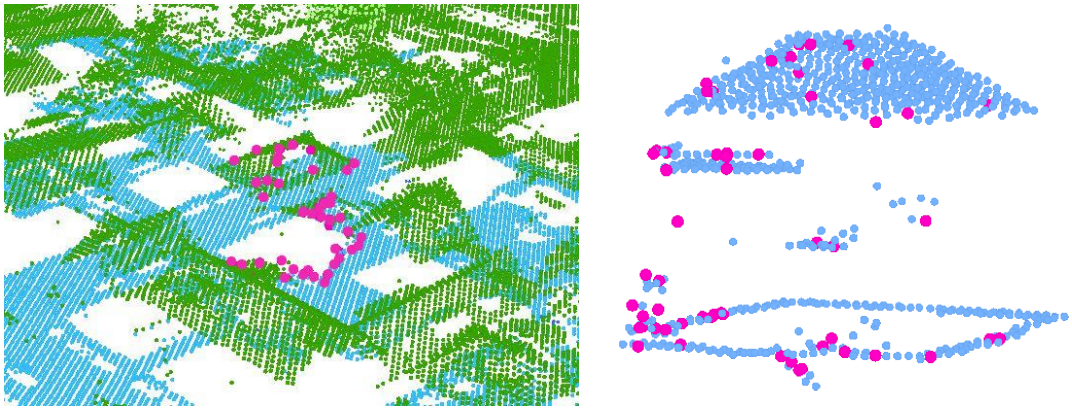
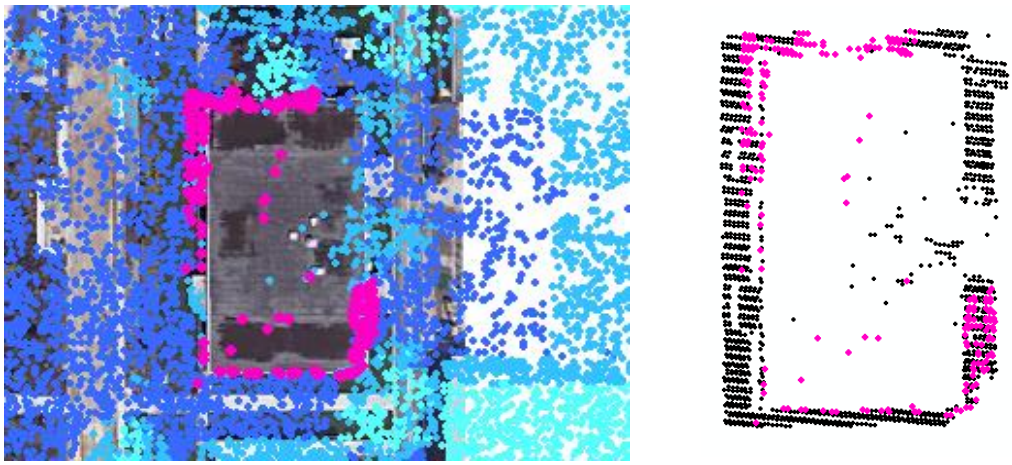


Figure 41: (Training data) Hole of building's occlusion 5 with 20.95m in longest distance, 584.33m^2 in 3D area and 5.01 in standard deviation of Z

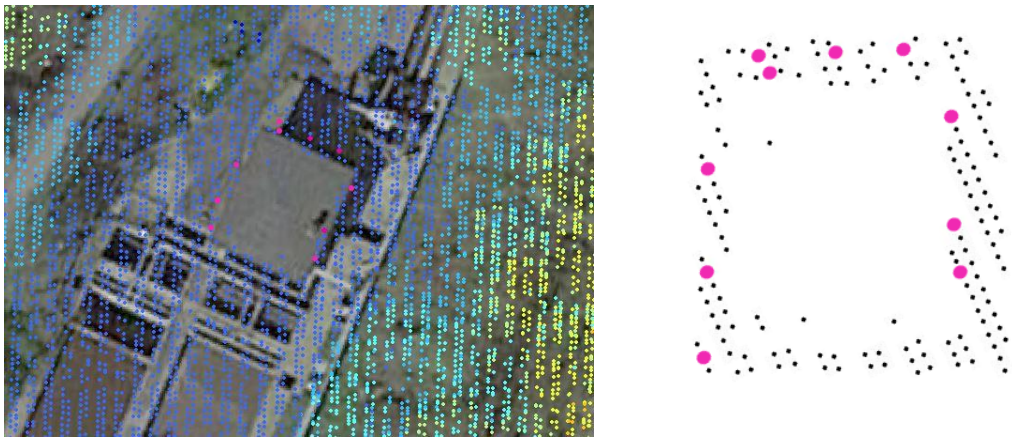
4.2.2.3 Rooftop

In this part, the training data of rooftop are displayed with diagram (a) showing the overlapping of point cloud and the hole detected (pink) and diagram (b) showing the overlapping of the hole in point cloud and the hole detected (pink).



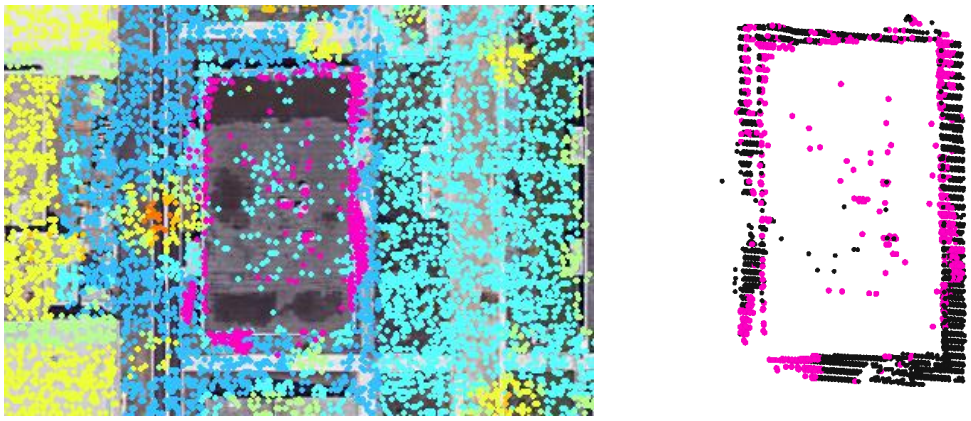
(a) (b)

Figure 42: (Training data) Hole of rooftop 1 with 41.01m in longest distance, 877.54 m^2 in 3D area and 1.62 in standard deviation of Z



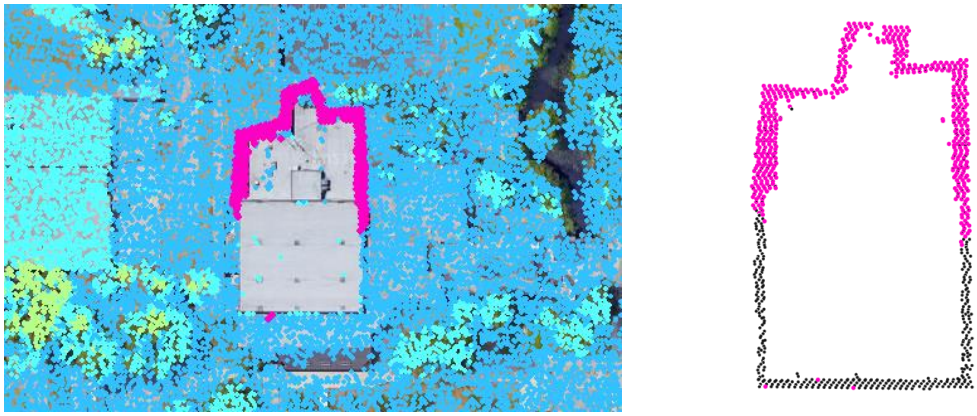
(a) (b)

Figure 43: (Training data) Hole of rooftop 2 with 12.22m in longest distance,66.93m^2 in 3D area and 0.43 in standard deviation of Z



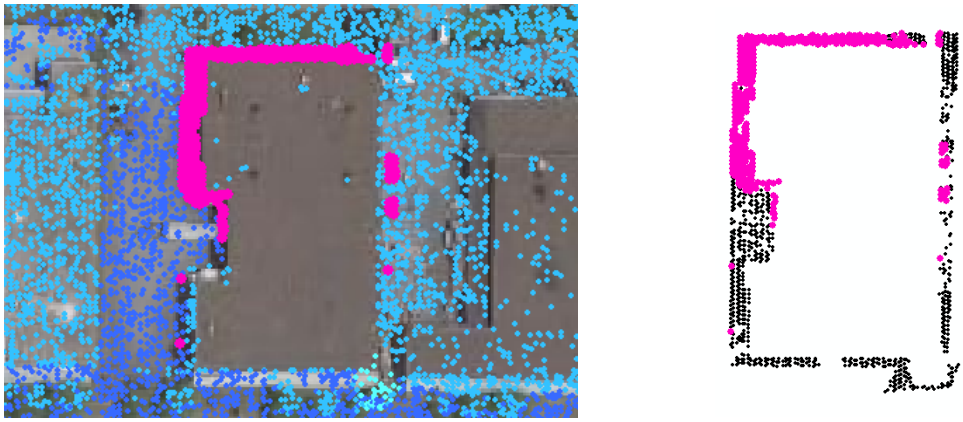
(a) (b)

Figure 44: (Training data) Hole of rooftop 3 with 40.63m in longest distance, 986.64m^2 in 3D area and 1.68 in standard deviation of Z



(a) (b)

Figure 45: (Training data) Hole of rooftop 4 with 80.52m in longest distance, 1724.64m^2 in 3D area and 1.94 in standard deviation of Z



(a) (b)

Figure 46: (Training data) Hole of rooftop 5 with 90.63m in longest distance, 2823.98m^2 in 3D area and 1.02 in standard deviation of Z

Table 2: Information of training data

	Longest Distance (m)	3D Area (m^2)	Standard Deviation of Z
	1.56	1.14	0.27
	2.39	1.13	0.80
Window	2.61	2.80	0.74
	2.11	3.21	0.60
	1.59	1.80	0.58
	31.92	616.67	3.34
	34.24	819.09	2.91
Occlusion	33.77	691.40	3.43
	33.39	879.82	2.97
	20.95	584.33	5.01
	41.01	877.54	1.62
	12.22	66.93	0.43
Rooftop	40.63	986.64	1.68
	80.52	1724.64	1.94
	90.63	2823.98	1.02

4.3 Results and Analysis

In this part, the classification result of 7 unknown holes detected are presented with their visualization and feature sets, where diagram(a) shows the overlapping of point cloud and hole detected (Pink) and diagram (b) provides a detailed version of it. The analysis of the result is carried out too.

From Table 3: Classification result of hole, it shows that the result achieved 86% accuracy in classifying the type of hole. Among 7 tests, 6 of them have a correct classification. For unknown hole no.4, it is mistakenly classified as rooftop instead of occlusion (Figure 50).

The holes of window are successfully classified. By reviewing Table 2: Information of training data, pattern can be observed. Among three types of hole, hole of window has the smallest value of longest distance, 3D area and standard deviation of Z. It is reasonable that window has smaller size than rooftop and occlusion as rooftop and occlusion mostly have similar area as a house. The hole belongs to window also has small standard deviation of Z because the height of window would not be too high that leads to the large standard deviation of Z. Therefore, it is relatively easy to classify the hole of window and the results also prove it that the hole of window is perfectly classified.

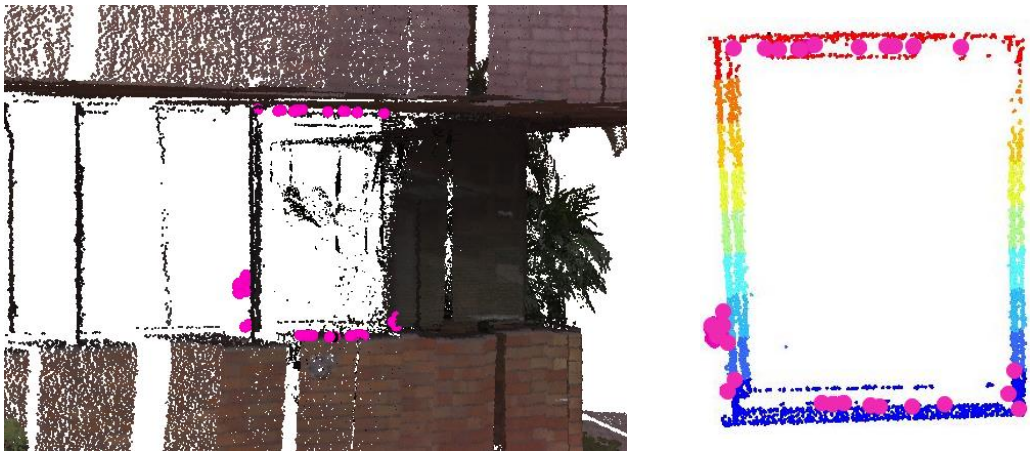
For occlusion, one of the holes fail to be classified. From the information of training data, it can find that occlusion has larger value of longer distance and 3D area than hole of window, but smaller than hole of rooftop. But the standard deviation of Z is the largest among three types of hole. However, there is a data in hole of rooftop that the information is not consistent with the above observation. A rooftop with 12.22m longest distance, 66.93 m^2 3D area and 0.43 standard deviation of z, is added as the training data of rooftop, which is Figure 43. This hole of rooftop has a smaller value of longest distance and 3D area than the average of rooftop. Although it is reasonable that there would be different size of rooftop and different size of hole belongs to rooftop would be detected, it may be the main reason of the improper classification result of unknown hole no.4, which is verified as occlusion but classified as rooftop using this model.

For unknown hole no.4, it has a longest distance with 22.06 m, 390.48 m^2 in 3D area and 3.43 in standard deviation of Z. Its size is generally much smaller than the hole of rooftop, most likely, it would be classified as occlusion with the higher value of standard deviation unless the existence the hole of rooftop in Figure 43. Since one of the rooftop's training data has small value in size and inconsistent with other rooftop's data but consistent with the occlusion relatively, the probability of the unknown hole no.4 belongs to rooftop

become the highest.

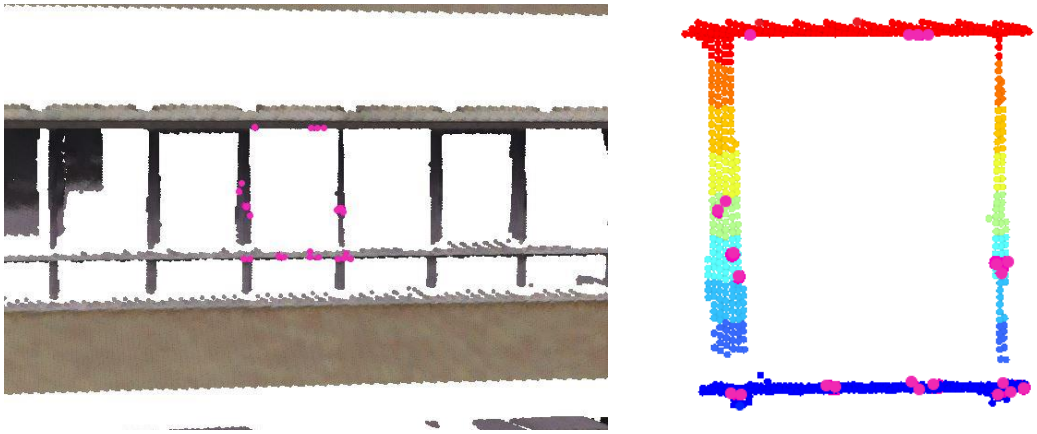
From the above result and analysis, the limitations of using maximum Likelihood Estimation as the tool of hole classification also show. Maximum Likelihood Estimation is efficient for classification by choosing the parameters to maximize the likelihood. However, it is sensitive to the choice of the training data. In this study, the inconsistent selection of training data leads to the mistake in classification. It shows that the selection of training data highly affects the classification result, especially when the sample size is small. The estimates can be biased easily in the case with small size of training data. And thus leads to the wrong classification result in unknown hole no.4. Therefore, it is better not to use Maximum Likelihood Estimation to perform hole classification when the sample size is small.

Overall, the hole classification performed successfully with Maximum Likelihood Estimation with 86% of accuracy. The hole detected by Persistent Homology can be then classified by Maximum Likelihood Estimation, which compensate the limitation of algebraic topology that only dimensions of homology can be classified in Persistent Homology.



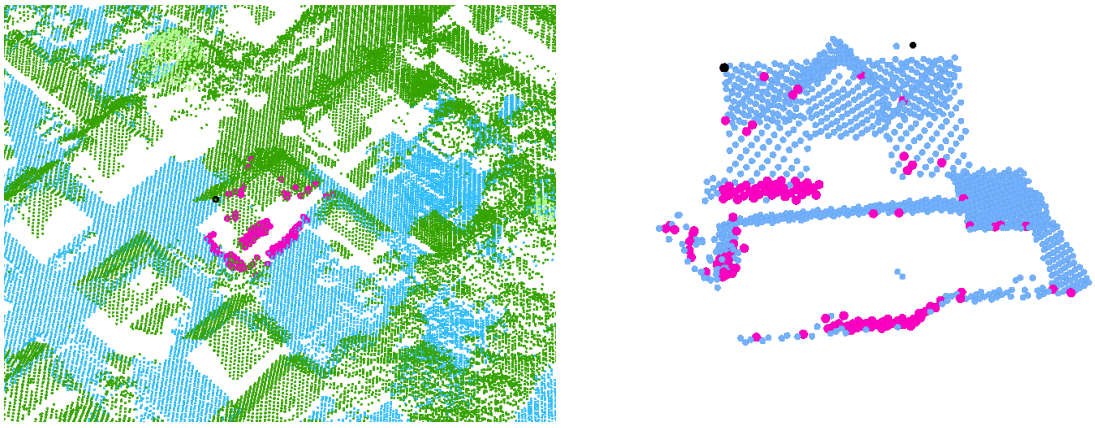
(a) (b))

Figure 47: **Hole of window classified as 'window'**, with 1.58m in longest distance, 1.19m^2 in 3D area and 0.53 in standard deviation of Z (Result 1)



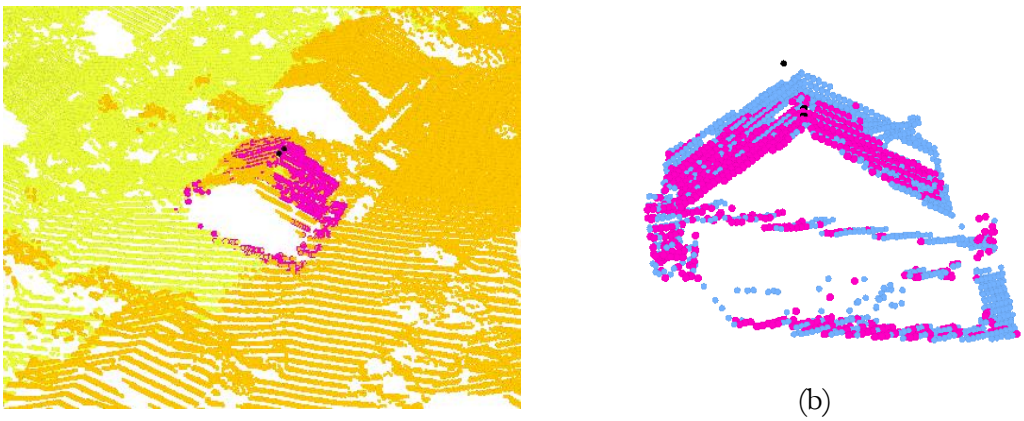
(a) (b)

Figure 48: **Hole of window classified as 'window'**, with 1.72m in longest distance, 1.31m^2 in 3D area and 0.52 in standard deviation of Z (Result 2)



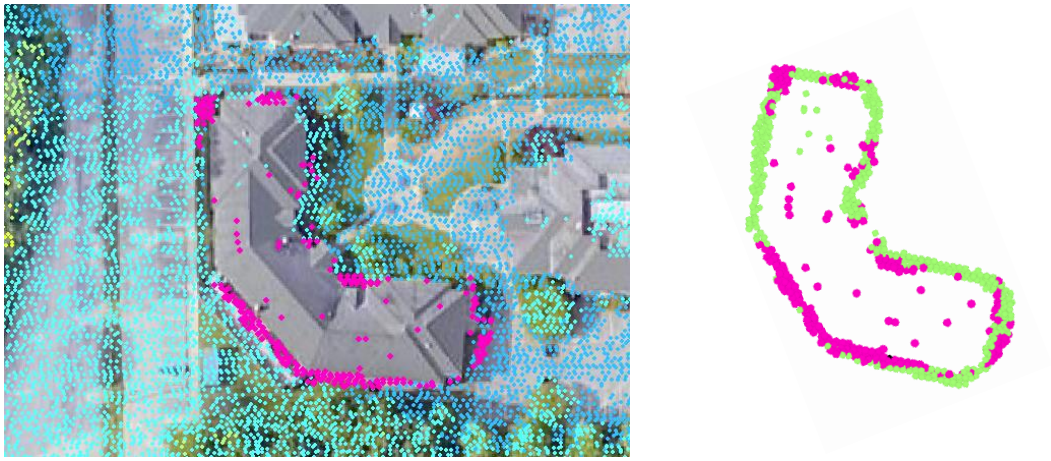
(a) (b)

Figure 49: **Hole of occlusion classified as 'occlusion'**, with 26m in longest distance, 314.21m^2 in 3D area and 4.42 in standard deviation of Z (Result 3)



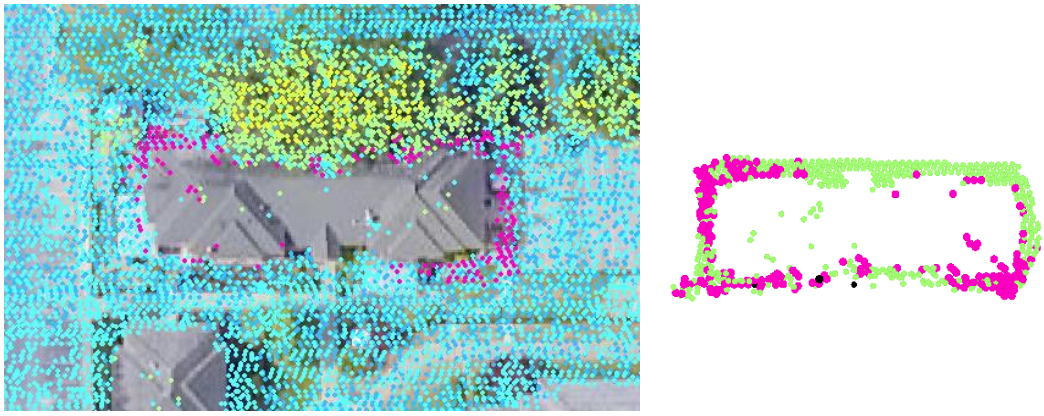
(a) (b)

Figure 50: **Hole of occlusion classified as 'rooftop'**, with 22.06m in longest distance, 390.48m^2 in 3D area and 3.43 in standard deviation of Z (Result 4)



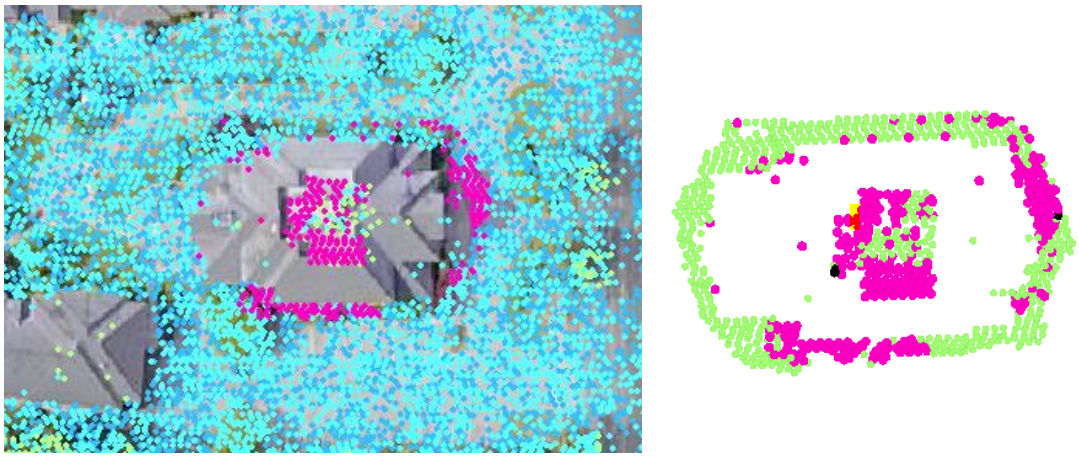
(a) (b)

Figure 51: **Hole of rooftop classified as 'rooftop'**, with 65.81m in longest distance, 1941.38m^2 in 3D area and 2.45 in standard deviation of Z (Result 5)



(a) (b)

Figure 52: **Hole of rooftop classified as 'rooftop'**, with 58.73m in longest distance, 1484.07m^2 in 3D area and 3.56 in standard deviation of Z (Result 6)



(a) (b)

Figure 53: **Hole of rooftop classified as 'rooftop'**, with 45m in longest distance, 1294.11m^2 in 3D area and 4.36 in standard deviation of Z (Result 7)

Table 3: Classification result of hole

Unknown Hole	Longest Distance (m)	3D Area (m^2)	Standard Deviation of Z	Expected Hole Type	Classification Result
1	1.58	1.19	0.53	Window	Window
2	1.72	1.31	0.52	Window	Window
3	26.00	314.21	4.42	Occlusion	Occlusion
4	22.06	390.48	3.43	Occlusion	Rooftop
5	65.81	1941.38	2.45	Rooftop	Rooftop
6	58.73	1484.07	3.56	Rooftop	Rooftop
7	45.00	1294.11	4.36	Rooftop	Rooftop

4.4 Chapter Summary

In this chapter, hole classification based on Maximum Likelihood Estimation is presented. Three types of hole are classified by applying their longest distance between any two points, 3D area and standard deviation of Z into the likelihood model. The proposed method successfully classifies the unknown hole detected from the Persistent Homology with 86% of accuracy. The result shows the efficiency of using Maximum Likelihood Estimation to classify the hole detected in the point cloud, but also reveal the limitation of it, which is the biased estimation when the sample size is small.

Chapter 5

5. Conclusions and Future Works

5.1 Conclusions

As depicted in 1. Introduction, laser scanning has become more common and popular with its efficiency and accuracy. However, the collected point cloud data contain holes by several reasons. The existing research proposed methods, such as detecting the hole in the triangulated point cloud, identifying the hole in point cloud by human eyes, etc. To reduce the computational cost and human involvement, this thesis research attempt to enhance the hole detection approach by proposing detecting the hole in point clouds using Persistent Homology. Further applications of the detection result from Persistent Homology are also achieved. The conclusions of this thesis research include the following:

1. **Persistent Homology performs well in hole detection.**
   The current approaches mostly require the threshold value or user intervention, or have restriction on data type enable to detect the hole. This thesis attempted to detect the hole in point clouds with the use of Persistent Homology, which is a tool of Topological Data Analysis that provide fully automatic identification of topological features. The algorithm is applied in various type of data and holes, including airborne LiDAR data, terrestrial LiDAR data, hole in regular shape, irregular shape and also occlusion. The successful result provides a new approach in the field of hole detection. With the use of Persistent Homology, the hole can be detected automatically, efficiently and accurately before the application of laser scanning data.

2. **Persistent entropy can be applied for the separation of topological noise and features.**
   The detection result from Persistent Homology includes all homology found in the point clouds, so topological noise and features are also included in the result. Generally, the features with longest lifetime is regarded as the significant hole but it may not be by the experiments conducted in this thesis. So Persistent Entropy is applied by defining the 'long' lifetime and separate the topological hole from noise. The result is successful that significant holes can be output, and for irregular shape of hole, the significant holes extracted can even be union to constitute the hole boundary. The application of Persistent Entropy significantly improves the efficiency of workflow in hole detection.

3. **Maximum Likelihood Estimation assist the classification of detected hole after the application of Persistent Homology.**
   The hole appears in the point cloud can be caused by various reason and different

objects. The detection result from Persistent Homology can be further utilized for classifying the type of hole in point cloud, which is useful that can have many applications. In this thesis, the hole classification is performed by Maximum Likelihood Estimation using the detected hole from Persistent Homology. The result is successful that shows the possibility of hole classification using the existing features of hole detected.

5.2 Future Works

The research presented in this thesis provides a foundation of hole detection in point clouds using Persistent Homology. Further investigations can be done to overcome the limitations and enhance the performance of the proposed approach. Several imperatives for future work are listed as follows:

Restriction on size of point cloud: There is restriction on the size of point cloud input into the library of Persistent Homology, Ripser.Py. Some of the point clouds have to be separated and chopped before applying the Persistent Homology. Research can be conducted to improve the flow of the algorithm and lower the computational memory needed in order to eliminate the restriction on the size of point cloud.

Separation of interference and significant features: The separation of topological features and topological noise is a big topic in Persistent Homology. Separating the interferences and features by lifetime of hole only consider the hole's density of point. Some of the current research proposed Persistence terrace for separation, which consider both the size and point density of hole (Moon et al., 2018). It is more accurate than only considering the point density of hole. Therefore, further research can be done on separating the topological features and noise in order to obtain the best detection result.

Enhancement on hole classification: Persistent Homology is algebraic topology that only focus on the calculation of homology and the shape of homology is not maintained. It is difficult to classify the type of hole by reviewing the detection result from Persistent Homology only. Therefore, the assistance of machine learning compensate the limitation of Persistent Homology. To have better performance on hole classification, more dataset can be used for Maximum Likelihood Estimation, or other approach of machine learning can be selected for more accurate hole classification.

References

[1] Atienza, N., Gonzalez-Diaz, R., & Rucco, M. (2016). Separating topological noise from features using persistent entropy. *Software Technologies: Applications and Foundations,* 9946, 3–12.

[2] Bendels, G., Schnabel, R. & Klein, R. (2006). Detecting holes in point set surfaces. *Journal of WSCG*. 14.

[3] Bird, B., Lennox, B., Watson, S., & Wright, T. (2019). Autonomous hole detection and characterisation in point clouds and triangular meshes.

[4] Bojan R., Borut Ž, & Domen M. (2015). Point density evaluation of airborne LiDAR datasets [data set]. *Verlag der Technischen Universität Graz.*

[5] Branch, J., Prieto, F., & Boulanger, P. (2006). Automatic hole-filling of triangular meshes using local radial basis function. *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06),* 727–734.

[6] Dantchev, S. & Ivrissimtzis, I. (2012). Efficient construction of the Čech complex. *Computers & Graphics,* 36(6), 708–713.

[7] Davis, J., Marschner, S. R., Garr, M. & Levoy, M. (2002). Filling holes in complex surfaces using volumetric diffusion. *First International Symposium on 3D Data Processing Visualization and Transmission (pp. 428-441).* IEEE.

[8] Deshmukh, V. (2019, March 28). Topological Data Analysis - A Very Short Introduction. Medium.

[9] Fang, H. T., & Huang, D. S. (2004). Noise reduction in LiDAR signal based on discrete wavelet transform. Optics Communications, 233(1), 67–76.

[10] Fugacci, U., Scaramuccia, S., Iuricich, F., & Floriani, L. D. (2016). Persistent Homology: a step-by-step introduction for newcomers. *STAG*.

[11] Hutchison, W.; Pyle, D., Mather, T., Biggs, J., & Yirgu, G. (2014): 2012 Aluto LiDAR data. figshare. Dataset.

[12] Jun, Y. (2005). A piecewise hole filling algorithm in reverse engineering. *Computer Aided Design,* 37(2), 263–270.

[13] Mingqiang W., Jianhuang W., & Mingyong P. (2010). An integrated approach to filling holes in meshes. *2010 International Conference on Artificial Intelligence and Computational Intelligence,* 3, 306–310.

[14] Moon, C., Giansiracusa, N., & Lazar, N. A. (2018). Persistence terrace for topological inference of point cloud data. *Journal of Computational and Graphical Statistics,* 27(3), 576–586.

[15] Nguyen, V. S., Trinh, T. H., & Tran, M. H. (2015). Hole boundary detection of a surface of 3D point clouds. *2015 International Conference on Advanced Computing and Applications (ACOMP),* 124–129.

[16] Otter, N., Porter, M. A, Tillmann, U., Grindrod, P. ,& Harrington, H. A. (2017). A roadmap for the computation of persistent homology. *EPJ Data Science,* 6(1), 1–38.

[17] P. Liang, G. Q. Zhou, Y. L. Lu, X. Zhou, X. Zhou, & B. Song. (2020). Hole-filling algorithm for airborne LiDAR point cloud data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.,* XLII-3-W10, 739–745.

[18] Sheffar, D. (2020). Introductory topological data analysis. arXiv preprint arXiv:2004.04108..

[19] Stal, C., Verbeurgt, J., De Sloover, L., & De Wulf, A. (2020). Assessment of handheld mobile terrestrial laser scanning for estimating tree parameters. *Journal of Forestry Research.*

[20] Tabib, R. A., Jadhav, Y. V., Tegginkeri, S., Gani, K., Desai, C., Patil, U., & Mudenagudi, U. (2020). Learning-based hole detection in 3D point cloud towards hole filling. *Procedia Computer Science,* 171, 475–482.

[21] Fiolka, T., Rouatbi, F. & Bender, D. (2017). Automated detection and closing of holes in aerial point clouds using an UAS. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.,* XLII-2-W6, 101–107.

[22] Vosselman, G., & Maas, H. G. (2010). Airborne and terrestrial laser scanning whittles publishing.

[23] Yan, W. Y., & Ryerson University. (2013). Radiometric correction and normalization of airborne LiDAR intensity data for land cover classification.

[24] Yang, L., Yan, Q. & Xiao, C. (2017). Shape-controllable geometry completion for point cloud models. *The Visual Computer,* 33(3), 385–398.

[25] Zomorodian, A. (2010). Fast construction of the Vietoris-Rips complex. *Computers & Graphics,* 34(3), 263–271.