A SEMI-AUTOMATED METHOD OF BUILDING ELEMENT RETRIEVAL FROM POINT CLOUD

A Dissertation
Presented to
The Academic Faculty

By

Shiqin Zeng

In Partial Fulfillment
of the Requirements for the Degree
Master of Engineering in the
School of Civil Engineering and Environment

Georgia Institute of Technology

May 2020

A SEMI-AUTOMATED METHOD OF BUILDING ELEMENT RETRIEVAL FROM POINT CLOUD

Approved by:

Dr. Cho, Advisor
School of Civil Engineering and Environment
Georgia Institute of Technology

Dr. Tsai
School of Civil Engineering and Environment
Georgia Institute of Technology

Dr. Marks
School of Civil Engineering and Environment
Georgia Institute of Technology

Date Approved: April 22, 2020

ACKNOWLEDGMENTS

I would first to thank my thesis advisor Prof. Yong Cho of the Civil Engineering department at Georgia Institute of Technology, for driving my curiosity and interests to a frontier field in construction, continually steering me in the right direction when I ran into the trouble spot and giving me confidence and academic support whenever I needed it. I really appreciate having the thesis study with Dr. Cho.

I would also like to acknowledge my committee members Prof. Marks and Prof. Tsai for their constructive suggestions and guidance for my papers. I must express my very profound gratitude to your patience, insightful comments and encouragement with my research.

Special thanks to Jingdao Chen, for sharing your invaluable time to assist me. This accomplishment would not be possible without your support and efforts. Thanks to my colleagues in RICAL Lab, Danielle and all of the other wonderful staff in the School of Civil and Environmental Engineering.

Finally, I am extremely grateful to my parents for their endless love, caring and support throughout my life. Thanks to Lei Xu for making the delicious and nutritious meals to stimulate my energy to study, especially in this special time. Thanks to my friends for their love and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

3D laser scanning is one of the most accurate approaches that can capture the geometry of free-form shapes on a construction site. The point cloud data from laser scanning can be utilized for site inspection and reverse engineering of building models. One subtask of these applications is building elements retrieval. However, conventional methods require a large database of 3D CAD and BIM models which are not suitable for the case of historical buildings without as-planned 3D models as well as temporary structures that lack corresponding data in the pre-built model. Thus, this paper proposes a semi-automated method to efficiently retrieve duplicate elements without these constraints. First, the geometrical information of each point (XYZ coordinates) was processed with a pre-trained deep learning feature extractor that can generate a 50-dimentional feature vector of each point. Next, the point cloud is grouped into segments using K-means clustering and region-growing algorithms, then built the user-selected interface. The last, an exemplar is provided as input to the retrieval algorithm to determine positive matches among the candidate elements. The results show the proposed method gets the average rates above 90% of precision and recall scores of each point cloud dataset. The proposed method can distinguish the correct building elements form the similarly-shaped candidates and complex building elements. In terms of the applicability, the study shows the proposed method has a certain tolerance of error with different selected instances or boundaries of the selected exemplar and voxel grid resolution. On the other hand, the actual computation time is reasonably fast and efficient.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Introduction

A number of technological advancements have emerged to improve the workflow and efficiency of measurement, building design, land or property assessment, and many other construction operations [1, 2]. Among these advancements, 3D laser scanning stands out as a non-contact, non-destructive technology that can measure millimeter (mm)-level details and capture the geometry of free-form shapes on a construction site in the form of 3D point clouds. While 3D point clouds can also be generated by alternative methods such as photogrammetry [3, 4, 5], 3D laser scanning has been shown to be able to more quickly and accurately generate high-resolution point clouds [6]. 3D point cloud data plays an imperative role for various construction processes such as site quality inspection [7, 8], reverse engineering of building models [9], progress tracking, inventory management, building performance analysis, and building renovation[1].

While the needs of capturing 3D point cloud data for existing buildings and civil infrastructures keep increasing with the techniques of point cloud data acquisition becoming more mature, the process of interpreting the huge quantity of point cloud data remains time-consuming and labor-intensive when applied to building renovation [10], construction progress tracking[11] and preservation of historical buildings [12]. One important subtask of the above practices is building element retrieval which is the task of identifying the exact numbers and precise locations of specific building elements consisting of structural components, MEP (mechanical, electrical, and plumbing) elements, and temporary structures, then converting them into as-built building models. Besides this, object retrieval

allows construction management and relevant stakeholders to perform asset management, progress estimation, and clash detection. Thus, an efficient and precise object retrieval method is necessary to annotate the point cloud's information automatically as the manual detection of these elements is time-demanding and error-prone. Several automated methods have been applied to object recognition based on 3D point cloud data. In some previous research, a model-based approach is used which focuses on registering the scanned point cloud data with CAD models using registration as a proxy step for recognition (e.g. with the Iterative Closest Point algorithm [13, 14]). Other methods perform data-driven Scan-to-BIM by inferring the geometry of different building elements using machine learning techniques[15]. However, these conventional methods require a large database of 3D CAD and BIM models which are not suitable for the case of historical buildings without as-planned 3D models as well as temporary structures that lack corresponding data or inaccurate as-built drawings in BIM. Moreover, such methods may not work very well when there is a significant disparity between the quality, resolution, and completeness of scanned point clouds compared to the designed models. These methods are also not easily extended to building environments having variability in point cloud density, surface roughness, curvature, and clutter within a complex scene [16]. Therefore, further research is necessary to improve the accuracy and efficiency of converting the scanned data into BIM from scratch considering the wide variety in point clouds that may be acquired from different types of infrastructure.

This paper proposes a semi-automated approach to perform exemplar-based building element retrieval from 3D point clouds. The proposed approach is aimed at handling challenging cases such as historical buildings or temporary structures where it is difficult to find exactly matching pre-built 3D models and where model-based retrieval methods do not work well. First, a pre-trained deep neural network is used to extract representative feature vectors for each point in the scanned point cloud. Second, an exemplar building

element is selected from the user interface to serve as the query object. Candidate matches are scored based on three methods of direct voxel grid matching [17], normal vector matching, and feature vector matching to measure the similarity of neighboring instances at the same floor level to that of the exemplar. Finally, a match refinement process is used to group together neighboring detections and eliminate false positives. The proposed method can detect multiple object instances at once in the same point cloud without the need for class-specific training data which reduces the pre-processing overhead of labeling training data while simultaneously maintaining a high precision and recall rate of building element retrieval. In this research, five laser-scanned point clouds captured from three different buildings are used to study the performance of the proposed method. The first point cloud is from a university building which contains columns, doors, windows and other different basic construction elements, the second point cloud is from a historical building that including different type windows and chimney, whereas the third to fifth point clouds are from a building construction site at different stages of construction including beams, slabs, door openings and other as-built building elements. Therefore, by analyzing the retrieval results with the above building elements, we can study a comprehensive and accurate performance assessment based on this proposed method.

1.2 Background

The building element retrieval process aims to find all instances of a query object from a point cloud dataset which can be from object categories such as walls, slabs, doors, etc. Based on previous research, there are four main approaches for building element retrieval from point cloud data, which are geometry-based, rule-based, model-based, and feature-based methods.

1.2.1 Geometry-based object retrieval

Geometry-based object retrieval method refers to plane-fitting and shape-fitting algorithms to recognize construction objects of similar geometry characteristics that consist of planar patches [18, 19, 20], cylindrical structures [21, 22], curvature of pipelines [23] and volumetric representation of building elements [24, 25]. This geometry-based approach is usually performed in the following steps: (1) generate a geometric description of the target object in terms of a parametric plane or curve equation, (2) retrieve the target object in the point cloud by comparing the geometric parameters, (3) examine whether the retrieved objects match the query object. However, this method is not feasible when the target object cannot be described as a simple geometrical shape. Moreover, this approach can also lead to ambiguity in object retrieval as different building element categories may have similar geometrical shape.

1.2.2 Rule-based object retrieval

Rule-based object retrieval uses prior knowledge about object characteristics including size, position, orientation, and topology [26] to be utilized in object retrieval. Rule-based methods are usually human-defined and based on a small number of object characteristics. For example, based on dimensions, relative position, principle direction or normal vector [27], and topological relationships of each element, this method can recover the indoor environment within the rules of the connectivity and containment relations between interior spaces [28]. It is also possible to build damage indicators to identify the extent of damage to each building object by considering multiple hard-coded descriptors [29, 30]. This approach is suitable for cases where the constraints can be defined based on hard-coded prior knowledge. However, it is challenging to extend this approach to classify complex building environments as it is difficult to define robust and distinct rules for complex-geometry elements.

1.2.3 Model-based object retrieval

The model-based object retrieval method can automatically retrieve building elements from point cloud data based on a reference 3D CAD/BIM model [7, 31, 32]. It is commonly used for geometry quality inspection [7, 33, 34], and progress tracking during the construction phase [35, 36]. This approach consists of two main procedures. First, registration and integration of the 3D model with point cloud data using the Iterative Closest Point (ICP) algorithm. This concept involves performing global registration [13, 14], then followed by the local registration to refine the position estimation [35]. Second, each point is matched with that of the building element in the reference BIM model based on the geometric and semantic features of the points. Thus, it can be utilized in reverse engineering to automate the creation of building information models for existing buildings. For example, a set of university campus buildings and a healthcare training facility have been modeled into a semantically-rich BIMs with adequate geometry for energy simulation and integration of data for building operation and facility management by using 3D CAD drawings [37]. This method can achieve high accuracy in the retrieval process as long as the point cloud data do not have huge discrepancies from the reference model datasets. However, this approach has certain requirements in terms of the reference 3D CAD/BIM model and the scanned point cloud quality [11]. In addition, fully-automated registration of resultant models to the point cloud data is computationally challenging due to the large size of point clouds acquired from the buildings. On the other hand, semi-automated registration approaches have been proposed [38], but they still have the limitation of demanding a close match between the scanned data and 3D models. Another solution is to perform a large-scale search on the web to get an indicative 3D model [39]. However, this only achieves a rough approximation since there may not be exact matches available.

1.2.4 Feature-based object retrieval

Feature-based object retrieval methods use machine learning algorithms to train 3D feature descriptors that can classify the point cloud data into object categories. Feature-based methods are usually data-driven and based on a large number of object characteristics. Generally, local descriptors are used to store features at the point level, while global descriptors are used to store features at the object level [40]. These features, which incorporated statistics about distance, area, and angle [41], can be defined under the machine learning framework that can compute robust features that uniquely describe each building element. There are two strategies in this feature-based method. One of them is point-level classification, which extracts the local features of each point, then classifies them individually into an object category. For example, by calculating geometric and color features of each point, a two-class support vector machine (SVM) classifier can recognize the rebar from the point cloud data of a reinforced precast concrete element [42]. The other strategy is segment-level classification which first requires subdividing the point cloud into meaningful segments corresponding to different building components. Then, these segments are classified into different object classes. This strategy can handle complex geometries such as construction equipment [43] and build a multi-view incremental point classification to improve the accuracy by decreasing the effect caused by noise and occlusion from the scanned point cloud data [44]. Furthermore, deep learning techniques can reduce the need to manually design feature descriptors and has more robust classification rules [15, 45]. This method can retrieve the geometrical structure of objects that is robust to noise and small variations between similar objects and can be generalized to unknown objects. However, it requires more effort to acquire a large database of building element models to train the machine learning algorithm.

In situations where the test environment consists of different classes of objects from the training data, directly applying a network trained on classification may not work well. In-

stead, the strategy of metric learning is used, where a neural network is tasked to learn useful semantic representations of data that can distinguish between similar objects and dissimilar objects [46, 47, 48]. This strategy is often used in image retrieval applications such as face recognition [47] or retrieval of similar images from a database [46]. These methods are still mostly applied in the image domain and have not yet been sufficiently explored in the point cloud domain.

The proposed approach in this study is to directly use point cloud data to create a query object and retrieve matching instances in the point cloud based on the idea of metric learning. Compared to geometry-based methods, the proposed method in this paper can classify building elements with complex geometrical shape characteristics. Moreover, this method does not require pre-built CAD/BIM models since it is based on user exemplar selection from the laser-scanned point cloud itself, which reduces the preprocessing time and data demands compared to model-based methods. Last but not least, this method can extend to complex building environments and generalize better to unknown objects, including temporary structures and equipment, when compared to rule-based approaches.

CHAPTER 2
TECHNICAL APPROACH

This research proposes a building element retrieval method[49] where a user-selected exemplar is manually selected firstly, then serves as the query object from which similar object instances can be automatically retrieved through feature comparison from the original point cloud data. Therefore this proposed methodology is semi-automated since it needs us to select a sample manually. The approach consists of the following five main steps: (i) point-level feature computation after downsampling the input point cloud data, (ii) point cloud segmentation with the computation with features (50 dimensional) of each point, (iii) exemplar selection with the built user interface (especially lasso selection), (iv) candidate element matching , and (v) match refinement with the retrieval visualization. Each step will be explained in detail in the following subsections. Figure 2.1 shows the workflow of this proposed building element retrieval method.

2.1 Point-level Feature Computation

The point cloud data is first processed to convert the raw point cloud, which only has XYZ-geometry and RGB-color information, into a more semantically rich representation. State-of-the-art research [15, 50, 51] has shown that deep neural networks have the potential to compute discriminative and robust features from point cloud data. Thus, in this research, a deep neural network is used as the feature extractor to compute for each point a representative vector which implicitly represents the surface geometry around the point such as surface normal, smoothness, and curvature. The advantage of using deep learning is that the features do not have to be explicitly defined but can be directly learned from data.

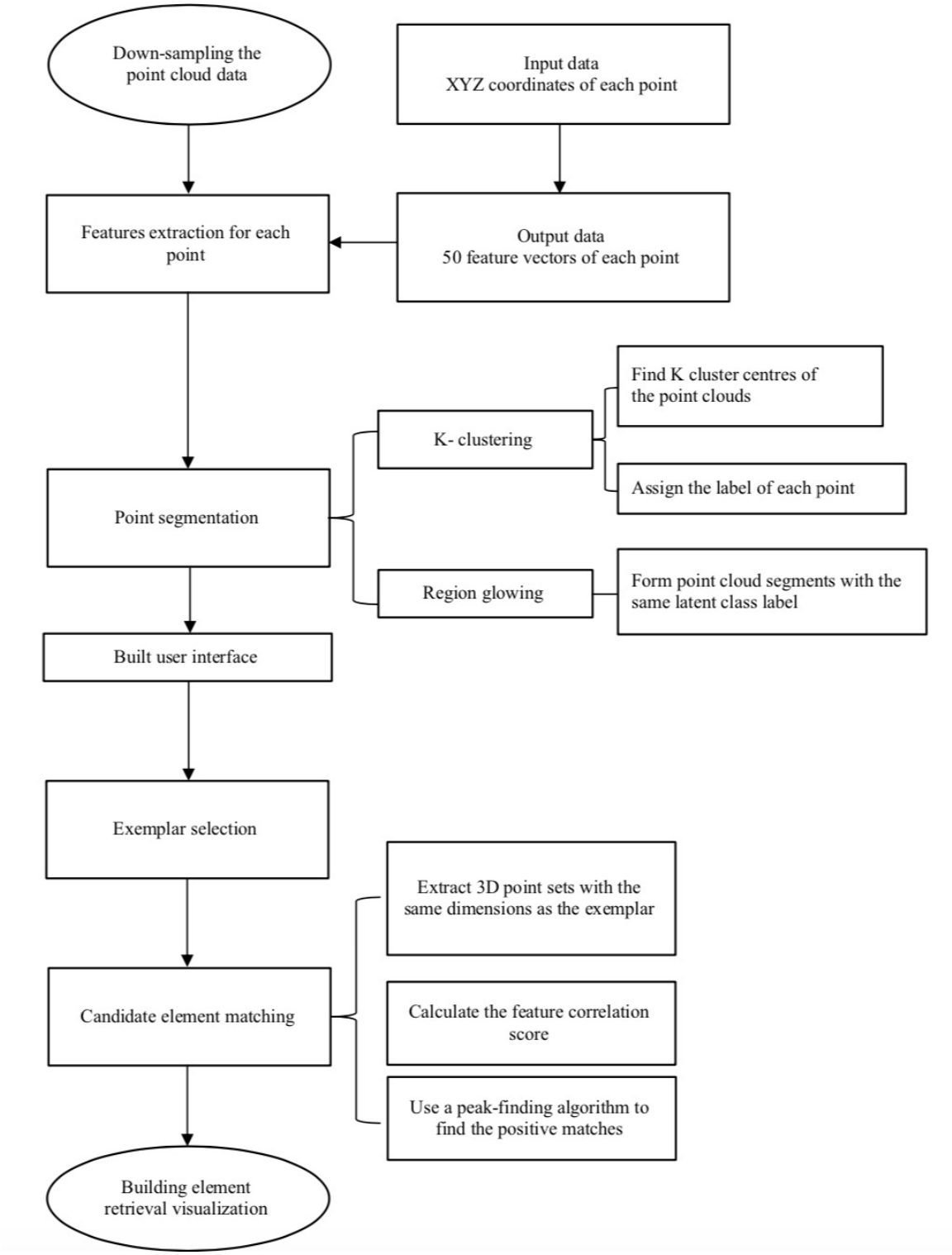


Figure 2.1: Building element retrieval pipeline

The proposed object retrieval method aims to work in the scenario of an unknown building point cloud where the target objects could include building elements of arbitrary shapes as well as temporary structures and equipment. In this scenario, it is infeasible to train a network to recognize every type of object since the number of possible objects would be too large. Thus, the concept of one-shot learning is used to learn features in a data-driven manner from an auxiliary dataset and apply it to the actual object retrieval task in a new dataset given only a single example. One-shot learning relies on prior knowledge of previously learnt objects and reusing model parameters in order to be able to recognize new sets of objects with minimal training examples. A pre-trained deep network is applied to the unknown point cloud to compute features that can be used for object retrieval. Even though the deep network is trained on point cloud data that is likely to contain different objects from the point cloud data in which it is applied, the learned features can still generalize to the new environment as long as the training dataset is sufficiently diverse and similar to objects in the new environment [52].

The auxiliary dataset of similar and dissimilar pairs is created by converting a dataset of pre-made BIM models into synthetic point clouds. The dataset contains 7 different building models with common building elements such as beams, columns, walls, pipes, doors, windows, and railings. After conversion into point clouds, each building contains an average of 1.1 million points and around 4000 building elements. Figure 2.2 shows some examples of the auxiliary dataset where the left column is the mesh model extracted from BIM and the right column is the synthetic point cloud. Each point in the synthetic point cloud will have an object ID corresponding to its parent object in the BIM model. Similar and dissimilar pairs of points are defined by whether the points have the same object ID. For illustration purposes, the point clouds in Figure 2.2 are color-coded by object ID. Thus points with the same color are defined as similar whereas points with different colors are defined as dissimilar. Figure 2.3 shows the architecture of the deep neural network used in this study

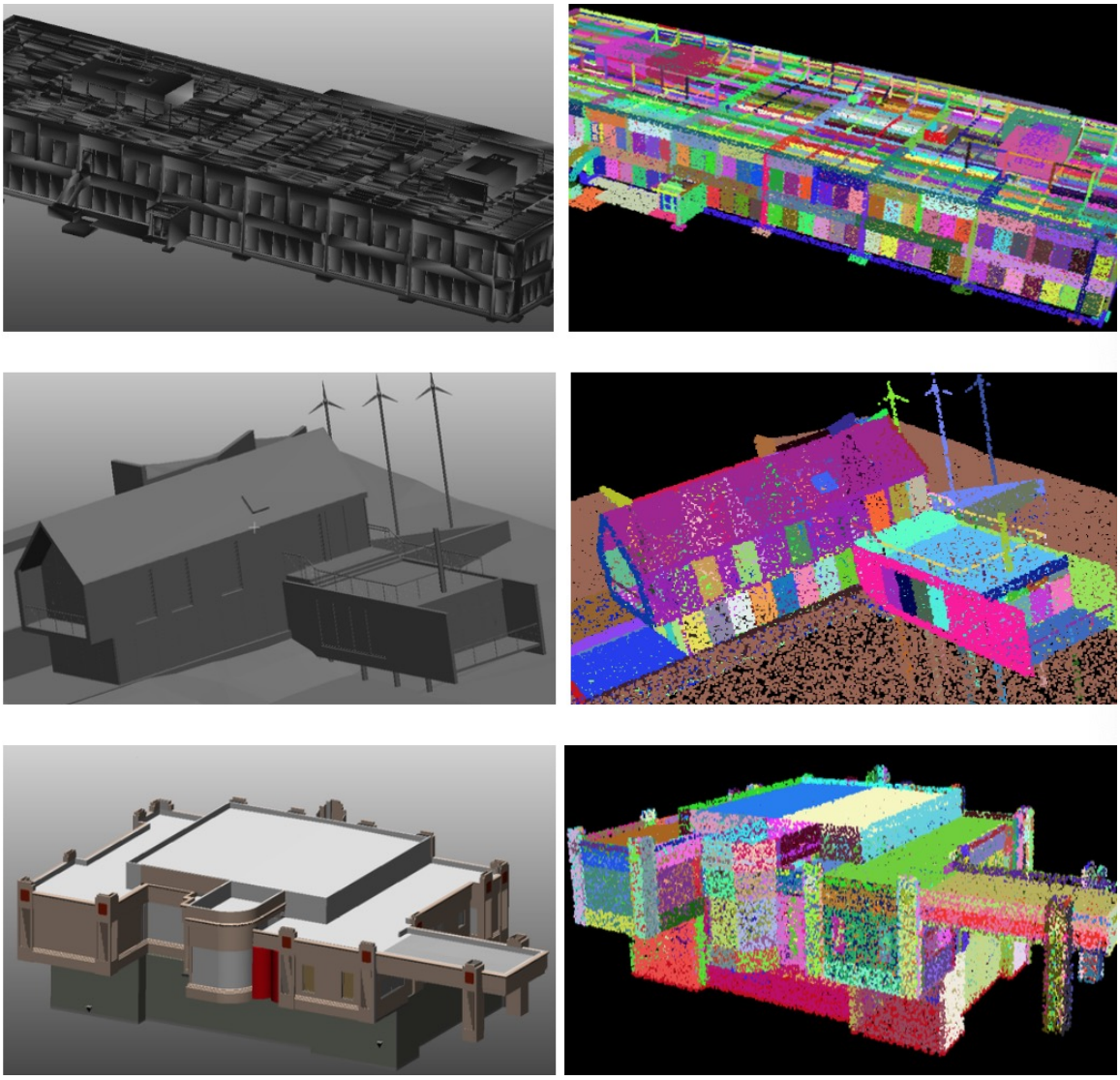


Figure 2.2: Auxiliary dataset of synthetic point clouds generated from BIM models

to compute point-level features. There are pre-existing networks such as Pointnet [53] and SGPN [54] for point cloud processing but these are mostly used for classification and not retrieval. The proposed network uses similar design choices such as convolution and pooling layers but adds a normalized feature embedding layer that is more suitable for metric learning [47]. The network takes in input point cloud data as an Nx3 matrix, where N is the number of points and outputs an Nx50 matrix which contains a 50-dimensional feature vector for each point. A multi-resolution pooling scheme is used where neighboring points

at three different resolutions, r1=0.2m, r2=0.4m, r3=0.6m, are pooled together to compute the feature vector at the center point.

The network is pre-trained using the triplet loss function which has been shown to demonstrate good performance in image retrieval [47]. The triplet loss function works by collecting triples of points, each containing a similar pair and a dissimilar pair, and updates the network such that it projects the similar pair closer together in feature space and the dissimilar pair farther apart in feature space. The triplet loss margin, which controls the difference in distance in embedding space between similar pairs and dissimilar pairs, is set to 1.0 as prescribed in [46]. During the training process, point cloud data is passed to the deep neural network in batches of 256 points. The batches are formed by randomly extracting 5m x 5m x 5m boxes from the training point cloud scene and then randomly sampling multiple points from each object instance in the box until 256 points are obtained. A batch of points will thus contain some pairs of points that originate from the same object and some pairs of points that originate from different objects. The network is then trained by computing the triplet loss based on that batch of points and updating the corresponding feature space. For example, features from two points on the same window become more similar whereas features from a window point and a wall point become less similar.

2.2 Point Cloud Segmentation

After computing point-level features with the pre-trained network, the next step in processing the point cloud data is to group neighboring points together into cohesive segments which have object-level semantics. This process is not strictly necessary for object retrieval in this study since the feature comparison is performed at the point level, but having a pre-segmented point cloud makes it easier for users to select objects in the user interface and also makes the point cloud scene easier to visualize. Clustering is carried out once in feature space and then in geometric space. The feature vectors are first assigned to clusters in fea-

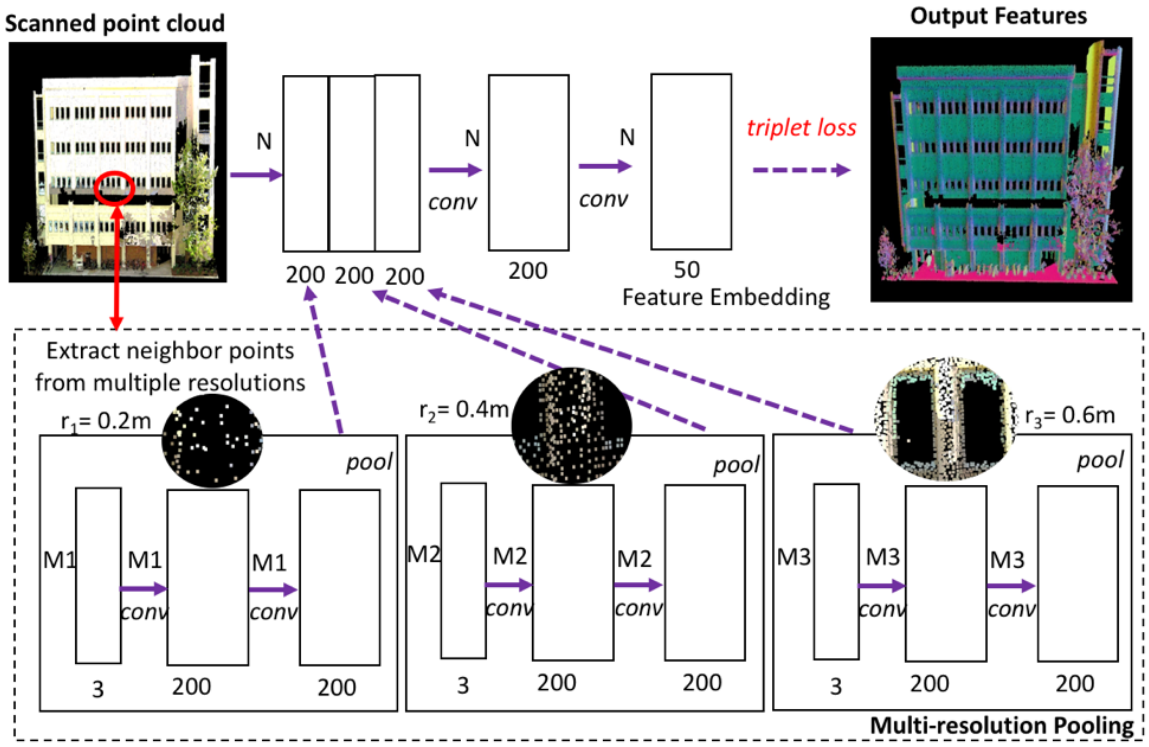
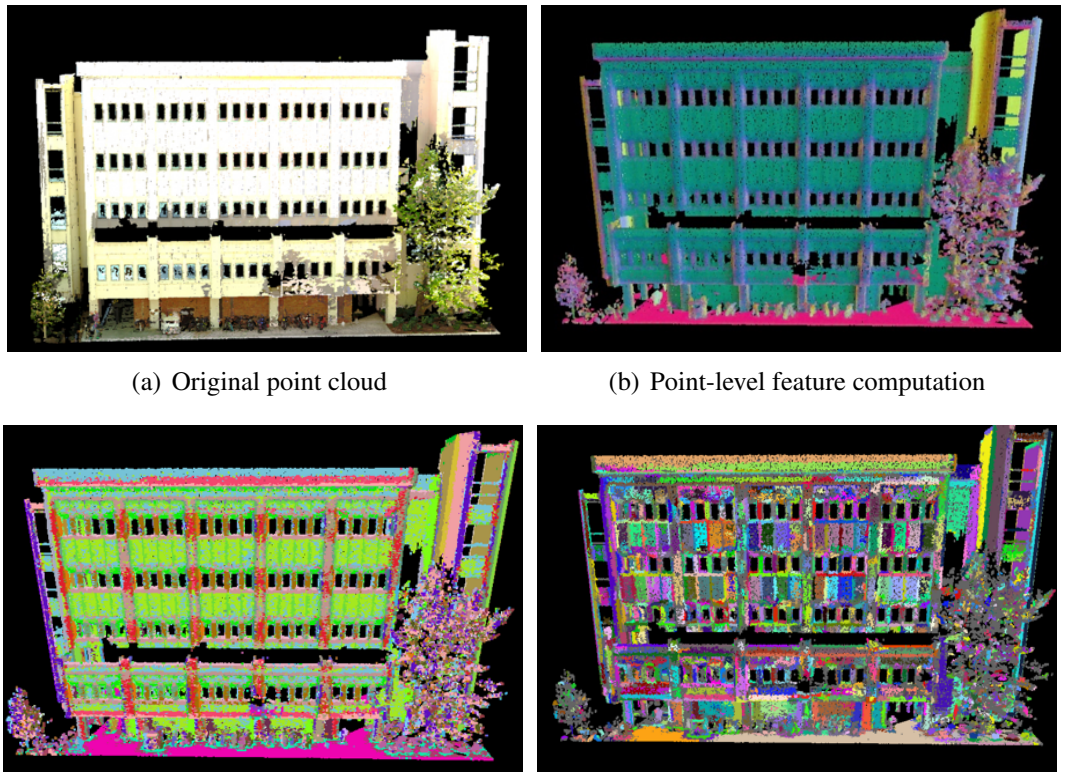


Figure 2.3: Deep neural network architecture used to compute point-level features

ture space using K-means clustering. Then, Euclidean clustering is used to incrementally merge neighboring points in geometric space with feature vectors that belong in the same cluster. This process is also known as region growing. Figure 2.4 shows a visualization of the resulting point cloud after each intermediate step. Figure 2.4(b) shows the results of point-level feature computation, where each point is color-coded according to its semantic feature vector. For example, wall points are mostly green, ground points are mostly red, whereas tree points are mostly purple. Figure 2.4(c) has a wider palette of colors because feature clustering more finely subdivides points in feature space into different groups. Finally, Figure 3.11(d) shows the results after Euclidean clustering, where neighboring points in space are merged together into segments. In Figure 3.11(d), each individual segment in geometric space is visualized in a different color.



(a) Original point cloud (b) Point-level feature computation

(c) Feature clustering (d) Euclidean clustering

Figure 2.4: Building point cloud after each intermediate step

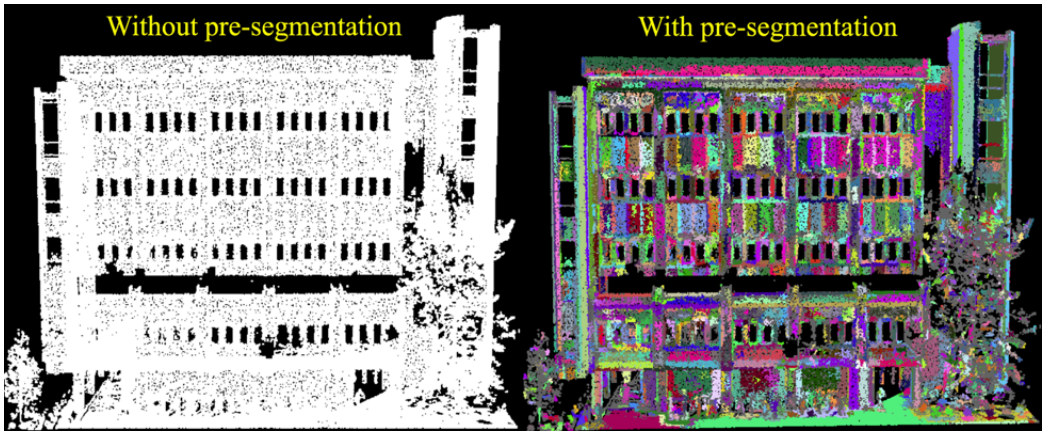
2.3 Exemplar Selection Interface

Next, the user has to select an exemplar to serve as the query object from which similar object instances can be automatically retrieved. Several design considerations have to be taken into account when designing a user interface for the proposed object retrieval system. The user interface has to be simple, fast, responsive, and intuitive. To achieve this purpose, the user interface is implemented with multiple enhancements to improve usability. Pre-segmentation of the point cloud (Figure 2.5(a)) is carried out to make sure that object boundaries are visually prominent (i.e. appear as different colors) to the user before selection. Without pre-segmentation, all the points exist in a uniform color and it is difficult for the user to visually identify an object. Automated background filtering (Figure 2.5(b)) is also implemented so that background points are not selected by accident when the user draws a region of interest, which makes it easier to isolate a foreground object. Finally, the

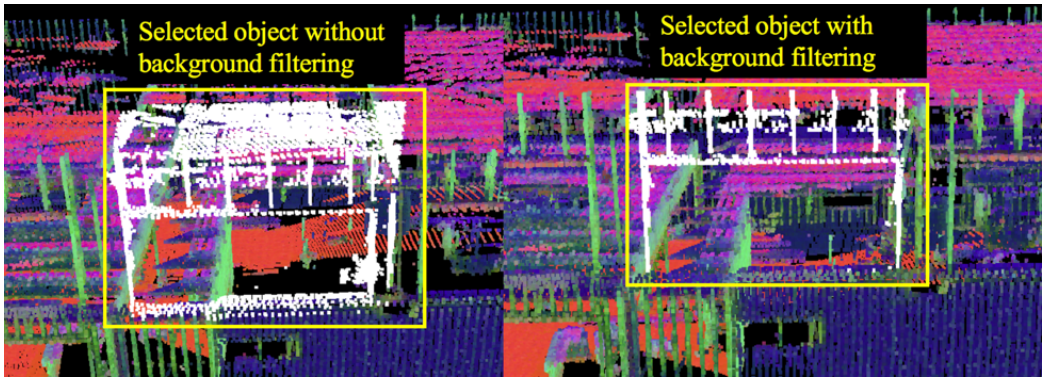
lasso selection tool is used so that objects with arbitrary geometry (e.g., curved or other non-rectangular shapes) can be selected. Figure 2.5(c) shows an example where the user is able to move the cursor to delineate the boundary of a non-rectangular shape. Figure 2.6 shows that the pre-segmentation process is indispensable and imperative for selecting the suitable candidate retrieval exemplar element. Without the pre-segmentation step, the light features in dataset #1 is too ambiguous to identify the exact light features location.

2.4 Candidate Building Element Retrieval with Feature Matching

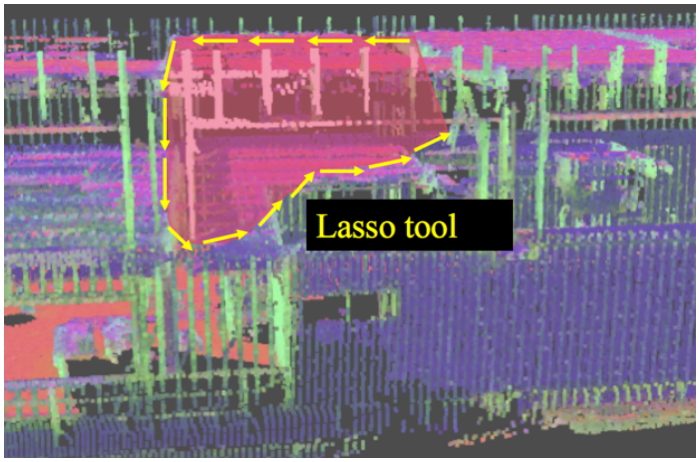
The following step in building element retrieval is to find candidate elements in the global point cloud with positive feature matches to the selected exemplar. As a pre-processing step, a voxel grid data structure is created to represent the point cloud scene. A voxel grid is the three-dimensional analog of image pixels in two-dimensional space and is a way to partition a large point cloud into evenly-spaced bins. The voxel grid representation equalizes the resolution throughout the entire point cloud and helps account for the non-uniform sampling of a target surface during the laser scanning process. The voxel grid is also used as a fast lookup table to find points that are at a specific offset from a given point. Candidates are first extracted by finding points with the same Z-coordinate as a randomly selected seed point in the exemplar. Since most building elements are organized according to levels of a building, one simplification step is to only retrieve candidates with the same Z-coordinate (same floor level). The limitation of this assumption is that building elements that are present at multiple levels (e.g. windows) cannot be retrieved with a single search. To retrieve elements at multiple levels of a building, the user would have to select multiple exemplars, one for each level of the building. Due to the reduced search space, it is still more efficient for the user to select multiple exemplars (one for each level) and obtain the results in a few seconds than to select a single exemplar for entire levels and wait a few minutes to obtain the results. This study also uses another simplifying assumption that the candidates are all rigid translations of the exemplar. Thus, XYZ-offsets are computed for



(a) pre-segmentation



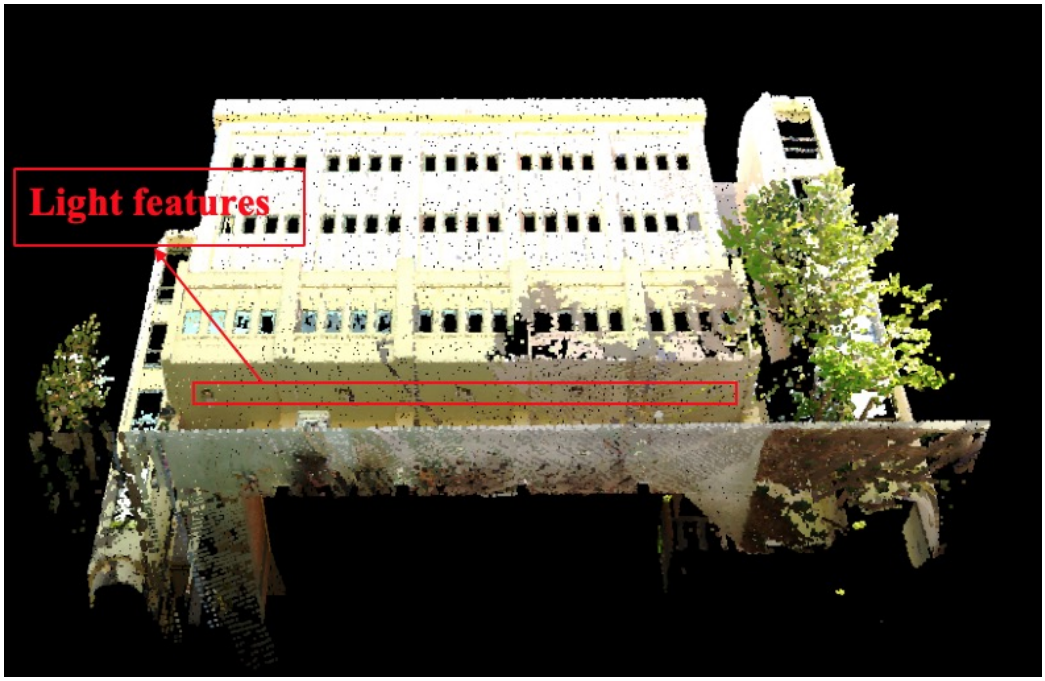
(b) background filtering



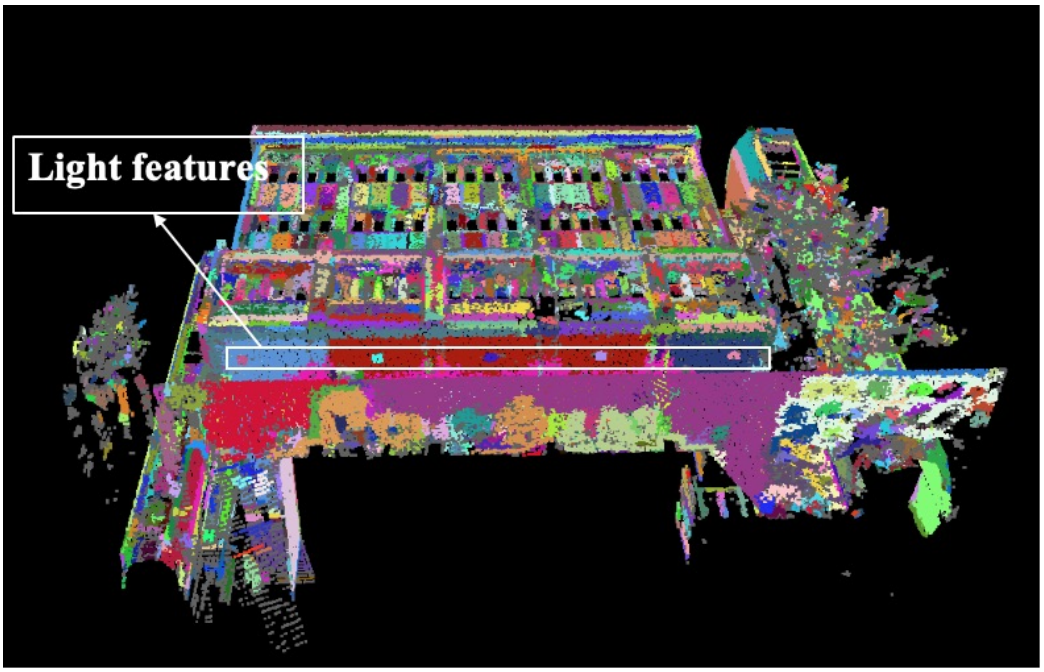
(c) lasso selection

Figure 2.5: Exemplar selection interface features

each other point in the exemplar relative to the seed point and applied to each candidate point to determine matching point sets. For each matching point set, a correlation score is computed to determine its similarity with the exemplar. The closer the similarity between



(a) Original point cloud



(b) Pre-segmentation

Figure 2.6: Exemplar selection interface features - Light features in Dataset #1

two building elements, the higher the candidate correlation score. The candidate-finding correlation score is calculated as shown in Equation 2.1 - 2.4. Three variations of com-

puting the correlation score is considered in this study: (i) direct, (ii) normal, and (iii) feature-based. If the point p_k^i exists in the voxel grid, its feature vector is compared to the corresponding point in the exemplar (or normal vector if the normal method is used), and the error is added to the cumulative score. If the point p_k^i does not exist in the voxel grid (e.g. due to occlusion), then there is no information available for comparison with the exemplar. Instead, a constant error term is added, either μ_f or μ_n. The error term μ_f depends on the triplet loss margin between similar and dissimilar pairs of point features, but may vary because the test data distribution is different from the training data distribution. In this study, the constants are approximated by taking the mean error value of existing normal or feature vectors across the entire test data, resulting in μ_f = 1.32 and μ_n= 0.54. The candidates can be visualized in a top-down view heatmap with XY coordinates in Figure 2.7. Red points indicate areas with high correlation scores, whereas blue points indicate areas with low correlation scores.

$$\text{features=} \begin{cases} C_{ik} = \left\| f(p_k^i) - f(p_k^E) \right\|^2 & \text{if } p_i^k \text{ exist in voxel grid} \\ C_{ik} = \mu_f, & \text{otherwise} \end{cases} \tag{2.1}$$

$$\text{normal=} \begin{cases} C_{ik} = \left\| n(p_k^i) - n(p_k^E) \right\|^2 & \text{if } p_i^k \text{ exist in voxel grid} \\ C_{ik} = \mu_n, & \text{otherwise} \end{cases} \tag{2.2}$$

$$\text{direct=} \begin{cases} C_{ik} = 0 & \text{if } p_i^k \text{ exist in voxel grid} \\ C_{ik} = 1, & \text{otherwise} \end{cases} \tag{2.3}$$

$$C_i = -\sum_k C_{ik} \tag{2.4}$$

where:

C_i - computed scores with respect to the exemplar element;

f - the point feature vector;

n - the point normal vector;

p^i, p^E - points on the candidate element and points on the exemplar element.

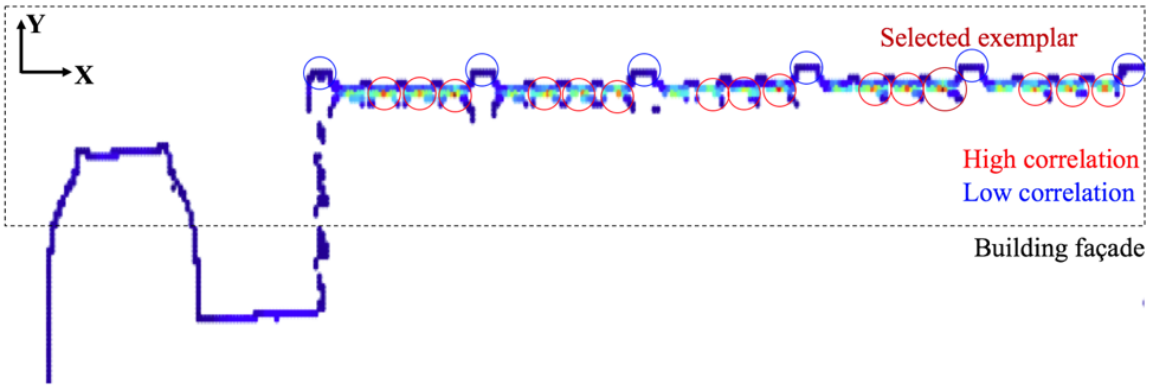


Figure 2.7: Heatmap of feature correlation scores of points along the building façade

2.5 Match Refinement

The final step is to perform match refinement where matching candidates are kept whereas non-matching candidates are filtered out. After the correlation scores are computed, a non-maximal suppression algorithm can be used to find the peaks which are defined as a locally-maximal value of correlation for a candidate element compared to its neighboring elements. Then, a K-means clustering algorithm is used to group the peaks into 3 groups: (i) strong matches, (ii) weak matches, and (iii) non-matches, as shown in Figure 2.8. Strong matches indicate detected building elements that have high similarity to the query element. Whereas weak matches indicate detected elements that are similar to the query element but differ slightly in point cloud geometry due to incomplete data, occlusion and other factors. Non-matches are the background elements that do not match the query element. In this study, only the strong matches are used for computing the accuracy, but the weak matches can also be optionally displayed to the user to indicate more potential matches. Finally, 3D bounding boxes are drawn around the positive matches and displayed through the user interface (Figure 2.9). The total number of positive matches is also shown in the user interface.

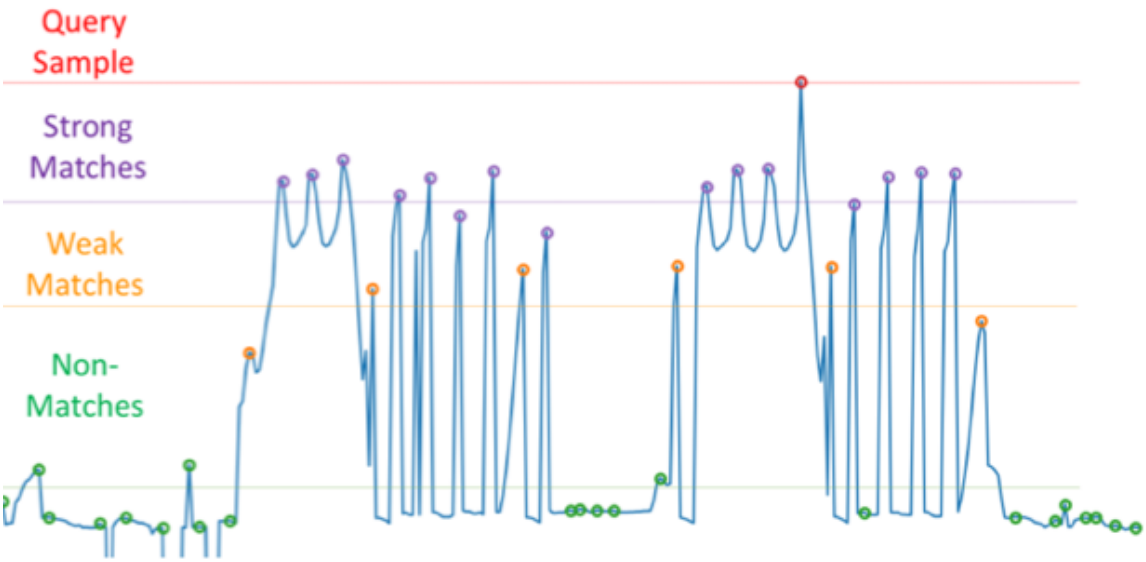


Figure 2.8: Match refinement with K-means clustering

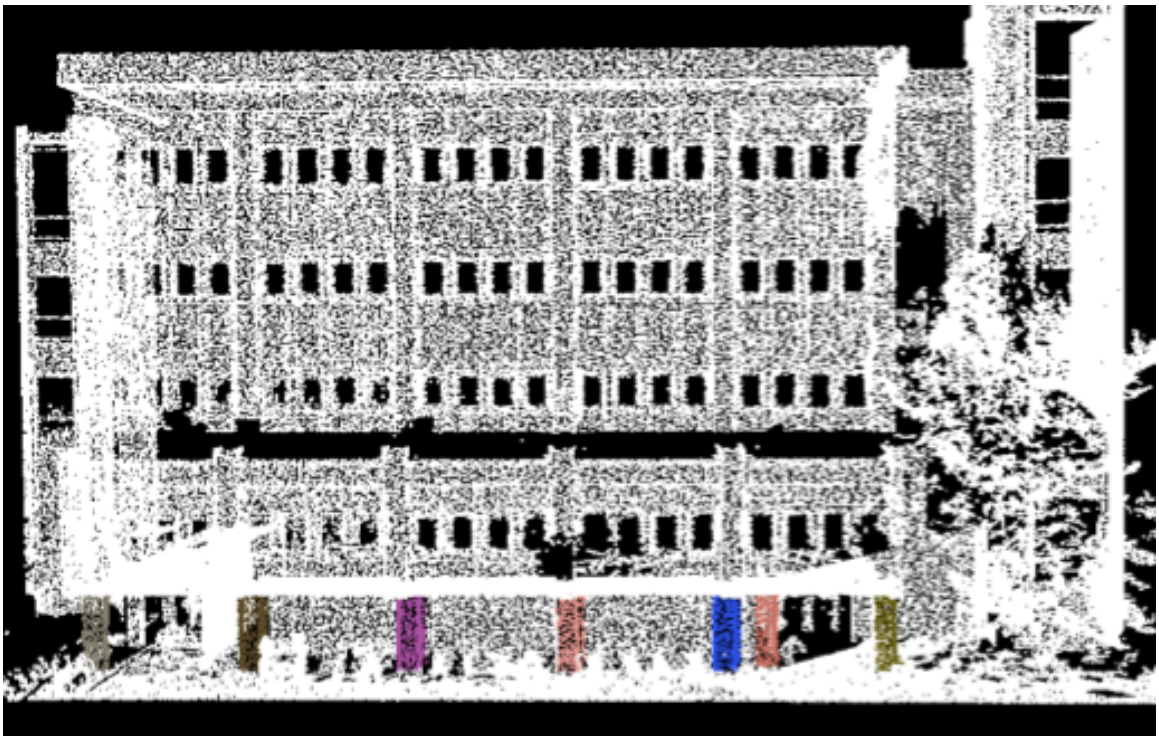


Figure 2.9: Retrieval results visualization

CHAPTER 3

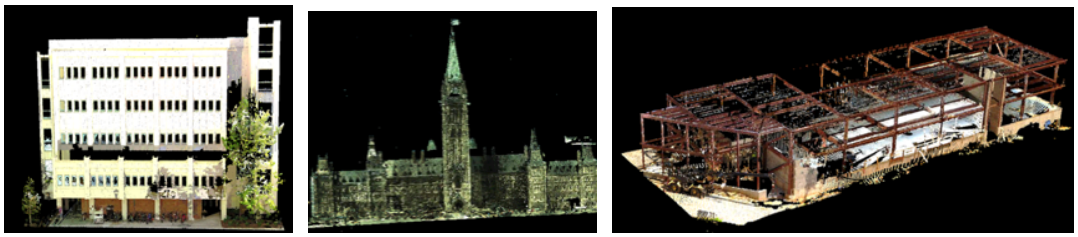
RESULTS

3.1 Point Cloud Datasets

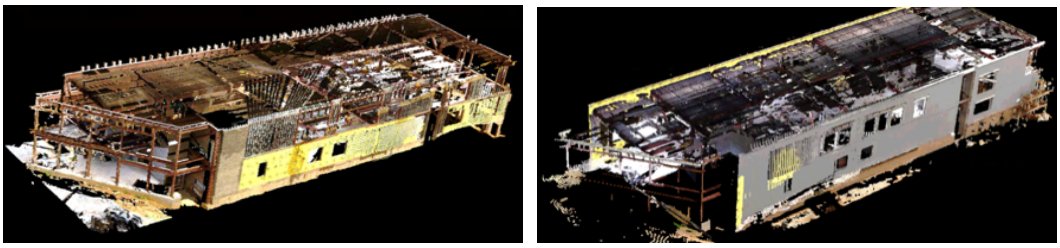
The building element retrieval performance was thoroughly evaluated on five different laser-scanned point clouds taken from three buildings with varying size and architecture. The point clouds were acquired using Terrestrial Laser Scanners (TLS) in E57 format. As shown in Figure 3.1, Dataset #1 is from a modern five-story classroom building at Georgia Institute of Technology (building1) which spans an area of 35m x 16m x 28m, whereas Dataset #2 is from a historical building with Gothic architectural elements (building2) which covers an area of 159m x 51m x 98m. Datasets #3, #4, #5 are from a modern three-story office building (building3) under different phases of construction separated by time intervals of one month which covers an area of 72m x 21m x 16m. After pre-processing steps such as registration, down-sampling, and cropping, Dataset #1 contains 188637 points, Dataset #2 contains 1274517 points, whereas Datasets#3, #4, #5 contain 264837, 391995, 443465 points respectively. In this section, the performance of building element retrieval is compared using three approaches (refer Methodology section): (i) direct voxel grid search and matching ($direct$), (ii) normal vector matching ($normal$), and (iii) feature vector matching ($features$). The feature-based approach is the proposed method, whereas the other two are baselines for comparison purposes. The following paragraphs show the detailed results and discussions.

3.2 Building Element Retrieval Accuracy

Tables 3.1 - 3.3 below show the precision and recall results for building element retrieval for each of the test datasets. Each table shows a performance comparison between the "di-



(a) Dataset #1 (b) Dataset #2 (c) Dataset #3



(d) Dataset #4 (e) Dataset #5

Figure 3.1: Point cloud datasets #1-5 acquired with terrestrial laser scanning

rect", "normal", and "feature" matching methods for different types of building elements. In addition, Figures 3.2-3.5 show a visualization of the retrieval results for each of the test datasets. The ground truth data is labeled based on the image or point cloud data detection of each building as shown in Figure 3.7. For the building 3, the IFC model has provided which could be utilized to validate the retrieval results and an automated Scan-vs-BIM deviation detection[7].

As shown in the Tables 3.1 - 3.3, the precision result of the feature-based matching approach is the best since most of the identified candidates are correct. On the other hand, Table 3.2 shows that the recall result of the direct matching approach is better as most of the elements necessary to be retrieved are correctly identified. In general, the feature-based matching approach achieves the most robust building elements retrieval results as the overall precision and recall rates are both above 90% in all test datasets.

As shown in Figure 3.2, the feature-based matching approach is better able to distinguish the true positive wall segments from similarly-shaped columns or wall segments compared

to the direct matching and normal vector matching approach. The direct and normal matching methods are still able to find candidates with similar shapes but are less able to correctly distinguish between true positives and false positives. Figure 3.3 shows that the direct-based method is hardly to distinguish the small exemplar data (light features) from the similar geometric shape object when compared to the other two methods. This is because the direct approach does not consider to comparison much more features of each point besides the shape boundary. Figure 3.4 shows that the feature-based matching approach can identify complex window elements from a historical building correctly compared to the other two methods, which mistakenly retrieved smaller or differently-shaped windows. This is because it can extract more discriminative and unique features among the points through a deep network, rather than only considering the coarse geometric shape or normal vector characteristics of each point.

Figures 3.5 and 3.6 show that the proposed method can also be used to identify unknown objects or temporary structures for a building during the construction phase. Using the proposed method, temporary structures, materials, and equipment can be tracked and modeled in the as-built BIM model, even though there is no information in the BIM model during the design phase for these temporary objects. Query results in Figure 3.6 show that it is challenging to retrieve all true positive ladders from the point cloud data, because the orientation, posture of placement and point cloud completeness for each ladder instance is different, which makes other ladders difficult to match to the exemplar. However, the proposed method is still able to identify several ladders correctly.

3.3 Study of variance depending on user selection

Since this proposed method is a semi-automated approach based on a selected exemplar in the user interface, selected points for the same object could differ based on how the user draws the object boundaries on the user interface. To measure the effect of the variability

Table 3.1: Precision and recall comparison of dataset #1 elements

Building elements	Numbers of instance	Precision		
		Direct	Normal	Features
Windows	80	75%	99%	100%
Doors	4	100%	100%	100%
Columns	6	50%	71%	83%
Light features	5	83%	100%	100%
Wall segments	40	75%	93%	100%
Overall	135	77%	93%	**97%**
Building elements	Numbers of instance	Recall		
		Direct	Normal	Features
Windows	80	96%	90%	94%
Doors	4	75%	75%	75%
Columns	6	83%	83%	83%
Light features	5	100%	100%	100%
Wall segments	40	100%	100%	100%
Overall	135	**91%**	90%	90%

Table 3.2: Precision and recall comparison of dataset #2 elements

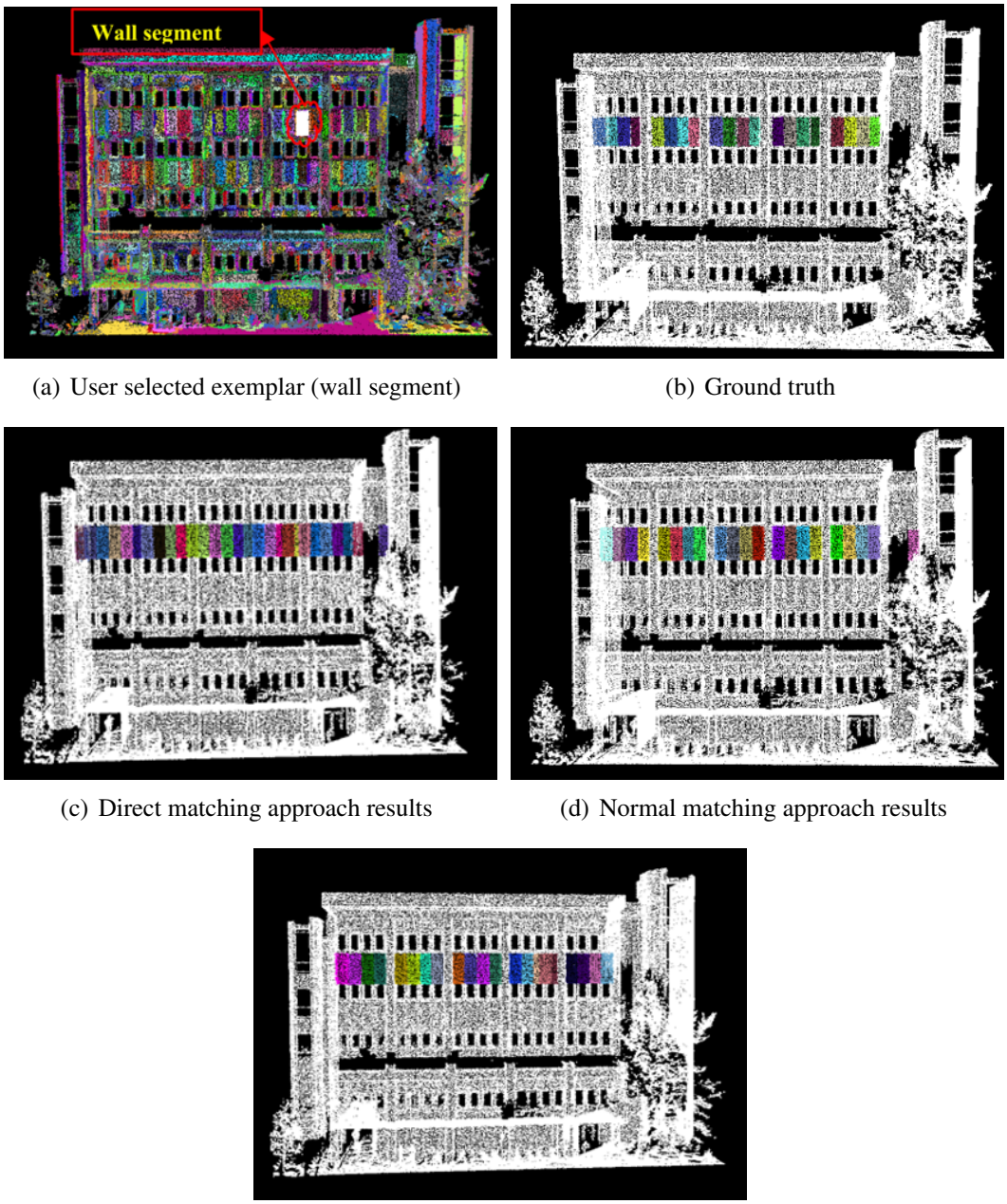
Building elements	Numbers of instance	Precision		
		Direct	Normal	Features
Double windows	24	65%	71%	67%
Triple windows	4	80%	24%	100%
Chimney	6	100%	100%	100%
Roof windows	26	100%	100%	100%
Arch windows	30	90%	100%	100%
Overall	90	87%	79%	**93%**
Building elements	Numbers of instance	Recall		
		Direct	Normal	Features
Double windows	24	100%	100%	100%
Triple windows	4	100%	100%	100%
Chimney	6	100%	83%	83%
Roof windows	26	92%	92%	88%
Arch windows	30	90%	90%	90%
Overall	90	**96%**	93%	92%

Table 3.3: Precision and recall comparison of dataset #3-5 elements

Building elements		Numbers of instance	Precision		
			Direct	Normal	Features
Dataset#3	Columns	32	75%	99%	100%
	Beams	15	100%	100%	100%
	Wall segments	5	50%	71%	83%
	Door openings	26	100%	100%	100%
	Slabs	10	75%	93%	100%
Dataset#4	Curtain wall openings	5	45%	71%	83%
	Slabs	10	100%	100%	100%
Dataset#5	Curtain wall openings	5	63%	100%	83%
	Wall segments	4	80%	80%	80%
Overall		88	77%	93%	**97%**
Building elements		Numbers of instance	Recall		
			Direct	Normal	Features
Dataset#3	Columns	32	81%	41%	81%
	Beams	15	87%	80%	87%
	Wall segments	5	100%	100%	100%
	Door openings	26	100%	100%	100%
	Slabs	10	80%	100%	90%
Dataset#4	Curtain wall openings	5	100%	100%	100%
	Slabs	10	70%	70%	70%
Dataset#5	Curtain wall openings	5	100%	80%	100%
	Wall segments	4	100%	100%	100%
Overall		88	90%	84%	**92%**

of user-selected exemplar on the retrieval results, the student's T-Distribution [55] is used in this paper to compute confidence intervals with a small sample size with the Equation 3.1. The retrieval results for window objects (number=80) in Dataset #1 were repeated for 7 trials, and the variability in precision and recall are computed in Table 3.4:

$$\tau = \frac{t_{n-1,a} s_{(n)}}{\sqrt{n}} \tag{3.1}$$



(a) User selected exemplar (wall segment)

(b) Ground truth

(c) Direct matching approach results

(d) Normal matching approach results

(e) Feature matching approach results

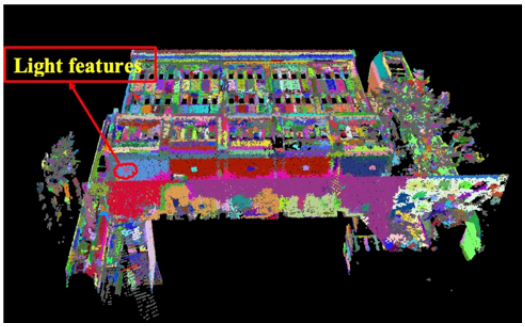
Figure 3.2: Matching method comparison with dataset #1 wall segments

where:

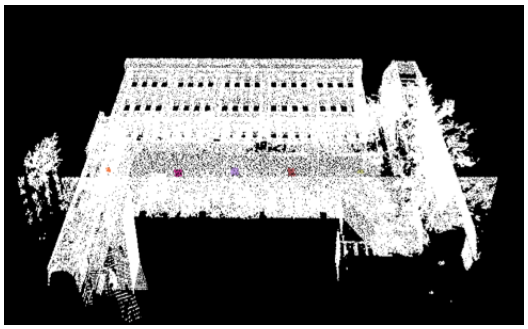
τ - Half interval;

$s_{(n)}$ - Sample variance;

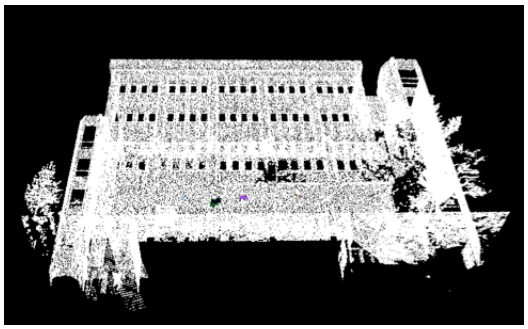
n - Total number of trails;



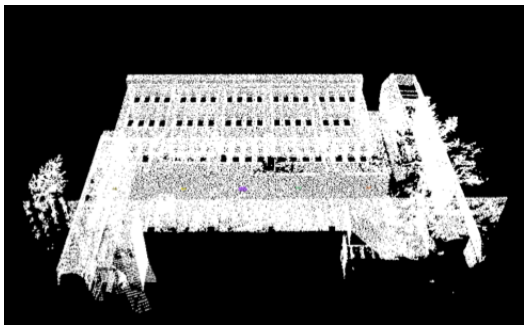
(a) User selected exemplar (Light features)



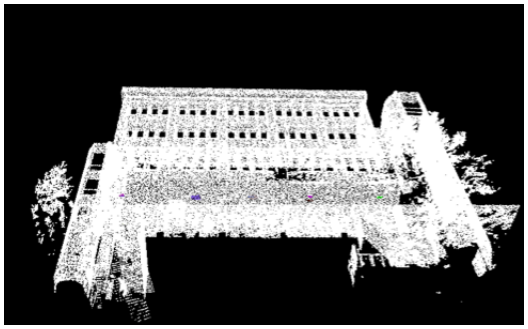
(b) Ground truth



(c) Direct matching approach results



(d) Normal matching approach results



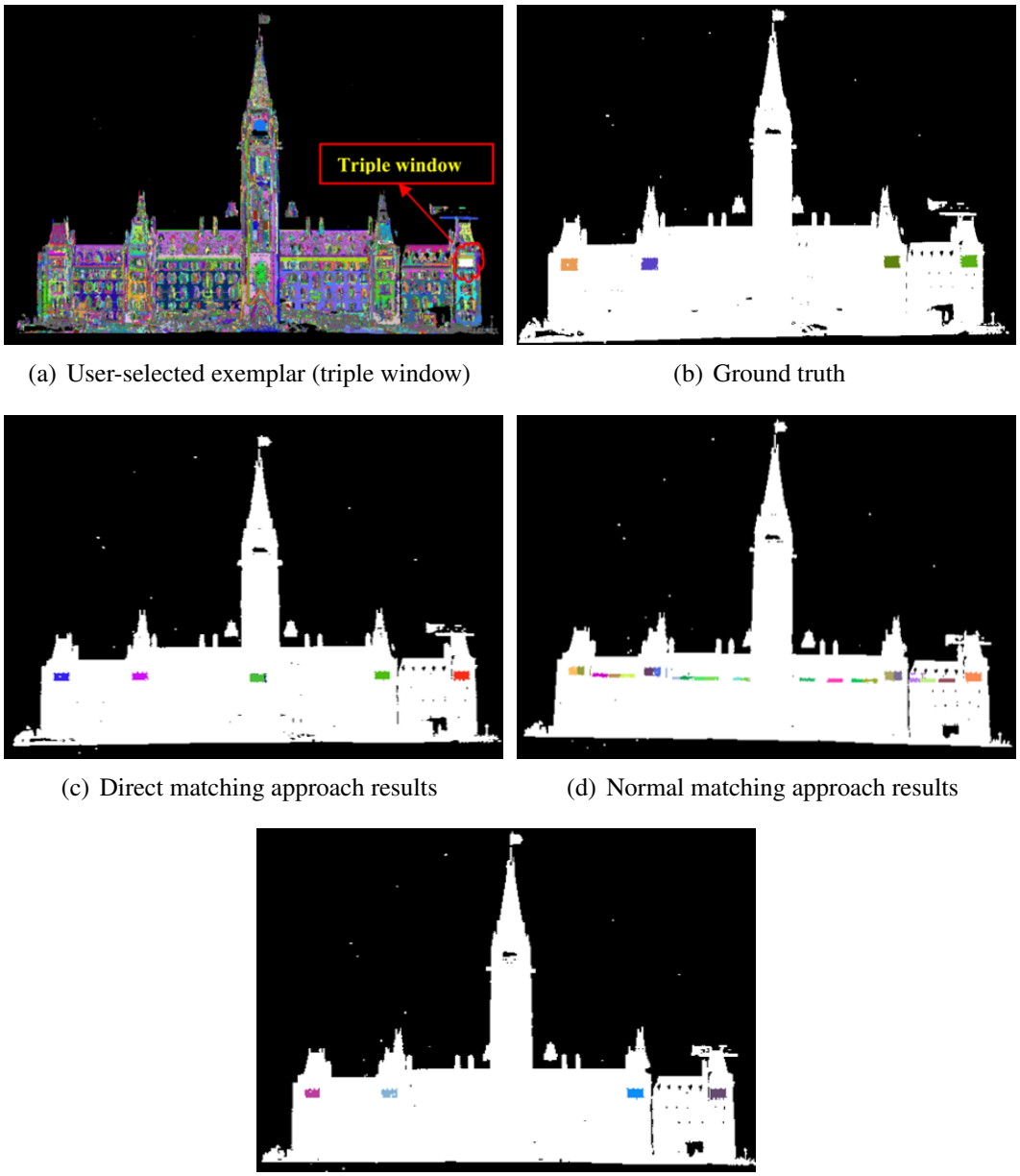
(e) Feature matching approach results

Figure 3.3: Matching method comparison with dataset #1 light features

Table 3.4: Study of variance due to user selection

	Precision	Recall
Sample mean	98.22%	89.64%
Sample Variance	0.00042	0.00358
Half interval	0.01505	0.4394
Overall(at 90% confidence interval)	$98.22\% \pm 1.51\%$	$89.64\% \pm 4.4\%$

$t_{n-1,a}$ - $(p_{n-1} > t_{n-1,a} = a)$ with $n-1$ degrees of freedom.



(a) User-selected exemplar (triple window)

(b) Ground truth

(c) Direct matching approach results

(d) Normal matching approach results

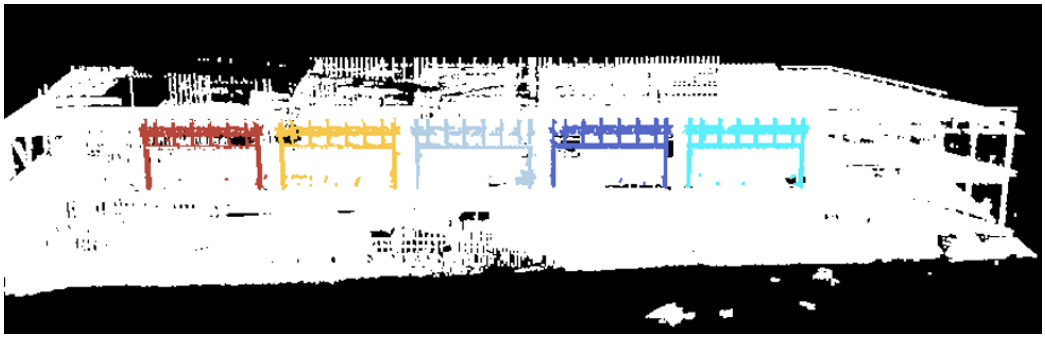
(e) Feature matching approach results

Figure 3.4: Matching method comparison with dataset #2 triple windows

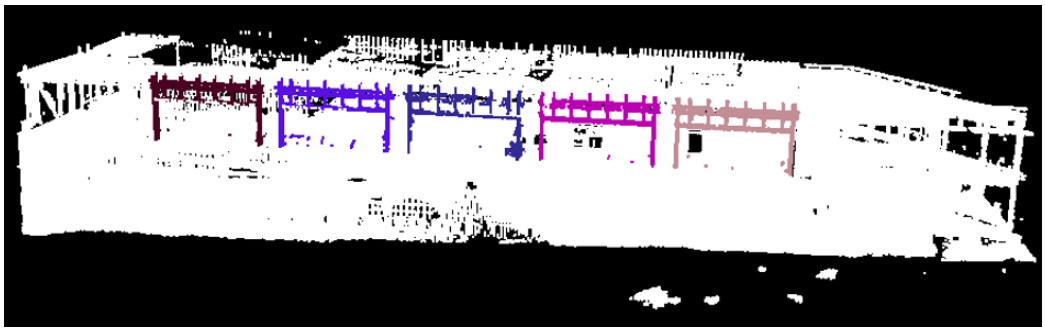
3.4 Study of the effect of voxel grid resolution

In the pre-processing step, a voxel grid data structure is created to represent the point cloud scene (refer Methodology section). In this study, the size of each voxel defined as the voxel resolution is varied between 0.05m, 0.1m, 0.2m, 0.3m, and 0.5m. The Figure 3.8 - Figure 3.10 shows the visualization results of each method in dataset #2 with the double windows

(a) User selected exemplar (steel structure frame)



(b) Ground truth



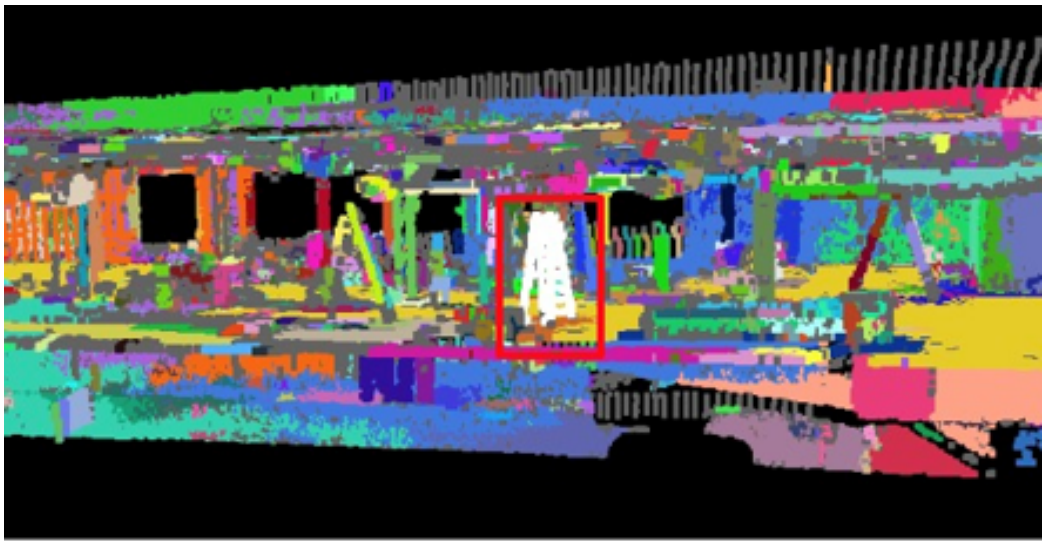
(c) Candidate matching results

(d) Query results in isolation to the rest of the point cloud

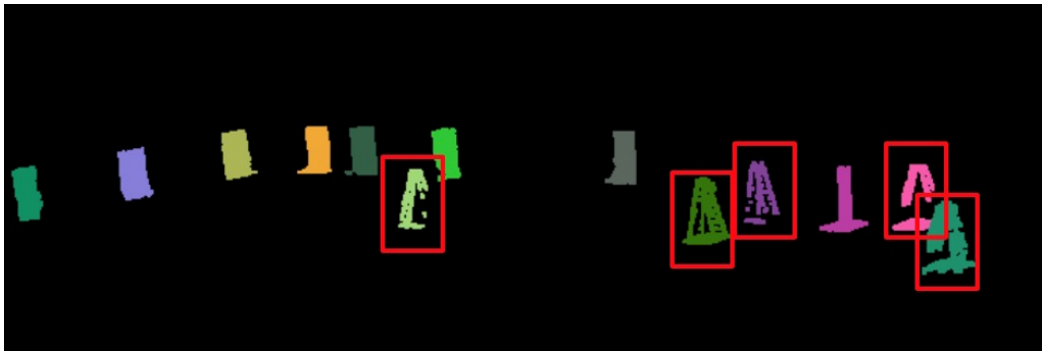
Figure 3.5: Steel structure frame retrieval results with dataset #4

instances. The total numbers of this double windows in ground truth is 24. The True Positive (TP) detected elements, False Positive (FP) detected elements and False Negative (FN)

(a) Ladders on-site



(b) User selected exemplar

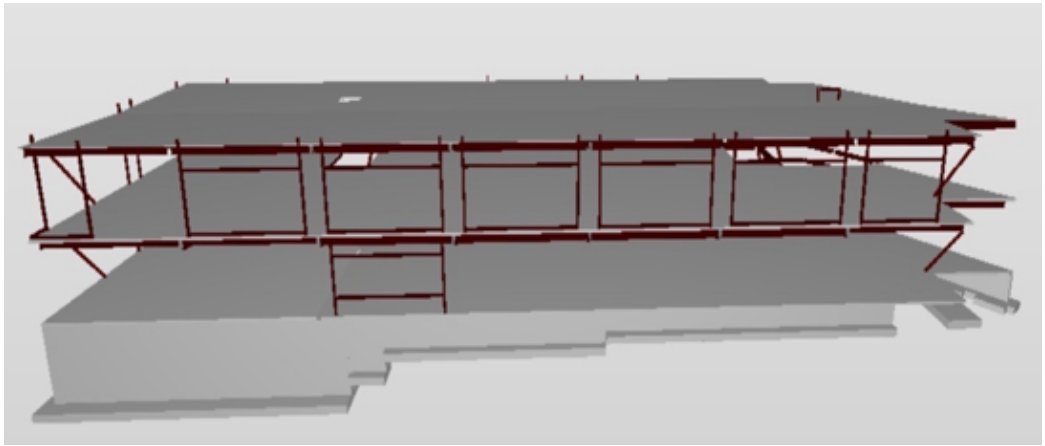


(c) Query results in isolation to the rest of the point cloud (true positive retrieval ladders are framed by red rectangles)

Figure 3.6: Temporary structure elements retrieval results with dataset #5

(a) Image onsite with dataset #4

(b) Image onsite with dataset #5



(c) IFC model for building 3

Figure 3.7: Ground truth data

elements has been counted of each method corresponding to different Voxel Resolution as shown in Table 3.5 - Table 3.7. The resulting precision and recall rates of the three candidate matching methods are compared in Table 3.8 below.

From Table 3.8, the precision shows a decreasing trend with voxel size whereas the recall shows a trend that is first decreasing then increasing with voxel size. This is because when the size of the voxels increases, the fewer the number of the points characteristics or features that are used to find the right candidate elements, thus increasing the number of candidate matches but also increasing the likelihood of causing false positive matches. To optimize the balance between precision and recall, a smaller voxel size should be used.

Table 3.5: Direct method querying results with different Voxel Resolutions

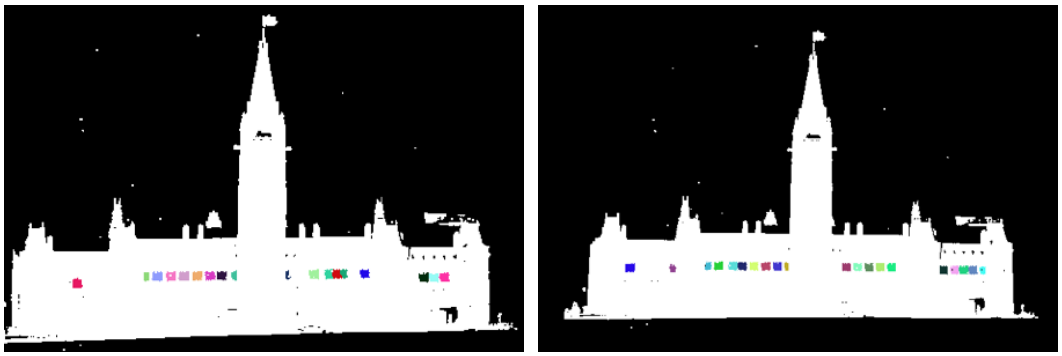
Voxel Resolution (m)	TP	FP	FN
0.05	18	0	6
0.1	18	2	6
0.2	20	4	4
0.3	16	6	8
0.5	24	15	0

Table 3.6: Normal method querying results with different Voxel Resolutions

Voxel Resolution (m)	TP	FP	FN
0.05	24	1	0
0.1	17	1	7
0.2	17	2	7
0.3	10	1	14
0.5	23	8	1

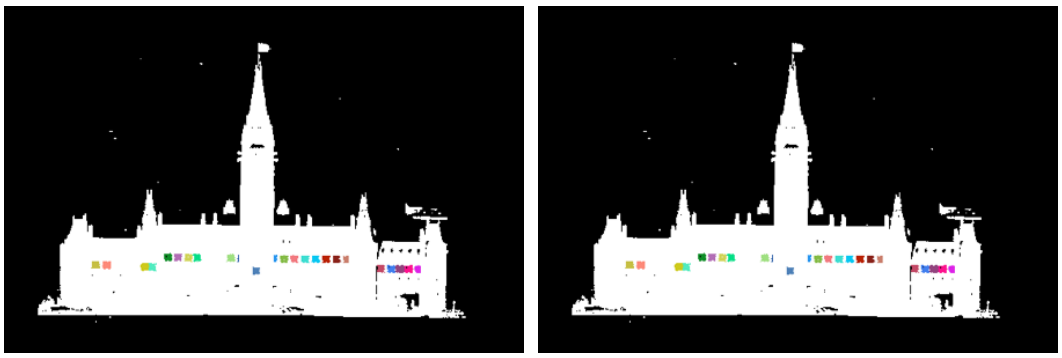
3.5 Study of the effect of the clustering algorithm

The effect of different choices of clustering algorithms in the match refinement step is analyzed by comparing between (i) K-means, (ii) K-means++, (iii) X-means, and (iv) spectral clustering algorithms.



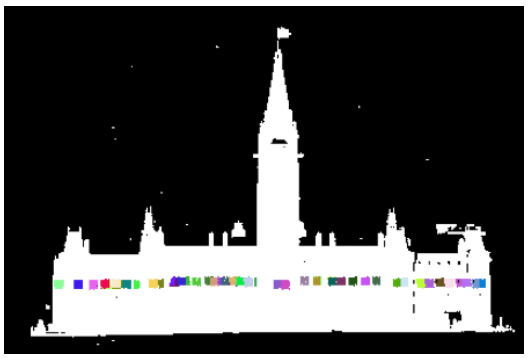
(a) Voxel Resolution(m) = 0.05

(b) Voxel Resolution(m) = 0.1



(c) Voxel Resolution(m) = 0.2

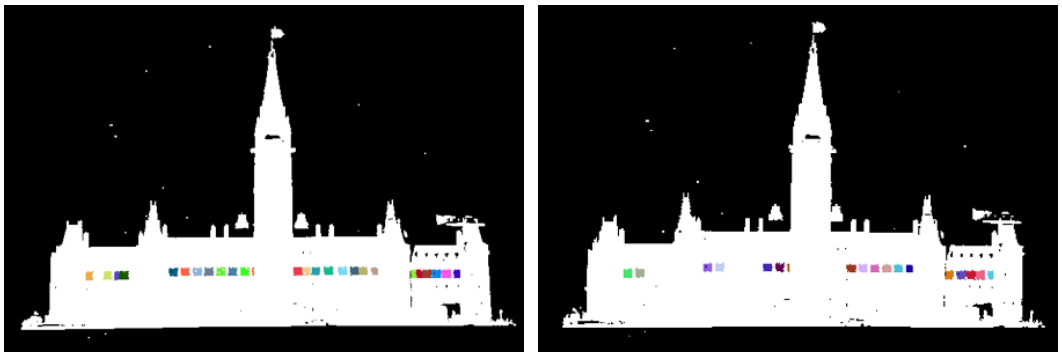
(d) Voxel Resolution(m) = 0.3



(e) Voxel Resolution(m) = 0.5

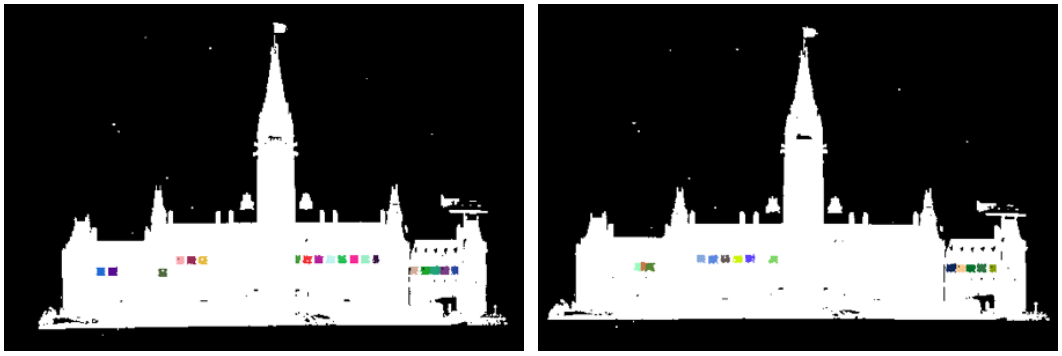
Figure 3.8: Direct method results with different Voxel resolutions - Double windows instances in Dataset #2

In practice Spectral Clustering is very useful when the structure of the individual clusters is highly non-convex or more generally when a measure of the center and spread of the cluster is not a suitable description of the complete cluster. For instance when clusters are nested circles on the 2D plane. The performance is measured based on precision and recall



(a) Voxel Resolution(m) = 0.05

(b) Voxel Resolution(m) = 0.1



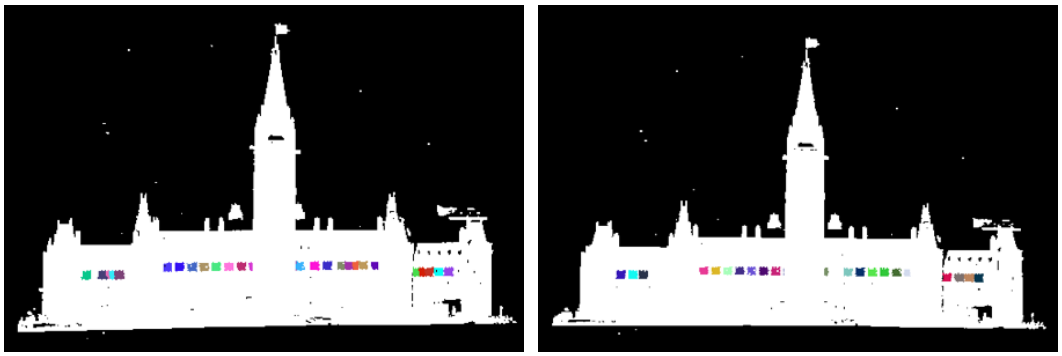
(c) Voxel Resolution(m) = 0.2

(d) Voxel Resolution(m) = 0.3

(e) Voxel Resolution(m) = 0.5

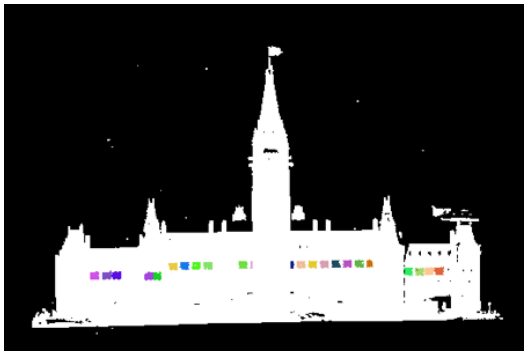
Figure 3.9: Normal method results with different Voxel Resolution - Double windows instances in Dataset #2

rates on wall, window, and column objects in Dataset #1. Results in Table 3.10 show that using different clustering algorithms result in a slight increase or decrease of the precision and recall by around 10%. Overall, the spectral clustering algorithm performed the best in terms of precision whereas the K-means++ algorithm performed the best in terms of recall rates.

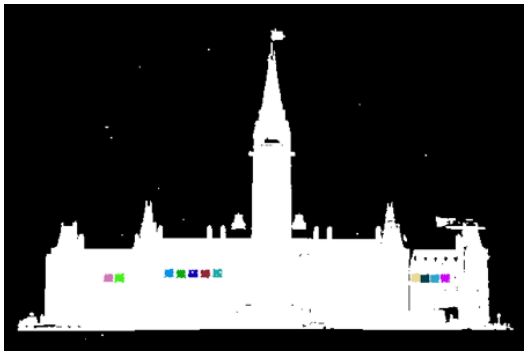


(a) Voxel Resolution(m) = 0.05

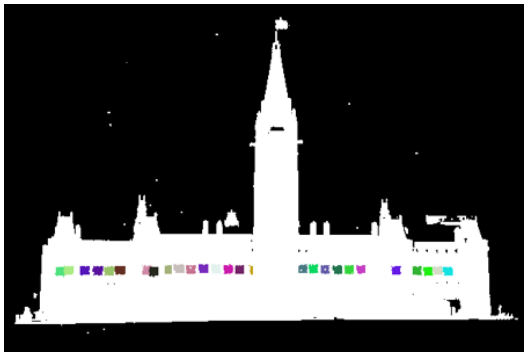
(b) Voxel Resolution(m) = 0.1



(c) Voxel Resolution(m) = 0.2



(d) Voxel Resolution(m) = 0.3



(e) Voxel Resolution(m) = 0.5

Figure 3.10: Features method results with different Voxel Resolution - Double windows instances in Dataset #2

On the other hand, the querying result varies when we set different K values. The column in Dataset #3 has been selected as an instance to conduct the comparison of different retrieval approaches. Figure 3.7 shows the ground truth image of columns in building 3, the total number of columns in building 3 is 32. As shown in Figure 3.12 - Figure 3.14, False Positive detected objects decrease while False Negative ones increase with larger K value

Table 3.7: Features method querying results with different Voxel Resolutions

Voxel Resolution (m)	TP	FP	FN
0.05	23	1	1
0.1	21	1	3
0.2	20	3	4
0.3	10	1	14
0.5	23	6	1

Table 3.8: Precision and recall comparison with different voxel resolutions

Voxel Resolution (m)	Direct		Normal		Features	
	Precision	Recall	Precision	Recall	Precision	Recall
0.05	100%	75%	96%	100%	96%	96%
0.1	90%	75%	94%	71%	95%	88%
0.2	83%	83%	89%	71%	87%	83%
0.3	73%	67%	91%	42%	91%	42%
0.5	62%	100%	74%	96%	79%	96%
Average	82%	80%	89%	76%	**90%**	**81%**

of three approaches. This is because more weak matching objects are recognized as non-matching ones that can filter more False Positives. Although the bigger K can achieve a higher precision, the recall performance is worsened than the smaller one since true candidates in weak matching ranges are unable to be detected. Table 3.9 shows that the proposed feature-based retrieval method works well when we set K value equals to 2, and its precision and recall performance can both maintain a high score.

Table 3.9: Precision and recall rates of different K Values in dataset #3 (Columns)

K values	Approach	TP	FP	FN	Precision	Recall
K =2	Direct	26	4	6	86.7%	81.3%
	Normal	13	1	19	92.9%	40.6%
	Features	26	1	6	**96.3%**	**81.3%**
K =3	Direct	20	2	12	90.9%	62.5%
	Normal	13	1	19	92.9%	40.6%
	Features	8	0	24	100%	25.0%
K =4	Direct	10	1	22	90.9%	31.5%
	Normal	12	1	20	92.3%	37.5%
	Features	6	0	26	100%	18.75%

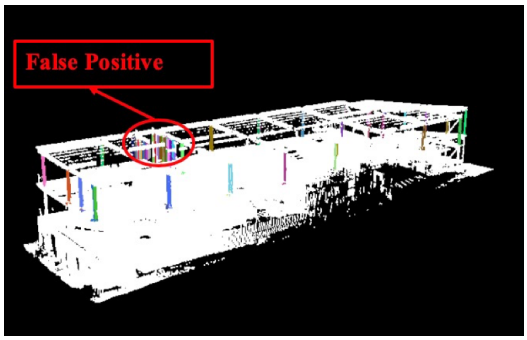
(a) Front

(b) Back

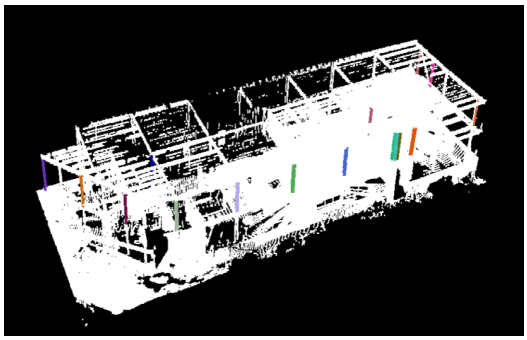
(c) Side 1

(d) Side 2

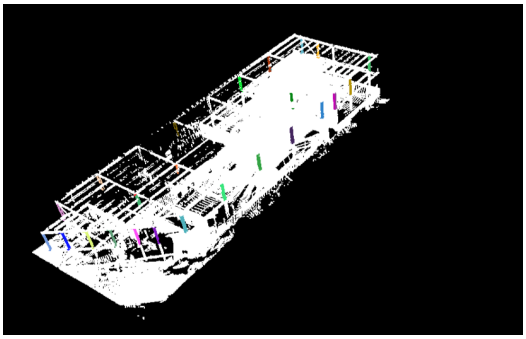
Figure 3.11: Ground truth image of columns in dataset #3



(a) Direct approach

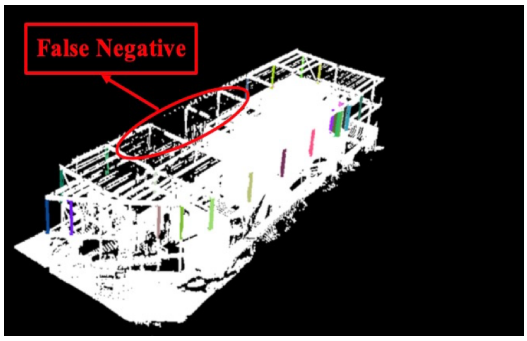


(b) Normal approach

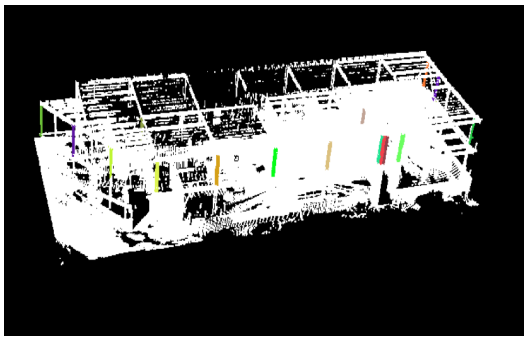


(c) Features approach

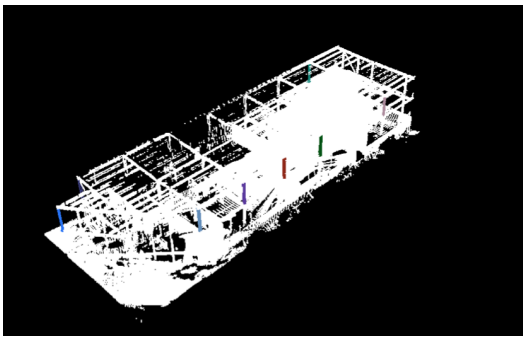
Figure 3.12: Retrieval result with K = 2



(a) Direct approach



(b) Normal approach



(c) Features approach

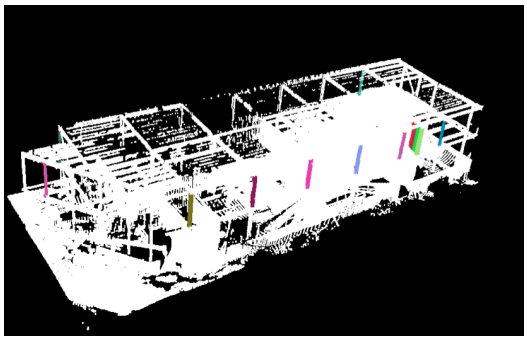
Figure 3.13: Retrieval result with K = 3

Table 3.10: Precision and recall rates of different clustering algorithms

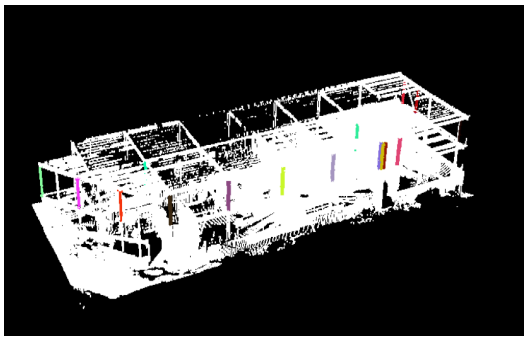
Clustering algorithm	Window Precision	Window Recall	Wall Precision	Wall Recall	Column Precision	Column Recall
K-means	95%	90%	94%	85%	83%	83%
K-means++	95%	90%	94%	85%	100%	83%
X-means	93%	70%	78%	90%	83%	83%
Spectral clustering	100%	65%	100%	80%	83%	83%

3.6 Study of the effect of relevance feedback

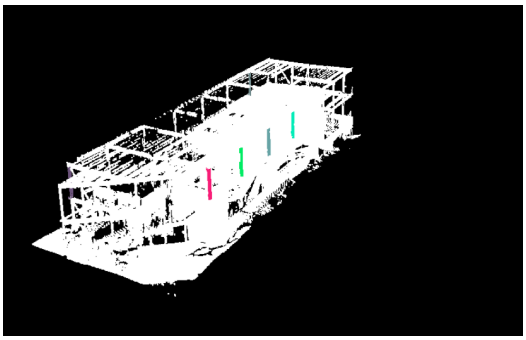
The technique of relevance feedback [56, 57, 58] is used to improve the precision and recall rates by having multiple rounds of user input. After selecting an exemplar object and getting the retrieval results, the user is given the option to provide feedback to the retrieval algorithm in the form of selecting additional exemplar objects or unselecting wrongly retrieved objects. In the case of selecting additional exemplar objects, the correlation scores



(a) Direct approach



(b) Normal approach



(c) Features approach

Figure 3.14: Retrieval result with K = 4

are calculated again relative to the new exemplar and appended to an array of correlation scores. Finally, match refinement is carried out with the updated correlation scores. Results in Table 3.11 show that after each feedback round, the precision and recall rates on wall, window, and column objects in Dataset #1 improved. This suggests that relevance feedback is effective in improving the retrieval results, although at the cost of requiring more manual input from the user.

From Tables 3.12 and 3.13, the feature-based method requires the highest computation time in the initial processing steps, while the normal vector-based and direct voxel grid search methods only require 1/25 1/10 the computation time of the feature-based method. This is because the feature-based method involves a higher amount of computational steps to pass an input point cloud through a deep neural network and compute high-dimensional feature vectors. Figure 3.15 shows that there exists a linear relationship between the num-

ber of points in the input point cloud and the computation time for the initial processing steps, due to having to repeat the initial processing step for each point. On the other hand, Figure 3.16 shows that there is no clear trend for each method in terms of the computation time for element retrieval. As the number of selected exemplar points increases, the computation time of the element retrieval process increases in general but not consistently. This is because each exemplar corresponds to a different number of candidates to match, which will affect the computation time. For example, a window element consists of a small number of points but may have a large number of matches, whereas a floor element consists of a large number of points but may have fewer matches.

Table 3.11: Precision and recall rates after a different number of feedback

Feedback rounds	Window Precision	Window Recall	Wall Precision	Wall Recall	Column Precision	Column Recall
0	95%	90%	94%	85%	83%	83%
1	95%	95%	94%	90%	100%	100%
2	95%	100%	94%	95%	100%	100%
3	100%	65%	100%	95%	100%	100%

3.7 Computation time analysis

In this section, the computation time is evaluated with a MacBook Profwith a 2.2 GHz Intel Core i7 CPU and an Intel Iris Pro 1536 MB GPU. The computational steps for each candidate finding approach are shown in Table 3.12.

From Tables 3.12 and 3.13, the feature-based method requires the highest computation time in the initial processing steps, while the normal vector-based and direct voxel grid search methods only require 1/25 1/10 the computation time of the feature-based method. This is because the feature-based method involves a higher amount of computational steps to pass an input point cloud through a deep neural network and compute high-dimensional feature vectors. Figure 3.15 shows that there exists a linear relationship between the num-

ber of points in the input point cloud and the computation time for the initial processing steps, due to having to repeat the initial processing step for each point. On the other hand, Figure 3.16 - Figure 3.18 shows that there is no clear trend for each method in terms of the computation time for element retrieval. As the number of selected exemplar points increases, the computation time of the element retrieval process increases in general but not consistently. This is because each exemplar corresponds to a different number of candidates to match, which will affect the computation time. For example, a window element consists of a small number of points but may have a large number of matches, whereas a floor element consists of a large number of points but may have fewer matches. Generally, the computation time in element retrieval steps of features-based normal-based methods is lesser than the direct approach.

Table 3.12: Computational steps involved with each candidate matching approach

Candidate finding approach	Initial processing steps			Element retrieval processing steps		
	Load point cloud data	Computing normal vector	Feature extraction	Extract same dimension point sets	Candidates finding	Matching refinement
Direct	√	×	×	√	√	√
Normal	√	√	×	√	√	√
Features	√	×	√	√	√	√

Table 3.13: Computational time involved with each building

Laser scanning datasets	Initial processing steps			Element retrieval processing steps		
	Load point cloud data (s)	Computing normal vector (s)	Feature extraction (s)	Extract same dimension point sets (s)	Candidates finding (s)	Matching refinement (s)
Dataset #1	0.05	0.15	2.76	1.16	1.07	0.04
Dataset #2	8.42	10.48	182.70	8.53	3.25	0.21
Dataset #3	1.72	2.08	38.53	1.59	1.95	0.05
Dataset #4	2.59	3.32	58.47	2.52	4.44	0.03
Dataset #5	2.98	3.32	65.18	2.73	4.62	0.05

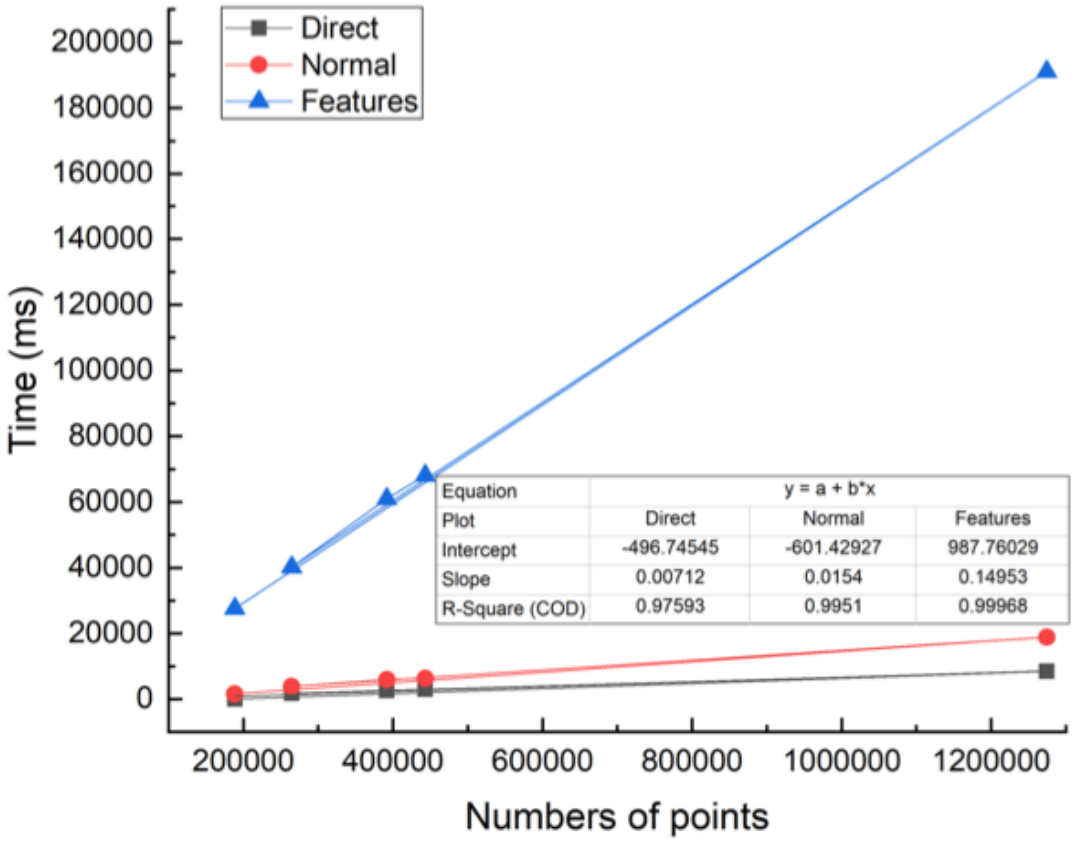


Figure 3.15: Initial processing computation time for each building point cloud data

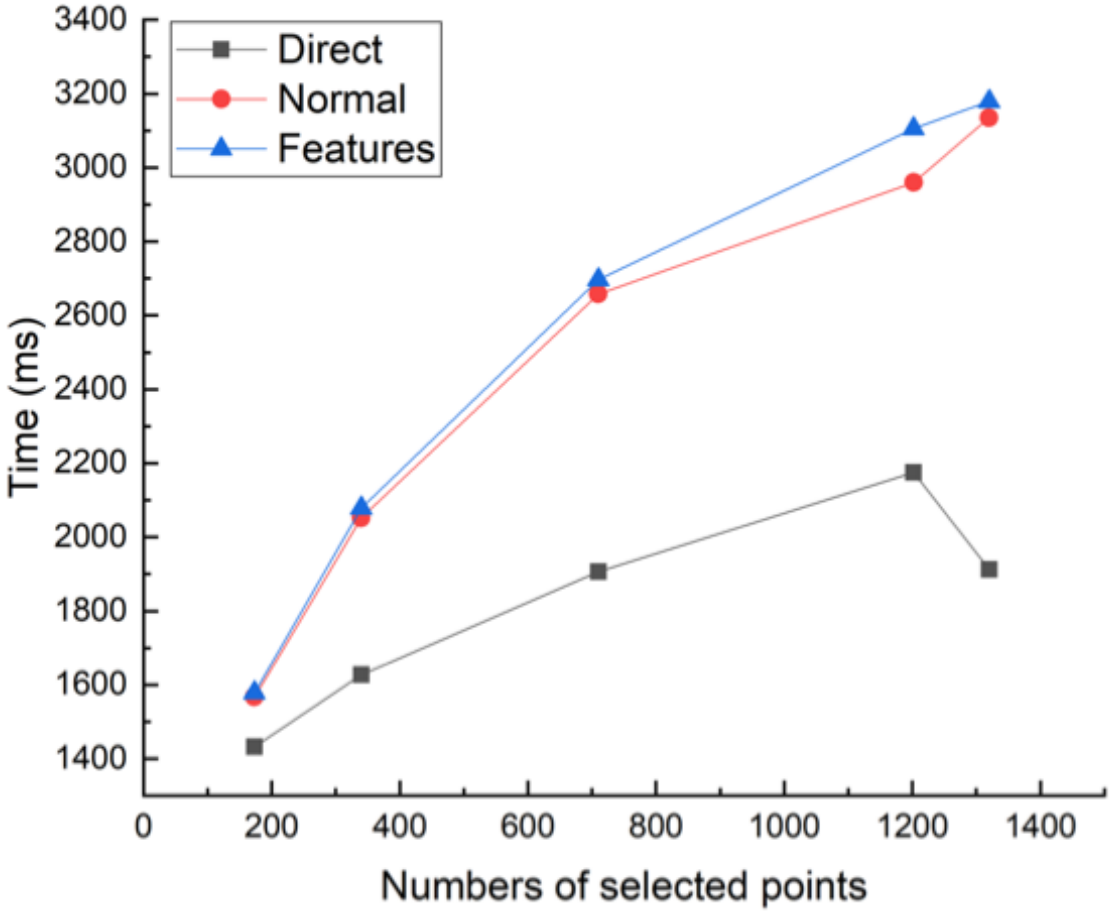


Figure 3.16: Element retrieval process computation time of building 1

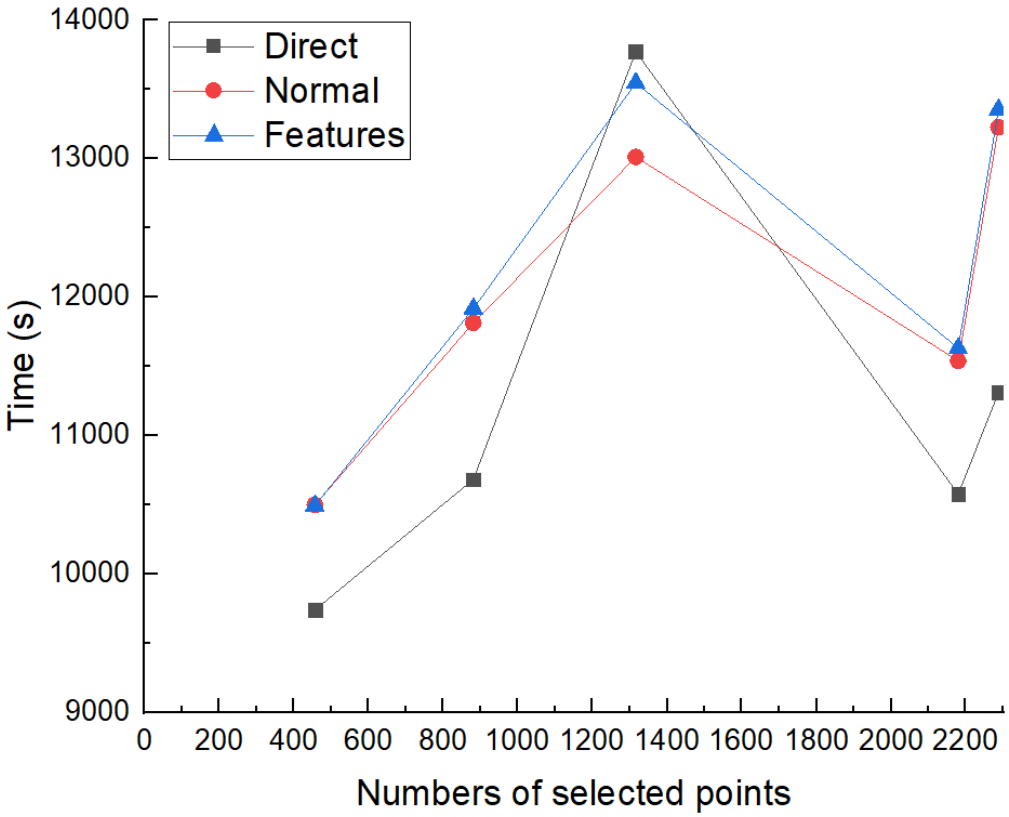


Figure 3.17: Element retrieval process computation time of building 2

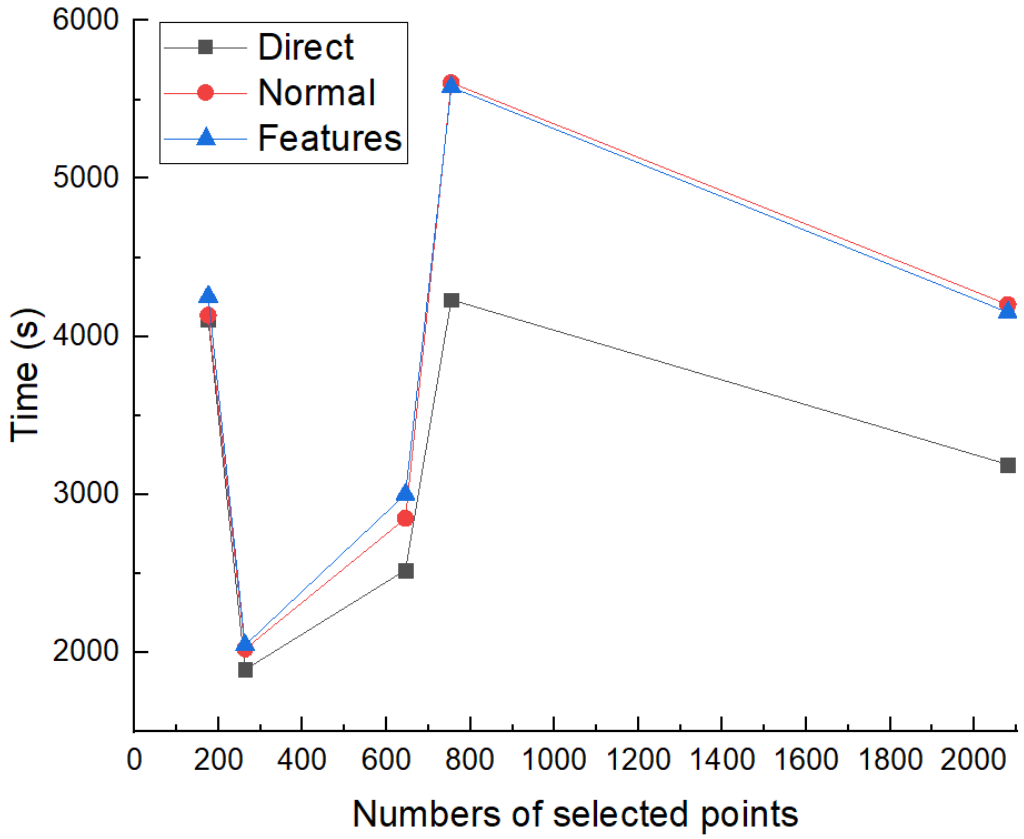


Figure 3.18: Element retrieval process computation time of building 3

CHAPTER 4
DISCUSSION

In this study, the performance of the proposed feature-based element retrieval method is demonstrated with quantitative results using precision and recall rate metrics, as well as qualitative results using visual comparison of query results. Overall, the proposed feature-based approach achieved the best precision score across Dataset 1-5, suggesting that it is able to better distinguish the correct building elements from similarly-shaped clutter when compared to the direct or normal-based matching approaches. This is because this method takes the characteristics of each point and their interconnections with neighboring points into account by computing a 50-dimensional feature vector for that point using a deep neural network. It is not limited to simply directly comparing the geometric shape or normal vectors for each point. As a result, more false-positive candidates are filtered out without excluding additional true positive matches due to incomplete data, occlusion and other factors, so the precision and recall rates maintain a desirable value. As shown in Figures 3.5 and 3.6, the proposed method can retrieve unidentified building elements and temporary equipment in a point cloud without having a pre-built BIM model, since it is based on the comparison of a user-selected exemplar and the candidate elements in the point cloud consisting of groups of points that have same dimension as the exemplar. Thus, the proposed method can work for any arbitrarily-shaped object even if it is not defined in the as-planned BIM model.

In this study, one of the goals is used to learn features in a data-driven manner from an auxiliary dataset and apply it to the actual object retrieval task in a new dataset. This is inspired by concepts in the image-based facial recognition literature [47]. For example in facial recognition, it is usually the case that the test images are from persons that have never

been seen before in the training data. In order to match images in the test dataset, the strategy used is to first learn useful features that can distinguish between different faces in the training data. Then, a pair of test images can be predicted to be similar or dissimilar based on these underlying features. This concept can be adapted for the task of building element retrieval. For example, the training dataset could contain multiple point cloud instances of Window Type A and Window Type B. The goal is to learn a model of useful features to distinguish between different types of objects from point cloud data, such that the model can tell that two instances of Window Type C are similar and that Window Type C and Window Type D are dissimilar in the validation data, even though they have not been seen in the training data. In this study, the auxiliary dataset is created by converting pre-made BIM models into synthetic point clouds which does not have noise and occlusion. The actual point cloud data collected with a laser scanner with noise and occlusion can also be used for this purpose. However, using this as training data is much more challenging because the points have to be manually labeled. Nonetheless, the object retrieval results in Tables 3.1-3.3 show that the data-driven "feature"-based method outperforms the non-data driven "direct" or "normal"-based methods. This suggests that feature learning is effective in improving the object retrieval performance even though it is difficult to factor in noise and occlusion characteristics in the training process.

In the feature computation step, it is possible to train the point features to be rotation-invariant to handle objects that exist in different orientations. However, there are several reasons why the implementation of this study is not rotation-invariant. First, the candidate matching step searches for candidate elements that are rigid translations of the exemplar so it would not be able to find rotated matches even if the features are rotation-invariant. Second, the majority of building elements considered in this study exist in the same orientation (refer Datasets 1-5). In this case, taking different orientations into consideration would only increase the number of false positives without significantly improving the recall rate.

Finally, orientation is an important distinguishing feature between different objects (e.g. a floor can be distinguished from a wall because floors are oriented in the horizontal plane whereas walls are oriented in the vertical plane).

In terms of the applicability, the study of the variability of a user-selected exemplar on the retrieval results shows that the results are within a reasonable range, being 1.5% in the precision rate and 4.4% in the recall rate. This means that differently selected instances or boundaries of the selected exemplar do not have too much impact on the building element retrieval results. In addition, the study of the effect of voxel grid resolution shows that the feature-based candidate matching method achieves high average precision and recall rates across different voxel resolutions. Although the fewer number of points can be compared and matched when the voxel grid resolution becomes coarser, the use of a feature-based approach allows each point to be accurately matched in high-dimensional space so that the retrieval performance can be maintained. Moreover, the accuracy of the retrieval results remains acceptable when the voxel grid resolution is varied from 0.05m to 0.2m.

The study of computation time shows that the feature-based method requires more initial processing time since computing the 50-dimensional feature vector is time-consuming. However, the actual computation time of element matching and obtaining retrieval results is relatively fast, in the order of ten seconds. This property is beneficial since the initial preprocessing steps only have to be performed a single time for each building point cloud, whereas the element matching and retrieval steps have to be performed multiple times for different user-selected exemplars.

The proposed method faces several limitations in its current form. First, it is a semi-automated method which needs user input. Therefore, it requires some domain knowledge by the user to select the correct building element exemplar with suitable boundaries through

the user interface. To overcome this limitation, the user interface can show the point cloud overlaid with color images of objects acquired on the same site to make it more visually clear and intuitive for users to select objects.

In addition, this method still requires that the scanned point cloud data be mostly complete to obtain optimized building retrieval results. Although it is less sensitive to noise, occlusion, and scanning artefacts compared to other methods, it is unable to retrieve building elements that have too many missing points in the point cloud. One possible solution is to combine point clouds across different modalities such as ground-based laser scanning and aerial-based scanning, as well as across different stages of construction where different building elements would be exposed and visible in the point cloud. Another possible solution is to still consider candidate elements with weak matching scores, and apply an additional classification step to determine whether it is a valid match.

CHAPTER 5
CONCLUSION

In this study, a semi-automatic building element retrieval method was introduced to identify similar building elements with a user-provided exemplar from a point cloud scene. The proposed method can distinguish the correct building elements from similarly-shaped clutter even for complex building elements when compared to the direct voxel grid matching and normal vector matching approaches. This method is advantageous in that it can detect temporary and unidentified building elements without the pre-built model data during the construction phase. In addition, different selected instances or boundaries of the selected exemplar do not have too much impact on the building element retrieval results. The accuracy of the retrieval results remains acceptable when the voxel grid resolution is varied from 0.05m to 0.2m, and the actual computation time is reasonably efficient. Although there are several remaining limitations of the user-selection interface and requirement of completeness for the scanned point cloud data, potential solutions have been suggested including improving point cloud visualization in the user interface and combining multiple scan data. For future work, color information can be utilized to extract more robust features for each point to improve building elements retrieval results. Furthermore, the proposed method can be applied to make the process of converting scanned point clouds into as-built BIM (scan-to-BIM) more time-efficient, especially for complex historic buildings, where this approach can assist in determining the positions and assigning semantic information of the building elements automatically to reduce the workload of labeling repetitive elements. More complex building point cloud data will be used to verify the efficiency and accuracy for future promising applications in construction progress monitoring, deviation detection, and material inventory checking.

REFERENCES

[1] Q. Wang and M. K. Kim, "Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018," *Advanced Engineering Informatics*, vol. 39, no. February, pp. 306–319, 2019.

[2] J. Gong and C. H. Caldas, "Data processing for real-time construction site spatial modeling," *Automation in Construction*, vol. 17, no. 5, pp. 526–535, 2008.

[3] F. Remondino and S. El-Hakim, "Image-based 3D Modelling: A Review," *The Photogrammetric Record*, vol. 21, no. 115, pp. 269–291, 2006.

[4] F. Remondino and C. Fraser, "Digital Camera Calibration Methods: Considerations and Comparisons," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 266–272, 2006.

[5] S. Verykokou, C. Ioannidis, G. Athanasiou, N. Doulamis, and A. Amditis, "3D reconstruction of disaster scenes for urban search and rescue," *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 9691–9717, 2018.

[6] D. Moon, S. Chung, S. Kwon, J. Seo, and J. Shin, "Comparison and utilization of point cloud generated from photogrammetry and laser scanning: 3D world model for smart heavy equipment planning," *Automation in Construction*, vol. 98, no. June 2017, pp. 322–331, 2019.

[7] J. Chen and Y. K. Cho, "Point-to-point Comparison Method for Automated Scan-vs-BIM Deviation Detection," in *17th International Conference on Computing in Civil and Building Engineering*, 2018.

[8] M. Wang, Y. Deng, J. Won, and J. C. Cheng, "An integrated underground utility management and decision support based on BIM and GIS," *Automation in Construction*, vol. 107, p. 102 931, 2019.

[9] J. Chen, Y. K. Cho, and K. Kim, "Region Proposal Mechanism for Building Element Recognition for Advanced Scan-to-BIM Process," in *Construction Research Congress 2018*, vol. 2018-April, Reston, VA: American Society of Civil Engineers, 2018, pp. 221–231, ISBN: 9780784481264.

[10] A. Borodinecs, J. Zemitis, M. Dobelis, and M. Kalinka, "3D scanning data use for modular building renovation based on BIM model," *MATEC Web of Conferences*, vol. 251, A. Volkov, A. Pustovgar, and A. Adamtsevich, Eds., p. 03 004, 2018.

[11] D. Rebolj, Z. Pučko, N. Č. Babič, M. Bizjak, and D. Mongus, "Point cloud quality requirements for Scan-vs-BIM based automated construction progress monitoring," *Automation in Construction*, vol. 84, no. August, pp. 323–334, 2017.

[12] C. Portalés, J. Rodrigues, A. Rodrigues Gonçalves, E. Alba, and J. Sebastián, "Building Information Modeling for Cultural Heritage: The Management of Generative Process for Complex Historical Buildings F.," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 58, 2018.

[13] F. Bosche, C. T. Haas, and B. Akinci, "Automated Recognition of 3D CAD Objects in Site Laser Scans for Project 3D Status Visualization and Performance Control," *Journal of Computing in Civil Engineering*, vol. 23, no. 6, pp. 311–318, 2009.

[14] F. Bosché, "Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction," *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 107–118, 2010.

[15] J. Chen, Z. Kira, and Y. K. Cho, "Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction," *Journal of Computing in Civil Engineering*, vol. 33, no. 4, p. 04 019 027, 2019.

[16] A. Dimitrov and M. Golparvar-Fard, "Segmentation of building point cloud models including detailed architectural/structural features and MEP systems," *Automation in Construction*, vol. 51, no. C, pp. 32–45, 2015.

[17] J. Chen, Y. K. Cho, and K. Kim, "Region Proposal Mechanism for Building Element Recognition for Advanced Scan-to-BIM Process," in *Construction Research Congress 2018*, vol. 2018-April, Reston, VA: American Society of Civil Engineers, 2018, pp. 221–231, ISBN: 9780784481264.

[18] C. Wang, Y. K. Cho, and C. Kim, "Automatic BIM component extraction from point clouds of existing buildings for sustainability applications," *Automation in Construction*, vol. 56, pp. 1–13, 2015.

[19] X. Xiong, A. Adan, B. Akinci, and D. Huber, "Automatic creation of semantically rich 3D building models from laser scanner data," *Automation in Construction*, vol. 31, no. May, pp. 325–337, 2013.

[20] M. Bueno, F. Bosché, H. González-Jorge, J. Martínez-Sánchez, and P. Arias, "4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models," *Automation in Construction*, vol. 89, no. December 2017, pp. 120–134, 2018.

[21] V. S. Kalasapudi, P. Tang, and Y. Turkan, "Computationally efficient change analysis of piece-wise cylindrical building elements for proactive project control," *Automation in Construction*, vol. 81, no. February, pp. 300–312, 2017.

[22] J. Lee, H. Son, C. Kim, and C. Kim, "Skeleton-based 3D reconstruction of as-built pipelines from laser-scan data," *Automation in Construction*, vol. 35, pp. 199–207, 2013.

[23] T. Czerniawski, M. Nahangi, C. Haas, and S. Walbridge, "Pipe spool recognition in cluttered point clouds using a curvature-based shape descriptor," *Automation in Construction*, vol. 71, no. Part 2, pp. 346–358, 2016.

[24] J. Chen, Y. Fang, and Y. K. Cho, "Unsupervised Recognition of Volumetric Structural Components from Building Point Clouds," in *Computing in Civil Enginerring*, 2017, pp. 34–42.

[25] J. Jung, C. Stachniss, S. Ju, and J. Heo, "Automated 3D volumetric reconstruction of multiple-room building interiors for as-built BIM," *Advanced Engineering Informatics*, vol. 38, no. October, pp. 811–825, 2018.

[26] S. Pu and G. Vosselman, "Knowledge based reconstruction of building models from terrestrial laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 6, pp. 575–584, 2009.

[27] H. Han, X. Han, F. Sun, and C. Huang, "Point cloud simplification with preserved edge based on normal vector," *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 19, pp. 2157–2162, 2015.

[28] H. Tran, K. Khoshelham, A. Kealy, and L. Díaz-Vilariño, "Shape Grammar Approach to 3D Modeling of Indoor Environments Using Point Clouds," *Journal of Computing in Civil Engineering*, vol. 33, no. 1, p. 04 018 055, 2019.

[29] Z. Zhou, J. Gong, and X. Hu, "Community-scale multi-level post-hurricane damage assessment of residential buildings using multi-temporal airborne LiDAR data," *Automation in Construction*, vol. 98, no. July 2017, pp. 30–45, 2019.

[30] Z. Zhou and J. Gong, "Automated Analysis of Mobile LiDAR Data for Component-Level Damage Assessment of Building Structures during Large Coastal Storm Events," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 5, pp. 373–392, 2018.

[31] Y.-K. Cho and C. T. Haas, "Rapid Geometric Modeling for Unstructured Construction Workspaces," *Computer-Aided Civil and Infrastructure Engineering*, vol. 18, no. 4, pp. 242–253, 2003.

[32] Y.-K. Cho, C. T. Haas, K. Liapi, and S. Sreenivasan, "A framework for rapid local area modeling for construction automation," *Automation in Construction*, vol. 11, no. 6, pp. 629–641, 2002.

[33] J. Liu, Q. Zhang, J. Wu, and Y. Zhao, "Dimensional accuracy and structural performance assessment of spatial structure components using 3D laser scanning," *Automation in Construction*, vol. 96, no. October, pp. 324–336, 2018.

[34] M.-K. Kim, Q. Wang, S. Yoon, and H. Sohn, "A mirror-aided laser scanning system for geometric quality inspection of side surfaces of precast concrete elements," *Measurement*, vol. 141, pp. 420–428, 2019.

[35] C. Kim, H. Son, and C. Kim, "Automated construction progress measurement using a 4D building information model and 3D data," *Automation in Construction*, vol. 31, pp. 75–82, 2013.

[36] C. Zhang and D. Arditi, "Automated progress control using laser scanning technology," *Automation in Construction*, vol. 36, pp. 108–116, 2013.

[37] B. Bortoluzzi, I. Efremov, C. Medina, D. Sobieraj, and J. McArthur, "Automating the creation of building information models for existing buildings," *Automation in Construction*, vol. 105, no. March, p. 102 838, 2019.

[38] Y. Turkan, F. Bosché, C. T. Haas, and R. Haas, "Tracking of secondary and temporary objects in structural concrete work," *Construction Innovation*, vol. 14, no. 2, pp. 145–167, 2014.

[39] K. Makantasis, A. Doulamis, N. Doulamis, and M. Ioannides, "In the wild image retrieval and clustering for 3D cultural heritage landmarks reconstruction," *Multimedia Tools and Applications*, vol. 75, no. 7, pp. 3593–3629, 2016.

[40] J. Chen, Y. Fang, and Y. K. Cho, "Performance evaluation of 3D descriptors for object recognition in construction applications," *Automation in Construction*, vol. 86, no. September 2017, pp. 44–52, 2018.

[41] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *2011 IEEE International Conference on Robotics and Biomimetics*, IEEE, 2011, pp. 2987–2992, ISBN: 978-1-4577-2138-0.

[42] Q. Wang, J. C. P. Cheng, and H. Sohn, "Automated Estimation of Reinforced Precast Concrete Rebar Positions Using Colored Laser Scan Data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 9, pp. 787–802, 2017.

[43] J. Chen, Y. Fang, Y. K. Cho, and C. Kim, "Principal Axes Descriptor for Automated Construction-Equipment Classification from Point Clouds," *Journal of Computing in Civil Engineering*, vol. 31, no. 2, p. 04 016 058, 2017.

[44] J. Chen, Y. K. Cho, and Z. Kira, "Multi-View Incremental Segmentation of 3-D Point Clouds for Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1240–1246, 2019. arXiv: 1902.06768.

[45] J. Chen and Y. K. Cho, "Exemplar-Based Building Element Retrieval from Point Clouds," in *International Conference on Smart Infrastructure and Construction 2019 (ICSIC)*, ICE Publishing, 2019, pp. 225–231, ISBN: 0-7277-6466-7.

[46] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, H. Araujo, A. M. Mendonça, A. J. Pinho, and M. I. Torres, Eds., pp. 1109–1135, 2010.

[47] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 815–823, 2015. arXiv: arXiv:1503.03832v3.

[48] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, A. Feragen, M. Pelillo, and M. Loog, Eds., Cham: Springer International Publishing, 2015, pp. 84–92, ISBN: 978-3-319-24261-3.

[49] S. Zeng, J. Chen, and Y. Cho, "User exemplar-based building element retrieval from raw point clouds using deep point-level features," *Automation in Construction (in press)*, 2020.

[50] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June, IEEE, 2015, pp. 1912–1920, ISBN: 978-1-4673-6964-0. arXiv: 1406.5670.

[51] I. Cherabier, C. Hane, M. R. Oswald, and M. Pollefeys, "Multi-Label Semantic 3D Reconstruction Using Voxel Blocks," in *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 2016, pp. 601–610, ISBN: 978-1-5090-5407-7.

[52] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Montreal, 2014, pp. 3320–3328.

[53] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 77–85, 2017. arXiv: 1612.00593.

[54] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2569–2578, ISBN: 9781538664209. arXiv: 1711.08588.

[55] J Pfanzagl, "Studies in the history of probability and statistics XLIV. A forerunner of the t-distribution," *Biometrika*, vol. 83, no. 4, pp. 891–898, 1996.

[56] N. Doulamis and A. Doulamis, "Evaluation of relevance feedback schemes in content-based in retrieval systems," *Signal Processing: Image Communication*, vol. 21, no. 4, pp. 334–357, 2006.

[57] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Systems*, vol. 8, no. 6, pp. 536–544, 2003.

[58] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *Journal of the American Society for Information Science*, vol. 41, no. 4, pp. 288–297, 1990.