PCIM: DEEP LEARNING-BASED POINT CLOUD INFORMATION MODELING FRAMEWORK

A Dissertation
Presented to
The Academic Faculty

by

Jisoo Park

In Partial Fulfillment
of the Requirements for the Degree
Ph. D. in the
School of Civil and Environmental Engineering

Georgia Institute of Technology
December 2020

PCIM: DEEP LEARNING-BASED POINT CLOUD INFORMATION MODELING FRAMEWORK

Approved by:

Dr. Yong K. Cho, Advisor
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. W. Hong Yeo
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Russell Gentry
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Yelda Turkan
School of Civil and Construction Engineering
Oregon State University

Dr. Eric Marks
School of Civil and Environmental Engineering
Georgia Institute of Technology

Date Approved: November 11, 2020

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to all who helped me along my journey of Ph.D. studies. Among many, my utmost gratitude goes to my doctoral advisor, Dr. Yong K. Cho, for his patience, support, advice, and guidance throughout my study and research. He sometimes has been a mentor of my life like my parents. Also, I would like to thank the other members of my doctoral committee, including Drs. Russell Gentry, Yelda Turkan, Eric Marks, and W. Hong Yeo for their time and advice.

Much of my success is attributed to my colleagues. The support of the former and current members of RICAL, including Inbae Jeong, Youjin Jang, KyungKi Kim, JeeWoong Park, Yihai Fang, Jingdao Chen, Phil Eun Kim, Kinam Kim, and Yosuke Yajima, is highly appreciated and was critical to my accomplishment.

To conclude, I cannot forget to thank my family members for all the unconditional support during the compilation of this dissertation. Without my parents, Sang-il Park and Mi-kyung Sin, I may still have been wandering without finding the right destination in this long journey. I believe that I am truly blessed to be their son. Also, I would like to express my gratitude to my wife's parents, Yong Kwon and Hae Kwon, who always believed and supported me. My wife, Emma, was the most devoted and faithful partner of this long journey. She was the most significant source of my inspiration, enthusiasm, and motivation. I would like to dedicate this dissertation to her. Finally, my grandmother and sister must be noted for their constant love, concern, support, and encouragement.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **iii**

LIST OF TABLES **vi**

LIST OF FIGURES **vii**

LIST OF SYMBOLS AND ABBREVIATIONS **ix**

SUMMARY **xi**

CHAPTER 1. Introduction **1**
1.1 Building Information Modeling (BIM) **1**
1.2 Laser Scanned Point Cloud **2**
1.3 Scan-to-BIM **3**
1.4 Semantic Segmentation **4**
1.5 Research Needs/Knowledge Gaps **5**

CHAPTER 2. Research goal and objectives **7**
2.1 Research Goal **7**
2.2 Research Objectives and Scopes **7**
2.2.1 Objective 1: Point Cloud Classification 8
2.2.2 Objective 2: Data Schema Development for PCIM 9

CHAPTER 3. Literature review **11**
3.1 Point Cloud Segmentation **11**
3.1.1 Object segmentation 11
3.1.2 Material segmentation 15
3.2 Point Cloud Data Schema **17**
3.2.1 Data schema of IFC 17
3.2.2 Data schema to represent point cloud data 20
3.2.3 Point cloud data structure 21
3.3 Point of departure **23**

CHAPTER 4. Research Framework **26**

CHAPTER 5. Point cloud classification **29**
5.1 Construction Material Classification **30**
5.1.1 Laser intensity normalization 30
5.1.2 PointNet++ 32
5.1.3 Dataset for architecture training 34
5.1.4 Material classification performance evaluation 37
5.2 Construction Object Classification **41**
5.2.1 Hierarchical deep learning-based object 41
5.2.2 Object classification performance evaluation 44
5.3 Spatial Reasoning **47**

5.3.1 XY-histogram based building boundary prediction 48
5.3.2 Z-histogram based building level prediction 49
5.3.3 Point Cloud labeling 50

CHAPTER 6. PCIM data schema 52
6.1 Data schema development 52
6.1.1 Entity inheritance 52
6.1.2 PCIM data parser and viewer 57
6.2 Validation of the Integrated PCIM Framework from a Case Study 57
6.2.1 Test set-up 57
6.2.2 Classification module 58
6.2.3 Data Parsing module 62
6.2.4 PCIM viewer 63
6.2.5 Comparison between Scan-to-BIM and PCIM 65

CHAPTER 7. CONCLUSION AND DISCUSSION 68
7.1 Summary 68
7.2 Research Contribution and Impact 70
7.3 Limitations and Future Research 70

REFERENCES 72

LIST OF TABLES

Table 1. Deep learning-based object segmentation algorithms 13

Table 2. 3D benchmark datasets for point cloud segmentation 14

Table 3. Comparison of existing approaches for the construction material classification 16

Table 4. A research framework for PCIM implementation 27

Table 5. Classification performance comparison between PointNet and PointNet++...... 34

Table 6. The laser scanner's specification 35

Table 7. The description of the material and object datasets 36

Table 8. Comparison of the material classification performance 41

Table 9. Comparison of the construction object classification performance.................... 46

Table 10. Confusion matrix of the construction object classification for the case study . 60

Table 11. Comparison between PCIM and Scan-to-BIM 66

LIST OF FIGURES

Figure 1. The loss of reality and details during the solidifying process in automated Scan-to-BIM. (a): the original point cloud, and (b): the solid model from automatic Scan-to-BIM 4

Figure 2. A simplified example of the entity inheritance of building elements in IFC 19

Figure 3. Entity-relationship diagram for representing ‘IfcBeam’ element 20

Figure 4. Overall PCIM framework 26

Figure 5. Hierarchical classification process in PCIM 30

Figure 6. Intensity histograms of various types of surface material. (a): soil, (b): concrete, (c): steel, and (d): wood 31

Figure 7. PointNet++ architecture (Qi, Yi, et al. 2017) 33

Figure 8. The categories of the material and object datasets 37

Figure 9. Material classification results. (a): the original point cloud, (b): the ground truth labeled point cloud, and (c): segmentation results 39

Figure 10. The classification errors in wet surfaces 40

Figure 11. Hierarchical deep learning-based object classification 44

Figure 12. Construction object classification results. (a): the original point cloud, (b): the ground truth labeled point cloud, and (c): segmentation results 45

Figure 13. Flowchart of the spatial reasoning module in PCIM 47

Figure 14. XY-histogram-based building boundary recognition 48

Figure 15. Z-histogram-based building level (floors) recognition 50

Figure 16. Information labeled point cloud after classification processes 51

Figure 17. Entity-relationship diagram (ERD) of the PCIM data schema........................ 53

Figure 18. Definition of 'object' element in PCIM.xsd.. 55

Figure 19. Definition of 'IfcElement' element in IFC4.3 RC1.. 56

Figure 20. A registered point cloud model of a building under construction for PCIM framework validation.. 58

Figure 21. Material classification results for the test site 59

Figure 22. The results of the object classification using the material indexes 59

Figure 23. XY-histogram of the field test site 61

Figure 24. Z-histogram of the field test site.. 61

Figure 25. A part of the sample XML file generated by the PCIM framework for test site (Text view @ OxygenXML Viewer)... 62

Figure 26. A part of the sample XML file generated by the PCIM framework for the test site (Grid view @ OxygenXML Viewer) 63

Figure 27. The PCIM viewer highlighting a selected wall in a DB tree........................... 64

Figure 28. 3D model of a wall generated by Scan-to-BIM @ Revit 67

LIST OF SYMBOLS AND ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
AEC	Architecture, Engineering, and Construction
ALS	Aerial Laser Scanning
ANN	Artificial Neural Network
BIM	Building Information Modeling
CNN	Convolutional Neural Network
FC-CRF	Fully Connected Conditional Random Fields
FN	False Negative
FP	False Positive
FPFH	Fast Point Feature Histograms
GPS	Global Positioning System
GUID	Globally Unique Identifier
IFC	Industry Foundation Classes
KNN	K-Nearest Neighborhood
MLP	Multi-Layer Perception
MLS	Mobile Laser Scanning
NB	Naïve Bayes
NDT	Normal Distribution Transformation
PCA	Principal Component Analysis
PCIM	Point Cloud Information Modeling
PCL	Point Cloud Library

RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RGB	Red, Green, Blue
RGBD	Red, Green, Blue, Depth
STEP	Standard for The Exchange of Product
SVM	Support Vector Machine
TLS	Terrestrial Laser Scanning
TN	True Negative
TP	True Positive
XML	Extensible Markup Language
XSD	XML Schema Definition

SUMMARY

Although building information modeling (BIM) has been widely used in entire construction projects for data exchange between stakeholders, it sometimes rarely represents the current state of the construction sites because most of the information models are built at the design phase. For this reason, several studies have implemented a process to generate as-built BIM by leveraging laser-scanned point cloud data, referred to as Scan-to-BIM. However, the conventional Scan-to-BIM process is usually performed manually or semi-manually, which is time-consuming and labor-intensive. Moreover, captured reality and measurements such as actual dimensions, shapes, colors, and damages of the target subjects that the original point cloud retains can be disregarded during the solid modeling process in Scan-to-BIM. To address this issue, this research proposes an object-oriented information modeling framework for point cloud data, named point cloud information modeling (PCIM).

The main objective of this dissertation is to develop artificial intelligence (AI)-driven information modeling framework for point cloud data. To this end, this study 1) presents methods of classifying construction objects and their properties and 2) proposes a data schema to represent the classified information with an object-based hierarchy structure. At this time, the scope of this research is limited to building construction area, but the data schema can be extended to other jobs such as mechanical, electrical, and plumbing (MEP) engineering. The findings of this research may rebound to the benefit of stakeholders of construction projects considering that point clouds play an essential role in the construction management phase. Since this research will provide an automated

information modeling solution for point cloud data, stakeholders that apply the proposed approach will save time to generate an as-is construction site model. Moreover, this research may fill the gaps in current studies on object classification in 3D by leveraging extended input channels such as laser intensity and material index. As this paper presents the concept of PCIM, various follow-up studies are expected to be additionally derived.

CHAPTER 1. INTRODUCTION

CHAPTER 1 introduces limitations of current as-built modeling and related current practices and previous studies that have presented advanced technology to develop applications for point cloud data processing. This chapter also identifies existing challenges and drawbacks of current practices and prior research and then derives research/knowledge gaps for this dissertation's basis.

1.1 Building Information Modeling (BIM)

In the Architecture, Engineering, and Construction (AEC) industries, building information model (BIM) has been widely used throughout construction projects, from pre-construction to post-construction, for information exchange between stakeholders. In the planning and designing phases, BIM enables designers and engineers to record comprehensive information of individual building objects and their relationships as well as to visualize them (Eastman et al. 2008; Miettinen and Paavola 2014). For the built environments, BIM applications have potential advantages in quality control (Boukamp and Akinci 2007), heritage documentation (Eastman et al. 2008), as-built visualization (Pătrăucean et al. 2015), and monitoring and maintenance (Motawa and Almarshad 2013).

Despite the advantages of utilizing BIM, the use of BIM in practice for contractors and field mangers is relatively low compared to architects and engineers due to several technical reasons (Moreno, Olbina, and Issa 2019). One of the reasons is that as-designed BIM, has limitations in characterizing the as-is status of the actual construction sites (Anil, Akinci, and Huber 2011). For example, conventional BIMs omit temporary structures such as scaffolding, formwork, and shoring (Hyunjoo and Hongseob 2011; Kyungki, Yong, and Kinam 2018) and stored construction materials (Yu, Li, and Luo 2016) that commonly exist at the construction sites. Besides, BIM barely reflects reality, such as true color, shading, and the actual dimension of building components (Tan Qu and Wei Sun 2015). To address these limitations, the use of laser-scanned point clouds has been emerging (Adán et al. 2018; Bosché, Ahmed, Turkan, Carl T. Haas, et al. 2015; Hamledari, Rezazadeh Azar, and McCabe 2018).

1.2 Laser Scanned Point Cloud

As a means of representing reality measurements, laser scanned-point clouds have been used for various purposes in the AEC industry. (Mukupa et al. 2017) demonstrated the use of terrestrial laser scanning (TLS) for change detection and deformation monitoring in construction sites. They stated that the use of TLS has increased in change detection and deformation monitoring of structures as a surveying technique necessitated by advancements in modern technology. (Zhang and Arditi 2020) presented an application of laser-scanned point clouds for progress control of infrastructure construction. They argued that laser scanning technology could monitor the project's progress in a real construction environment with limited human input. Currently, not only ground laser scanning but also mobile laser scanning (MLS) and airborne laser scanning (ALS) methods have been introduced. (Wang et al. 2020) reviewed the applications of MLS for urban 3D modeling. (Vo, Laefer, and Bertolotto 2016) described the availability of airborne laser scanning data for urban surveying and monitoring. More recently, (Kim et al. 2019; Park et al. 2019) presented as-built 3D modeling frameworks using a mobile laser scanning robot for construction progress monitoring. To sum up, the advancement of automated scanning technology has led to an increase in the use of point clouds in AEC industry because the point cloud can represent the as-is 3D geometry of objects. However, these point clouds do not contain semantic information such as type, material, and location of the objects. For this reason, additional processes are needed for point clouds to achieve semantic enrichment.

1.3 Scan-to-BIM

Several researchers have developed a method for converting the point cloud into a semantic-rich 3D solid model by detecting and classifying the objects in the point clouds called Scan–to-BIM. (Tang et al. 2010) introduced methods for automated geometric modeling, object recognition and object relationship modeling from laser-scanned point clouds. (Bosché, Ahmed, Turkan, Carl T Haas, et al. 2015) proposed a way of converting the laser-scanned data into BIM by aligning the point clouds to CAD models. (Macher, Landes, and Grussenmeyer 2017) presented a semi-automatic 3D reconstruction method for a building inside from point clouds. (Chen, Kira, and Cho 2019) proposed an automated Scan-to-3D reconstruction pipeline using a multi-layer perception (MLP). Nonetheless, conventional Scan-to-BIM may cause a loss of reality and details that the original point clouds had during the geometric modeling process, as shown in Figure 1. Moreover, implementing the Scan-to-BIM in dynamically changing construction sites is challenging because it requires a significant amount of time and intensive labor for the classification and modeling process (Chen and Cho 2018), which is not feasible during construction.

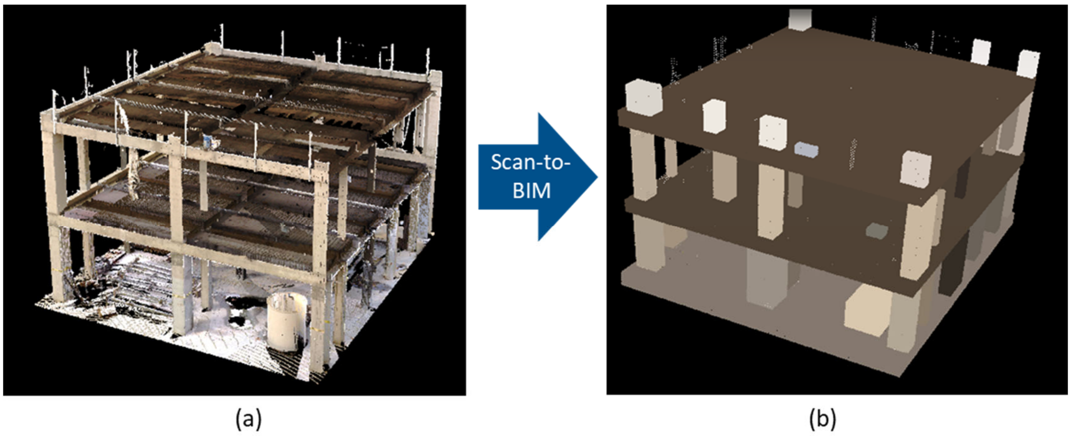


Figure 1. The loss of reality and details during the solidifying process in automated Scan-to-BIM. (a): the original point cloud, and (b): the solid model from automatic Scan-to-BIM

1.4 Semantic Segmentation

To mitigate these challenges in the Scan-to-BIM process, a few studies have presented methods for automatically segmenting objects and labeling the detected information to the point cloud data itself, called semantic segmentation (Dimitrov and Golparvar-Fard 2014a; Fehr et al. 2016). The semantic segmentation technologies have advanced with the development of machine learning and deep learning algorithms. Qi et al. (2017) presented a hierarchical neural network, named PointNet, using each point directly as input data for learning both local and global-level features (Qi, Su, et al. 2017). Chen et al. (2019) suggested an automated building element detection algorithm using a data-driven deep learning framework. They trained the model with 3D CAD objects collected from Google 3D Warehouse.

Regarding construction material classification, Han and Golparvar-Fard (2015) presented a construction material classification method using two separate histograms of texture and color in images (Kevin K Han and Golparvar-Fard 2015). This approach uses a support vector machine (SVM) based on a new construction material library (CML), including 20 types of construction materials. However, most existing segmentation methods just labeled the detected objects and their properties to each point without a specific data structure. The lack of data structure for the point clouds has limited their interoperability in construction projects.

1.5 Research Needs/Knowledge Gaps

Despite many efforts in the as-built modeling domains in construction, the current practices still have limitations. For establishing the research goal and objectives to develop an automated information modeling framework from laser-scanned point cloud data, research gaps are identified as follows:

1. Lack of comprehensive point cloud segmentation methods for construction objects recognition. Past research has not overcome one or more of the following:
    a. Ability to hierarchically classify construction components and their properties simultaneously
    b. Training datasets including various types of construction object and material

2) Lack of reliable and effective data structure for point cloud data to manage and represent the semantic information

3) Lack of a framework that integrates all necessary processes (from point cloud classification to information representation and visualization) for as-built construction scene understanding

CHAPTER 2. RESEARCH GOAL AND OBJECTIVES

2.1 Research Goal

This research aims to design an object-oriented information modeling framework for raw point cloud data, named PCIM. The PCIM is defined herein as an entire process, including recognizing construction objects and their properties from point cloud data and representing the information with a well-organized data structure. To achieve this, the following research questions need to be answered:

1. How can the construction objects and their properties be automatically extracted from the point cloud data?
2. How can the classified semantic data be effectively managed and represented in terms of data interoperability?

To answer these research questions, this research consists of two topics: 1) to present appropriate classification methods for point cloud data and 2) to develop a data schema to represent the classified information with a hierarchical structure.

2.2 Research Objectives and Scopes

To answer the research questions, this research establishes two primary research objectives as follows:

- The first objective, which is discussed in CHAPTER 5, is to presents methodological approaches to classify multiple construction objects and their properties in large scale point cloud data

- The second objective, which is discussed in CHAPTER 6, is to create a data schema to manage and represent the classified information with an object-oriented hierarchical format

2.2.1 Objective 1: Point Cloud Classification

The first objective of this research is to develop a methodological approach to classify the type of objects and their properties in point cloud data. In this research, the classification process is divided into three categories: 1) material classification, 2) object classification, and 3) spatial reasoning. This research employs hierarchical deep learning approaches for the material and object classification and assesses their classification performance. To improve the material classification performance, this study proposes leveraging laser intensity values for training the neural network architecture. The classified material information is also used as a feature for object classification. In addition, this research presents a point-density histogram-based spatial reasoning method to recognize the objects' location. In this way, the semantic information is sequentially labeled in the point cloud data.

In this study, the classification is limited to objects that exist at the building construction sites such as building elements (e.g., beams, columns, slabs, and walls), temporary structures (e.g., formworks, scaffoldings, and steel frames), and construction equipment (e.g., excavators and boom lifts). This study used its own dataset for the deep learning architecture training instead of using public datasets to enhance classification performance by using additional input parameters. The datasets are obtained from actual building construction sites. The validation is also performed with laser-scanned point cloud data collected from a live building construction site.

2.2.2 Objective 2: Data Schema Development for PCIM

The second objective of this research is to create a new data schema to represent the classified information in point cloud data with an object-oriented hierarchical manner. For this purpose, this research investigates the Industry Foundation Classes (IFC) data schema to understand the basic data structure for information exchange in construction projects and identify the proper information representation methods for PCIM. Since PCIM is an automated information modeling framework, only detectable objects in point clouds are considered as entities of the PCIM data schema. In PCIM data schema each object entity has a list of pointIds to represent their geometry. This study also develops a data parser to convert labeled point clouds through the classification process to a human-readable format such as Extensible Markup Language (XML) and a PCIM viewer to visualize the hierarchically classified point cloud data.

The first version of PCIM focuses on the building construction domain. Accordingly, the scope of the PCIM data schema in this study is limited to defining entities found in building construction sites. As aforementioned, PCIM data schema only defines automatically recognizable entities such as type of object, material and location. The entities for parametric modeling and relationship declaration by users are not considered in PCIM data schema. The validation of the integrated PCIM framework from classification to data parsing and visualization is performed with a full-scale field experiment at a building construction site. To identify the effectiveness of PCIM, a comparative analysis between PCIM and other Scan-to-BIM processes is conducted.

CHAPTER 3. LITERATURE REVIEW

This chapter presents a comprehensive review of previous research associated with the identified research topics: 1) semantic segmentation and 2) data structure for 3D point cloud data. Based on the literature review, the point of departure for two specific research objectives, addressed in CHAPTER 5 and CHAPTER 6, are identified.

3.1 Point Cloud Segmentation

3.1.1 Object segmentation

Conventional methods for building object recognition in the 3D point cloud can be divided into two approaches: feature extraction algorithms or the relationship of building objects. The first approach, one of the most widely used methods for building object classification, uses feature extraction algorithms such as Random Sample Consensus (RANSAC) (Fischler and Bolles 1981), Principal Component Analysis (PCA) (Jolliffe and Cadima 2016), or Fast Point Feature Histograms (FPFH) (Rusu et al., 2009). These algorithms have been presented in several studies. Notably, (Tarsha-Kurdi, Landes, and Grussenmeyer 2007) employed the Hough-transform and RANSAC algorithm to detect roof planes in airborne laser-scanned point clouds. (Wang, Cho, and Kim 2015) adopted a region growing plane segmentation algorithm to classify building components. More recently, (Li et al. 2017) proposed an improved RANSAC using Normal Distribution Transformation (NDT) cells to improve plane surface classification accuracy. The second approach involves methods of leveraging objects' contexts, such as the spatial relationships and ontological principles between the building components. For example, (Pu and

Vosselman 2009) proposed a knowledge-based object classification approach to reconstruct building façade models. They utilized observation regarding the size, location, and topology of building elements in a segmented 3D point cloud.

The recent works on semantic segmentation in 3D point cloud have adopted specifically designed neural network architectures. They can be mainly divided into three types, multi-view projection-based methods, voxelization-based methods, and point-based methods. (Boulch, Saux, and Audebert 2017) used projected RGBD images generated from 3D point cloud data for the semantic segmentation. The prediction scores from RGBD images could be improved by residual correction (Järemo Lawin et al. 2017). However, these methods have limitations in losing geometrical information during the projection steps (Guo et al. 2019). (Tchapmi et al. 2017) presented SEGCloud, which used 3D-FCNN (Long, Shelhamer, and Darrell 2014) for coarse voxel predictions and fully connected conditional random fields (FC-CRF) to enforce global consistency.

Although the volumetric representation methods preserve point clouds' neighborhood structure, the voxelization step can cause information loss. To this end, (Qi, Su, et al. 2017) introduced point-wise feature learning based on MLP, called PointNet. PointNet extracted a global feature vector from a point cloud data and performed prediction with max-pooling. (Qi, Yi, et al. 2017) also presented PointNet++ employing a hierarchical network, which remedies PointNet's disadvantage of not addressing geometric structures from the neighborhood of each point. However, the existing PointNet++ used only 3D coordinates as input parameters, which can be inefficient to classify the construction objects that are mostly large and have an ordinary shape. Therefore, this study employs modified PointNet++ for construction object classification, which uses additional input parameters such as laser intensity and material information. Table 1 compares existing deep learning-based semantic segmentation algorithms.

Table 1. Deep learning-based object segmentation algorithms

References	(Boulch et al. 2017)	(Järemo Lawin et al. 2017)	(Tchapmi et al. 2017)	(Graham, Engelcke, and van der Maaten 2017)	(Qi, Yi, et al. 2017)	Proposed
Methods	Projection-based	Projection-based	Voxel-based	Voxel-based	Point-based	Point-based
Classifier	SnapNet	Tangent-Conv	SEGCloud	Sparse-ConvNet	PointNet++	Modified PointNet++
Input parameters	RGBD	RGBD	XYZRGB	XYZ	XYZ	XYZRGB+ material index

Currently, several studies have provided full-scale 3D benchmark datasets for 3D point cloud segmentation, such as ScanNet (Dai et al. 2017), S3DIS (Armeni et al. 2016), Semantic3D (Hackel et al. 2017), and SemanticKITTI (Ioannidou et al. 2017). Those datasets contain multiple classes in each point cloud. However, since those datasets include only ordinary objects in daily use (e.g., cups, sofas, beds, cars, bicycles, etc.), it can be inefficient to detect structural elements in building construction sites (e.g., beams, walls, floors, and stairs). Additionally, the current public datasets only include 3D coordinates (XYZ) or plus color codes (RGB), which can be insufficient for construction material and object recognition. This study, therefore, builds a self-constructed point cloud dataset collected from actual construction sites. The dataset consists of a hierarchical class including temporary structures and construction equipment as well as building elements and contains laser intensity values for material classification and material index for object classification.

Table 2. 3D benchmark datasets for point cloud segmentation

Datasets	ScanNet	S3DIS	Semantic3D	Semantic-KITTI	proposed
# Points	-	273M	4000M	4549M	115M
# Classes (for eval.)	20(20)	13(13)	9(8)	28(25)	11(7)
# Scans	151	272	15	23201	23
Sensors	RGB-D	Matterport	TLS	MLS	TLS
Construction objects	Δ^*	Δ^*	X	X	O
Channel	RGBD	XYZRGB	XYZ	XYZ	XYZRGB+ intensity

* ScanNet and S3DIS include only a few building elements such as columns, walls, and floors

3.1.2 Material segmentation

The current state-of-the-art in construction material classification is to leverage machine learning approaches with material image datasets. (Dimitrov and Golparvar-Fard 2014a) suggested a material appearance-based construction material recognition method using a support vector machine (SVM) with χ^2 and radial basis function (RBF) kernel as classifiers. For training and validating the classifier, they developed a Construction Materials Library (CML) consisting of 20 primary construction materials with more than 150 images. (Son et al. 2014) presented a heterogeneous voting-based ensemble classifier. They examined the detection performance by comparing six types of single classifiers, which include SVM, artificial neural network (ANN), Naïve Bayes (NB), logistic regression (LR), k-Nearest Neighbors (KNN), and a modified decision tree algorithm referred to as C4.5.

More recently, in the field of computer science, various studies have been published on material detection. These studies are based on convolutional neural networks (CNN) with several network architectures, including AlexNet (Krizhevsky, Sutskever, and Hinton 2017), VGG-16 (Sharan et al. 2013), GoogLeNet (Szegedy et al. 2015), and ResNet (He et al. 2016). With the deep learning-based classification approaches, (Park, Chen, and Cho 2020) presented a CNN-based construction material classification method. Moreover, material databases have been developed to train network architecture, e.g., FMD (Sharan et al. 2013), MINC-2500 (Bell et al. 2015), and ImageNet7 (Hu, Bo, and Ren 2011). These contain thousands of material image samples. However, existing deep-learning-based material detection algorithms may not be suitable for detecting materials in 3D point cloud data because they are trained on high-resolution 2D images. Furthermore, since typical CNN-based algorithms classify materials with only RGB colors, the classification accuracy decreases in low-light areas or discolored materials; these conditions are typical on construction sites. This study, therefore, aims to reduce the impact of such obstacles by applying a point-based 3D neural network algorithm with additional intensity values. Table 3 shows the description of existing approaches for the construction material classification.

Table 3. Comparison of existing approaches for the construction material classification

Reference #	(Dimitrov and Golparvar-Fard 2014b)	(Son et al. 2014)	(Rashidi et al. 2016)	Proposed approach
Classifiers	SVM	SVM, ANN	SVM, RBF, MLP	Modified PointNet++
Datasets	CML	Self-developed	Self-developed	Self-developed
# Classes	20	3	3	6
# Input parameters	3 (RGB)	3 (RGB)	3 (RGB)	7 (XYZRGB +Intensity)

3.2 Point Cloud Data Schema

This study first reviewed the most recent version of IFC to establish the basis for the PCIM data schema. The studies on the development of point cloud data schema are then demonstrated.

3.2.1 Data schema of IFC

As a standard of information exchange format, IFC has steadily developed by 'buildingSmart – International Alliance for Interoperability (IAI)' with the consent of numerous researchers and engineers in AEC. The physical representation of IFC is expressed in the standard for the exchange of product (STEP) or Extensible Markup Language (XML) format, and this standard is registered to ISO 16739. Generally, the IFC has an object-oriented structure, and the data schema architecture of IFC defines four conceptual layers (buildingSMART 2018). Because the IFC was developed to support interoperability between stakeholders throughout the entire construction project, the IFC data schemas include an enormous amount of information that may not be necessary at the construction stage. This study, therefore, more focused on the IFC data schemas associated with the construction phase.

IFC defines an entity-relationship of building elements with an object-oriented hierarchy architecture. Figure 2 shows a simplified example of the entity inheritance of building elements. At the most abstract level, which is associated with Kernel schema, IFC classifies the entities into rooted and non-rooted entities. Rooted entities derive from IfcRoot, and IfcRoot is subdivided into three abstract entities: object definitions, relationships, property sets. IfcPropertyDefinition represents extensible properties' definition.

IfcRelationship represents relationships between building objects based on five fundamental relationship types: composition, assignment, connectivity, association, and definition. IfcObjectDefinition is composed of object occurrences and object types. IfcObject embodies location information such as a product installation having a serial number and physical placement. IfcTypeObject describes type definitions (or templates) such as a product type having a specific model number and typical shape. The entities of the occurrences and classes are also subdivided into six key entities: actors, controls, groups, products, processes, and resources.

IFC defines the building elements in the IfcProduct entity, which is the base class for all tangible building elements, and it is also separated into spatial elements, physical elements, structural analysis items, and others. IfcProducts have materials, shape representations, and location. The IfcSpatial elements consist of IfcSite, IfcBuilding, IfcBuildingStorey, and IfcSpace. Physical elements are divided into Building elements and Civil elements, and the Building elements entity includes Slab, Stair, Door, Wall, etc.

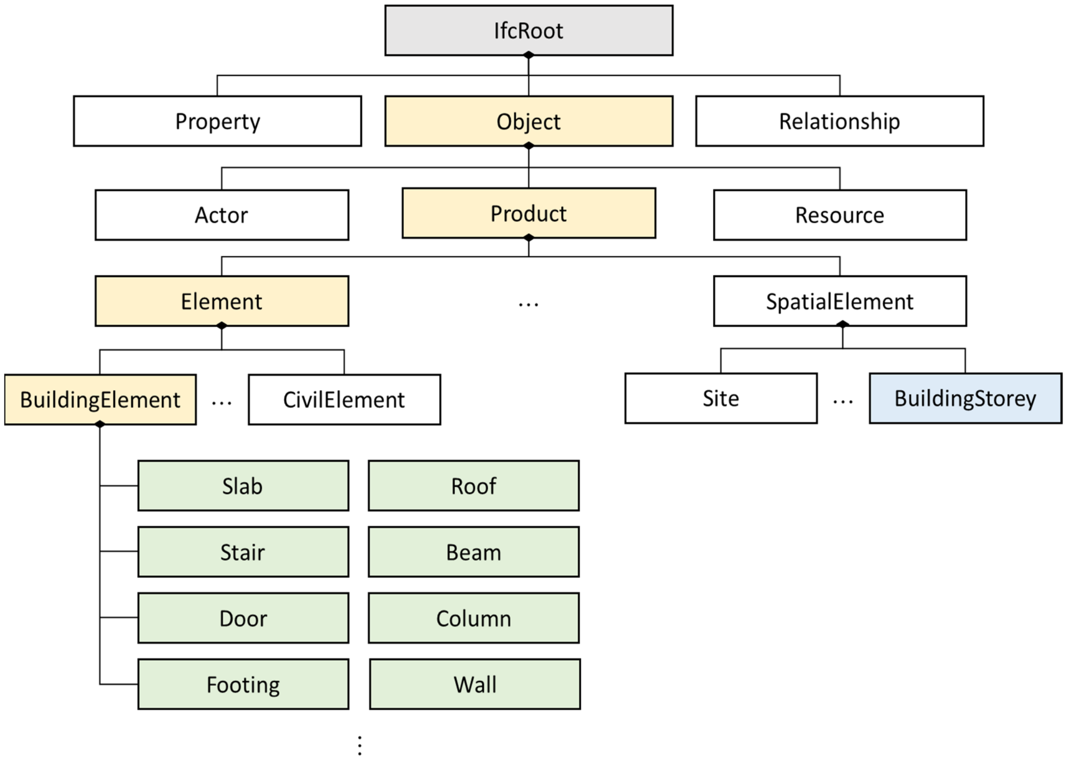


Figure 2. A simplified example of the entity inheritance of building elements in IFC

To date, the latest version of IFC schemas has pre-defined most of the entities that are occurred in actual construction, and the entities have a hierarchical structure, including the relationship between the entities. However, many of the IFC elements have many entities for parametric modeling and type declaration as shown in , which may be redundant for representing the as-is condition of construction sites with laser-scanned field data. Since the PCIM is a framework for automated as-is information modeling with field scanned point cloud, the parametric modeling entities are not required. Besides, some attributes in the IFC entities cannot be expressed automatically with only point cloud data. This study, therefore, needs to identify the necessary entities of PCIM among the entities pre-defined in IFC and to build a new PCIM unique information schema based on that of IFC.

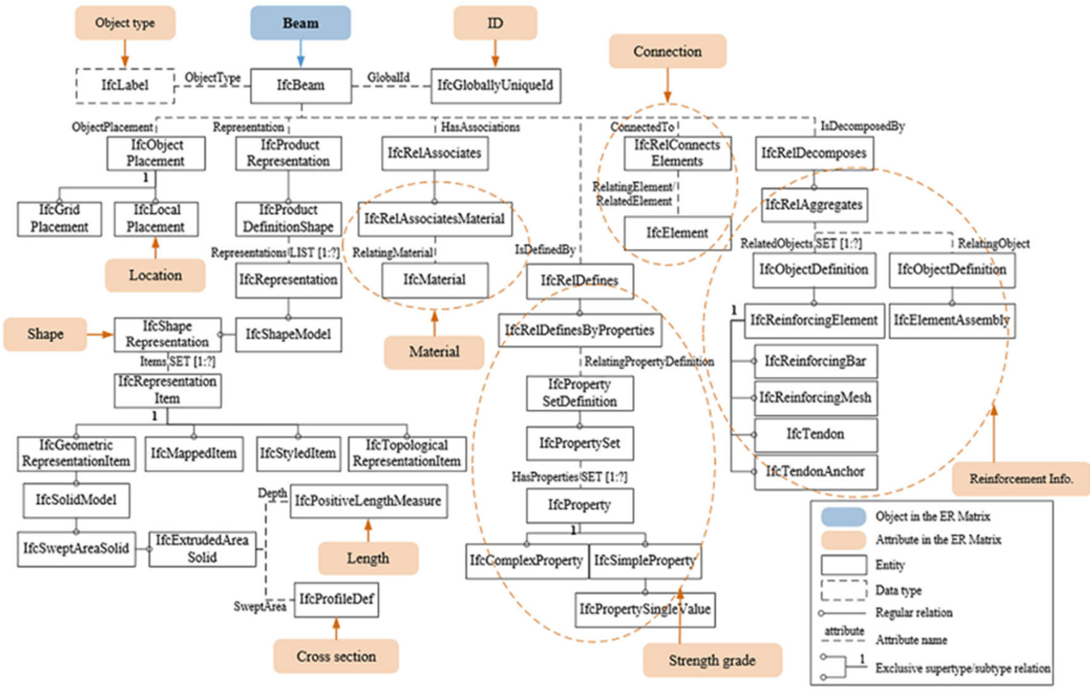


Figure 3. Entity-relationship diagram for representing 'IfcBeam' element

3.2.2 Data schema to represent point cloud data

To date, few studies on approaches to storing information on building elements in point cloud data have been presented. (Krijnen and Beetz 2017) proposed an Industry Foundation Classes (IFC) schema extension to represent point cloud data. They first overlaid the point cloud data onto BIM and found points that corresponded to elements in the BIM. The associated points were then clustered with a building element level. The geometric information, such as 3D cartesian coordinates and surface normals, were stored with an extended author-defined IFC format. One drawback of their research is their framework only works if a BIM model exists. Also, there would be many mismatches between as-is onsite point clouds and as-designed BIM if design changes were not frequently updated. (Armeni et al. 2017) presented a semantic segmentation method of point cloud data using a machine learning approach. Although their approach could automatically segment point clouds with object levels, the data schema only represented object types in the point cloud data. To complement the limitations of previous research efforts, PCIM is designed to automatically detects objects and their properties such as material, locations, and shapes in the point cloud and expresses them in a hierarchical data structure.

3.2.3 Point cloud data structure

The computer vision and computational geometry communities have produced hundreds of available file formats to describe the 3D geometry of laser-scanned point cloud. Some of these formats include:

1. PLY – a polygon file format

   The concept of PLY was derived from OBJ and purposed to store 3D data and uses lists of nominally flat polygons to represent objects. The purpose of PLY was to add extensibility capabilities and store a huge number of physical elements. The file format can depict color, transparency, surface normals, texture, coordinates, and data confidence values.

2. XYZ – a non-standardized set of files based on Cartesian coordinates

   XYZ is an archetypal ASCII file type, conveying data in lines of text. There are no unit standardizations for XYZ files. Although this format has wide compatibility across data processing programs for this type of file, the lack of standardization surrounding units and specifications makes it a fundamentally faulty method of data transfer unless additional information is supplied.

3. PCD – a native file format in PCL

   PCD, which is a standard format in Point Cloud Library (PCL), was developed by Radu B. Rusu. PCD has the ability to store and process organized point cloud datasets. This is of extreme importance for real-time applications, and research areas such as augmented reality and robotics. PCD also provides the information for n-D histograms for feature descriptors.

4. E57 – a vendor-neutral file format for point cloud storage.

   E57 was developed to store images and meta point cloud data generated by laser scanners and other 3D imaging systems. E57 has been used for the visualization of classified point cloud data at the object level. E57 can represent normals, colors, and laser intensity values.

5. LAS – an industry standard for lidar data

   LAS is an open, binary format specified by the American Society for Photogrammetry and Remote Sensing (ASPRS). A LAS file consists of a public header block, variable-length records (VLR), point data records that include global positioning system (GPS) time, RGB and near infrared image (NIR) color and wave packet information.

6. Potree – a native format for a web-based point cloud renderer

   The Institute of Computer Graphics and Algorithms developed a free open-source WebGL based point cloud renderer and their own file format, named "Potree". The Potree visualizes point clouds in a web-based environment and provides tools for editing. The Potree can export the edited point clouds that include manually classified object annotations with JSON, DXF, or Potree file format.

Additionally, the manufacturers of point cloud data processing software created their own unique file formats such as PTX (Leica), FLS (Faro), and RCP (ReCap) and their renderers. However, no existing point cloud file format organizes the points in the objects unit. Only short descriptions of the entire point cloud are described in the header lines. On the other hand, BIM file formats represent the information on an object basis. Especially, IFC applied various object-oriented information representation methods with XML or OWL formats. PCIM, therefore, requires a new point cloud file format implemented with object-based data modeling language such as XML to represent the object information detected in point clouds.

3.3 Point of departure

Relevant literatures are reviewed in the previous sections. Although various studies and practices related to the as-built modeling and information management using point cloud data have been presented, the knowledge and research gap still exist. This section summarizes the knowledge gaps found through the literature review and states the point of departure for each research objective.

1. Objective 1: Point Cloud Classification

   The state-of-the-art in 3D point cloud semantic segmentation is leveraging point-based deep learning approaches, such as PointNet and PointNet++. However, the existing point-based classification methods only uses 3D coordinates as input parameters for feature learning. This study, therefore, proposes a modified PointNet++ leveraging additional input parameters such as laser intensity and material index. Moreover, since existing public benchmark datasets for the point cloud scene segmentation rarely include construction objects and do not have laser intensity values, this study also produces self-developed benchmark datasets obtained from actual construction sites.

2. Objective 2: Data Schema Development for PCIM

   The latest version of IFC defines almost all entities that occurred in construction projects. However, many of the entities are for parametric modeling and data exchange between stakeholders. Besides, several attributes in IFC cannot be automatically classified with point cloud data. Therefore, even if the PCIM data schema follows IFC's entity definitions for building elements and material, it may contain only recognizable entities that can be detected in point cloud data. Moreover, additional entities to represent the point cloud's attributes are also defined in PCIM data schema. The labeled point cloud data through the segmentation process is converted to a human-readable data format with PCIM data parser. Furthermore, the existing point cloud viewers cannot visualize the points with an object-oriented hierarchical structure. Thus, this study also presents PCIM viewer, which have a hierarchical database (DB) tree.

CHAPTER 4. RESEARCH FRAMEWORK

This chapter presents the overall PCIM framework from classification to visualization. The integrated PCIM framework to facilitate the automated point cloud information detection and representation is proposed, as shown in Figure 4. This framework consists of three main modules, pre-processing, classification, and data parsing. The details of classification and data parsing modules are described in CHAPTER 5 and CHAPTER 6, respectively.

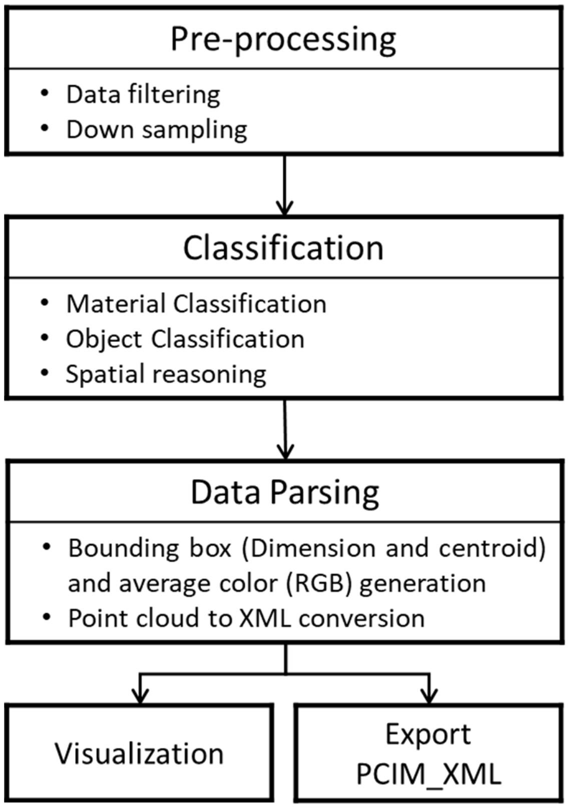


Figure 4. Overall PCIM framework

Given the framework, laser-scanned point clouds are first filtered by a radius distance and down-sampled with an octree-based balanced density down-sampling technique (El-Sayed et al., 2018). After that, the pre-processed point clouds are labeled through three classification processes, material, object, and location classification, sequentially. Then, the labeled point cloud data is parsed into a human-readable file format with a pre-defined PCIM data schema. In this step, the labeled point cloud data is clustered in object units, and the clustered data is organized with an object-oriented hierarchical structure. Finally, the visualization tool highlights the selected objects in an entity tree and exports a PCIM.xml file containing all semantic information.

To make the PCIM feasible, a research framework is established as described in Table 4. This framework consists of four layers for each of the modules that describe the sequential steps of the module implementation. The first three steps are independently performed for each module. Each module is evaluated in a different way. Then, the modules are integrated into the PCIM framework. At the end of this research, the integrated PCIM framework is compared to a Scan-to-BIM framework by a case study at the actual building construction site.

Table 4. A research framework for PCIM implementation

	Material classification	Object Classification	Spatial Reasoning	PCIM Data Parsing
Methodology	Intensity-assisted deep learning	Material-assisted deep learning	XY-, Z-histogram	XSD-based schema development

Output	Material labeled PCD*	Object labeled PCD	Location labeled PCD	PCIM.xml
Evaluation	Classification performance	Classification performance	Spatial reasoning Accuracy	Effectiveness
Analysis	Comparison between PCIM and Scan-to-BIM			
*PCD: Point Cloud Data				

CHAPTER 5. POINT CLOUD CLASSIFICATION

This chapter proposes a sequential classification approaches in three categories as follows:

1. Construction material classification
    a. Laser intensity normalization
    b. Dataset for deep network architecture training
    c. Performance evaluation
2. Construction object classification
    a. Deep learning-based construction material classification using material index
    b. Performance evaluation
3. Spatial reasoning
    a. XY-histogram-based building boundary prediction
    b. Z-histogram-based building level prediction

Figure 5 depicts the hierarchical classification process in PCIM. Given the classification process, outputs at each step are used as input parameters at the next step. Each of these steps is discussed in the following sections.

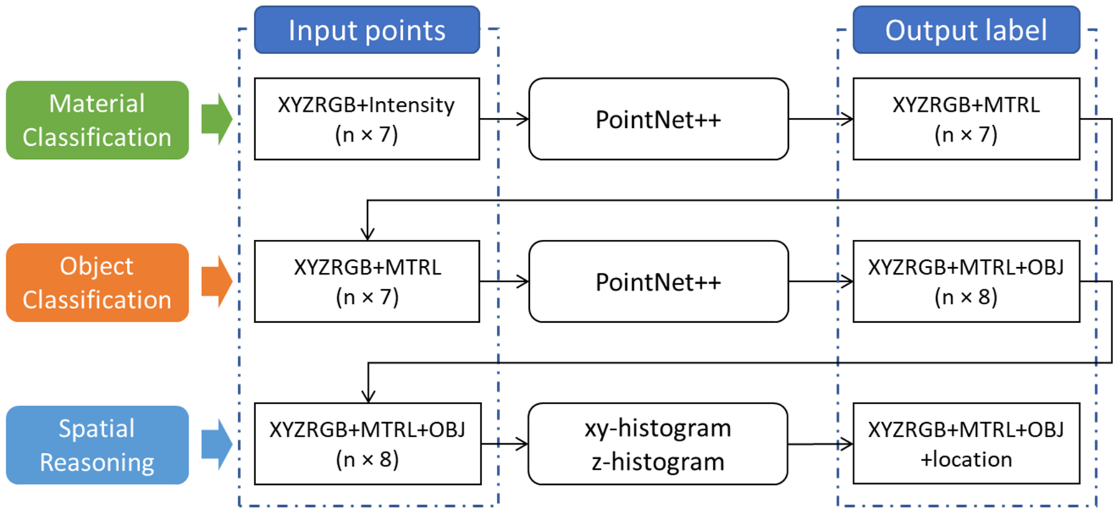


Figure 5. Hierarchical classification process in PCIM

5.1 Construction Material Classification

5.1.1 Laser intensity normalization

The intensity value is a measure of the electronic signal strength obtained from the backscattered optical. The intensity value is currently used for many purposes such as damage detection (Guldur and Hajjar 2014; Kashani and Graettinger 2015) and surface classification (Barnea and Filin 2012; Wing et al. 2015) because the returning strength of the laser pulse is influenced by material properties, incidence angles, and distance (Gross, Jutzi, and Thoennessen 2008). Equation 1 represents the received energy function based on the principle of energy conservation.

$$E_r = E_t \frac{C_t C_r T^2(R) \cos\theta}{R^2} f(c_s) \tag{1}$$

where and C_t and C_r are the constant values of the transmitter and the receiver, R and θ are the distance and incident angle to the object surface, respectively, and $f(c_s)$ is a function of all other influences such as surface material and geometry. That is, the laser intensity value can be varied by the type of surface material, as shown in Figure 6.

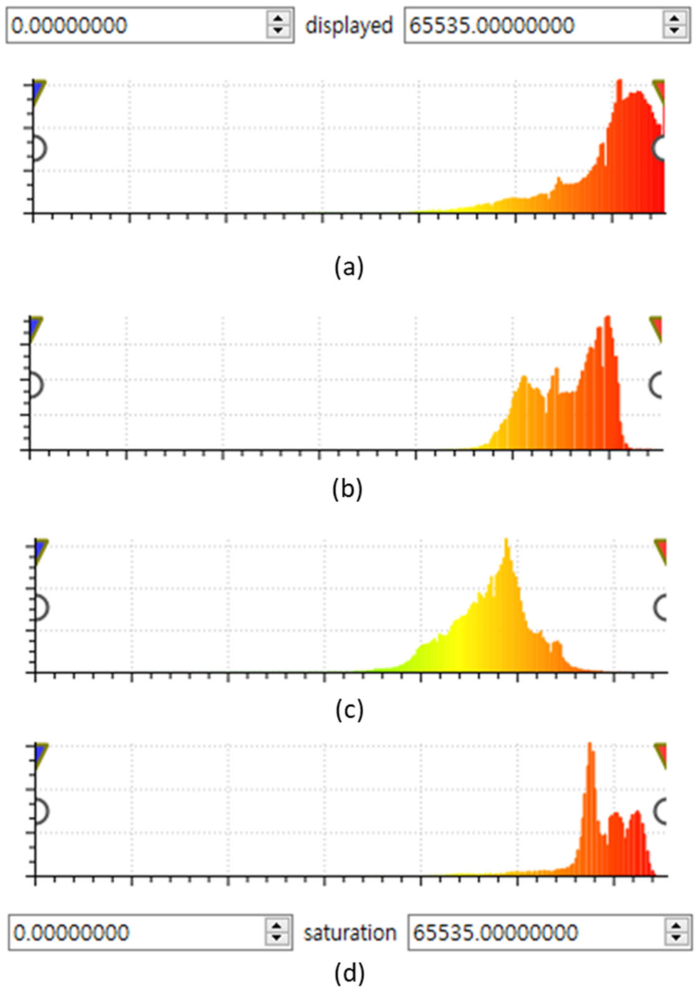


Figure 6. Intensity histograms of various types of surface material. (a): soil, (b): concrete, (c): steel, and (d): wood

For this reason, this study establishes *Hypothesis 1* as follow:

- *Hypothesis 1*: material classification performance can be improved by leveraging laser intensity values for training neural network architecture.

To mitigate the impact of ray distance and incident angle, this study normalizes the intensity by Equation 2.

$$I_{norm} = I_{raw} \left(\frac{R_{act}}{R_{ref}}\right)^2 \left(\frac{1}{cos\ (\theta)}\right) \tag{2}$$

where R_{ref} is set to 10 m in this study, which means the corrected intensities are equal to the intensity measured at 10 m.

5.1.2 PointNet++

According to (Qi, Yi, et al. 2017), PointNet++ complements the weakness of PointNet, which does not properly learn local structure by composing a hierarchical network with density adaptive PointNet layers. The hierarchical neural network applies PointNet recursively on a nested partitioning of the input dataset. That is, PointNet++ repeatedly performs PointNet in a partitioned area to learn local features to extract global features from the hierarchical structure. In this way, the classification performance in large scale point clouds can be improved. For this reason, this study employs PointNet++ as a baseline architecture. Figure 7 is the PointNet++ architecture.

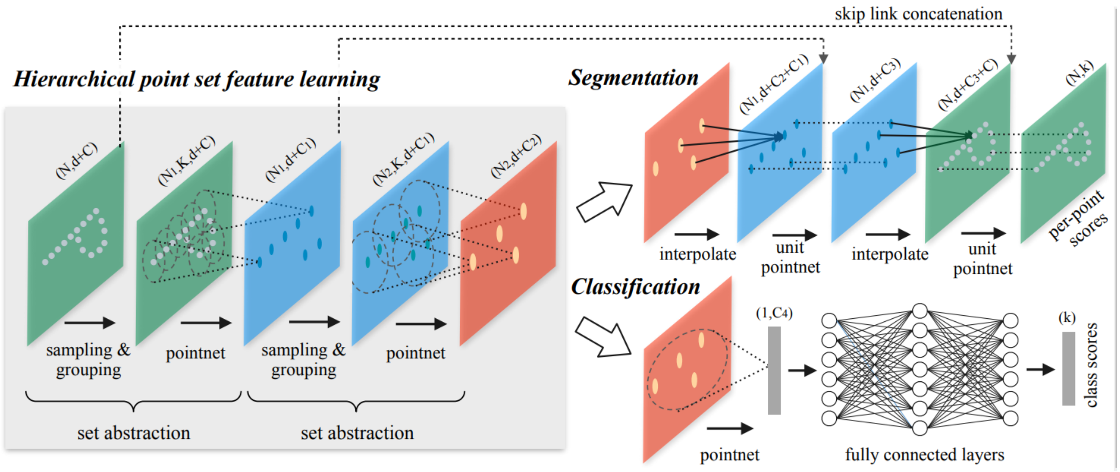


Figure 7. PointNet++ architecture (Qi, Yi, et al. 2017)

The PointNet++ architecture consists of three primary layers as follows:

- Sampling layer: selects a set of points from input points, which defines the centroid of local regions
- Grouping layer: constructs local region sets by finding neighboring points around the centroids
- PointNet layer: uses a mini-PointNet to encode local region patterns into feature vectors

To be sure of the validity of using PointNet++ as a baseline network architecture, this study conducted a performance comparison between PointNet and PointNet++. The comparative analysis is conducted with a material dataset containing three types of material, concrete, steel, and masonry. As shown in Table 5. Classification performance comparison between PointNet and PointNet++Table 5, PointNet++ showed about 1.5% higher classification performance than PointNet. Based on the result, this study employs PointNet++ as the baseline architecture for the material and object classification.

Table 5. Classification performance comparison between PointNet and PointNet++

PointNet			**PointNet++**		
Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
0.90141	0.88215	0.89168	0.92581	0.88846	0.90675

5.1.3 Dataset for architecture training

As aforementioned, this study additionally feeds the laser intensity values and material indexes for the deep learning-based material classification and object classification, respectively, so that the n-point input for the PointNet++ architecture has (n × 7) dimensions (XYZ+RGB+Intensity). The point cloud data was collected from a commercial terrestrial laser scanner.

Table 6. The laser scanner's specification

Categories	Specs.
Beamwidth	2.25 mm + 2 × 0.011°
Ranging error	± 2 mm (10 – 25 m)
Maximum range	330 m (used in this study: 100m)
Field-of-view (vertic./horiz.)	300° / 360°
Horizontal resolution	0.035° (6mm @ 10m)
Vertical resolution	0.035° (6mm @ 10m)

For training the network architecture for material and object classification, the ground truth-labeled datasets were constructed based on the point clouds collected from two actual construction sites (Sites 1 and 2) and an artificially built concrete building (Site 3). The point cloud data was collected additional laser-scanned point cloud data at a different location (Site 4), which was then used for validation of the proposed approach. Table 7 shows the description of the datasets, and Figure 8 depicts the categories of the material and object datasets.

Table 7. The description of the material and object datasets

Purpose	**Training**			**Validation**
Data collection location	Site 1	Site 2	Site 3	Site 4
# scans	6	5	7	3
# points per scan	1,400,000	1,100,000	1,400,000	1,400,000
# material types	6	5	5	5
# object types	12	10	8	8
Maximum raidus	100 m	100 m	100 m	100 m
Point Distance (mm/10m)	6.136	7.670	6.136	6.136
Parameters	- Material: x, y, z, R, G, B, intensity, mtr label - Object: x, y, z, R, G, B, mtr label, obj label			

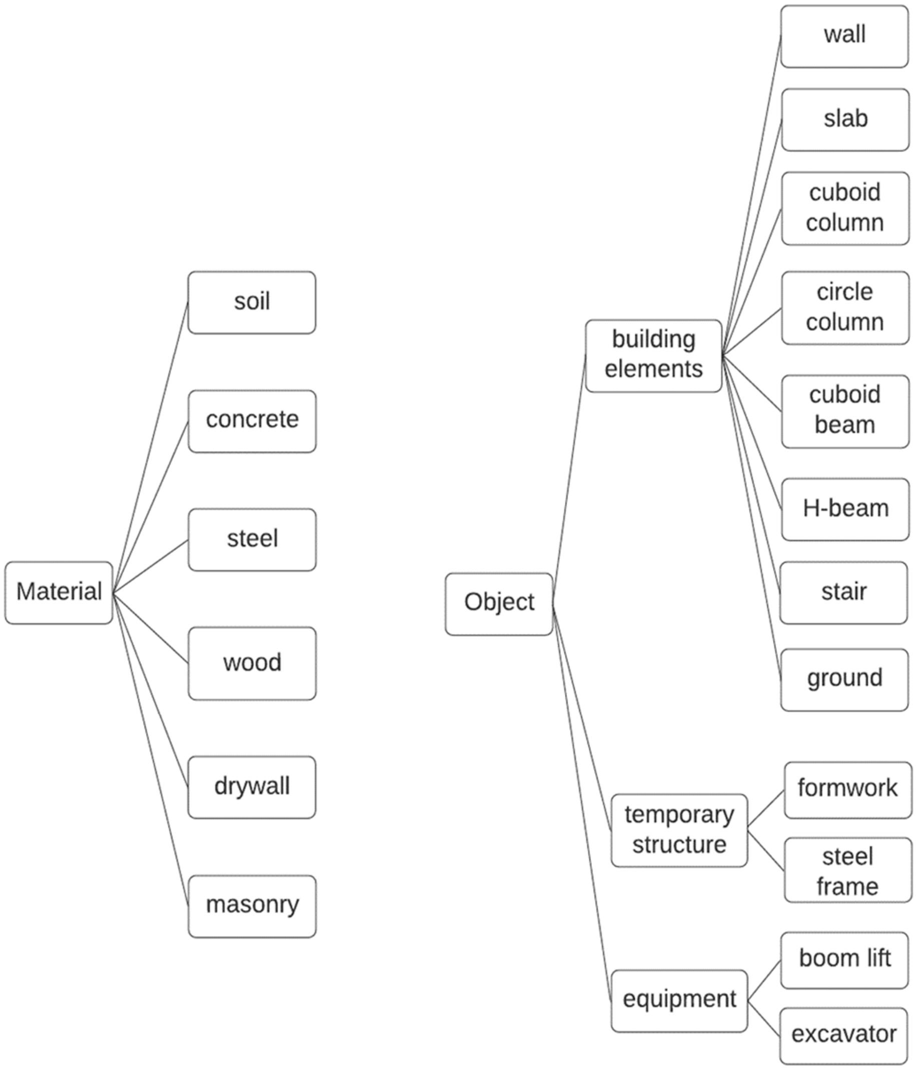


Figure 8. The categories of the material and object datasets

5.1.4 Material classification performance evaluation

The performance of the proposed is evaluated by the F1 score. The F1 score is the harmonic mean between precision and recall. It tells how precise, robust the classifier is. The range for the F1 score is [0, 1]. Mathematically, it can be expressed by Equation 3.

$$F1 = 2 \times \left(\frac{1}{\frac{1}{precision} + \frac{1}{recall}} \right) \tag{3}$$

where, the precision is the number of correct positive results divided by the number of positive results predicted by the classifier, and the recall is the number of correct positive results divided by the number of all relevant samples.

$$Precision = \left(\frac{TP}{TP + FP} \right) \tag{4}$$

$$Recall = \left(\frac{TP}{TP + FN} \right) \tag{5}$$

- TP: correctly labeled points
- TN: correctly unlabeled points
- FP: incorrectly labeled points
- FN: incorrectly unlabeled points

To test classification performance for all materials defined in PCIM, the evaluation is conducted with three outdoor point clouds from construction sites and one indoor point cloud from an existing building. Figure 9 shows the results of material classification from laser-scanned point cloud data using laser intensity.

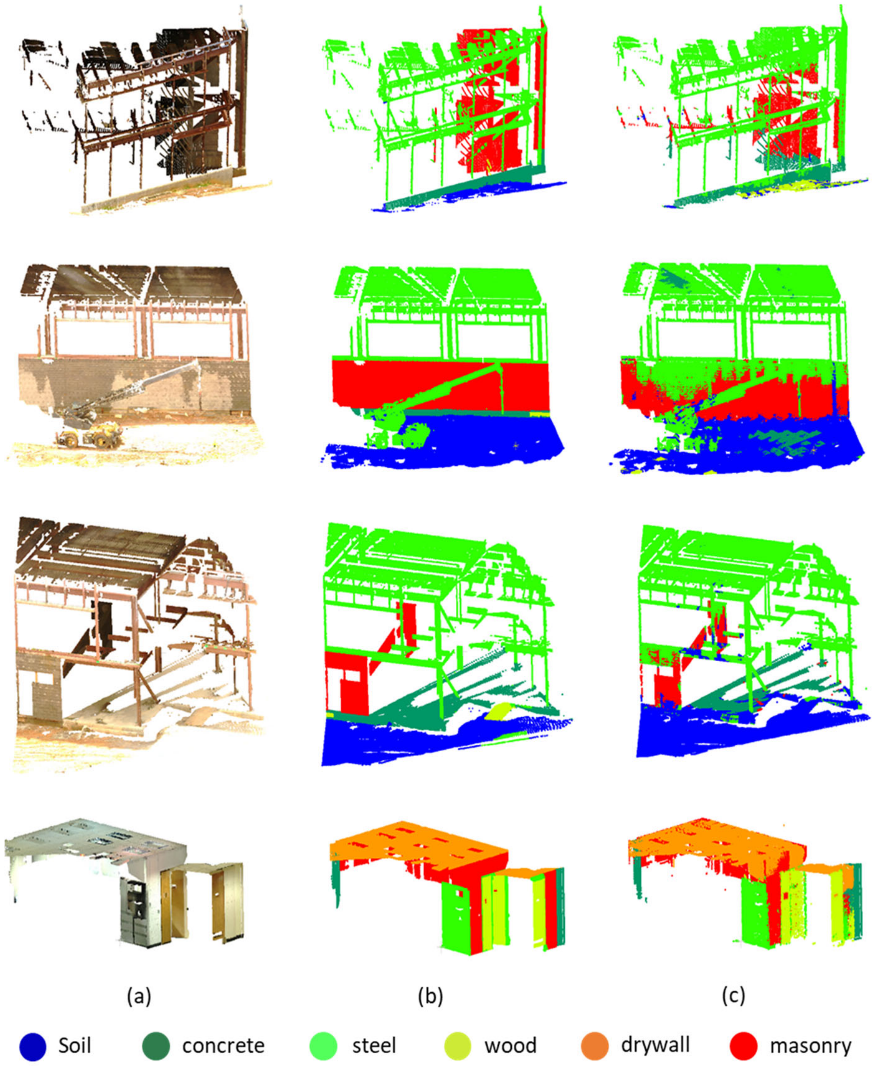


Figure 9. Material classification results. (a): the original point cloud, (b): the ground truth labeled point cloud, and (c): segmentation results

In most cases, the material was well segmented with the proposed classification approach using laser intensity. However, some wet surfaces caused classification errors, as shown in Figure 10. It is because the reflective strength of the laser can be reduced in wet areas. This errorcan be reduced by intensity correction with the surface condition (Lichti and Harvey 2002). In addition, classification errors existed in places such as exposed steel plates where diffraction of light occurs.

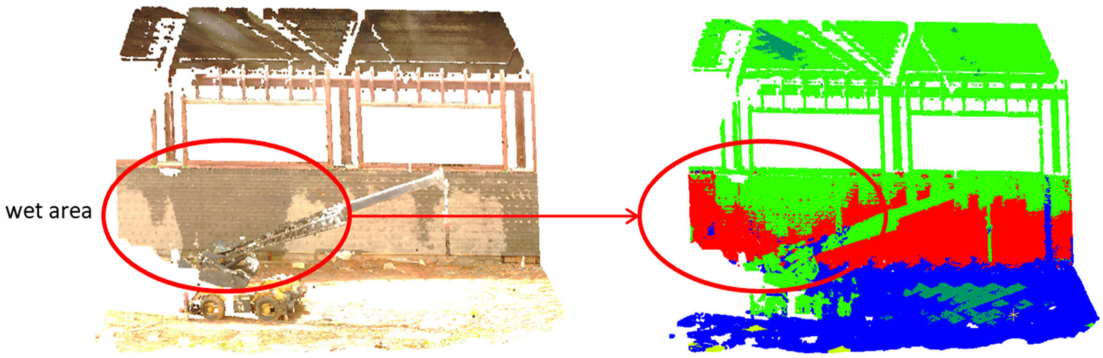


Figure 10. The classification errors in wet surfaces

To validate *Hypothesis 1*, “material classification performance can be improved by adding laser intensity values to feature learning input parameters,” this study compared the performance of material classification in point cloud data with and without laser intensity values. The comparative analysis is conducted by comparing the F1 score, and the F1 scores are calculated in point level. Table 3 shows the results of the comparative analysis. The F1 score increased by up to 15% to 3% for all material types except soil. Based on these results, the hypothesis that adding laser intensity to input parameters of deep neural network architecture training can improve the material classification in point cloud data is verified.

Table 8. Comparison of the material classification performance

	Without Intensity			With Intensity		
Material type	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
soil	0.92494	0.88865	0.90643	0.95988	0.85846	0.90634
concrete	0.70982	0.74015	0.72397	0.92388	0.84046	0.87499
steel	0.90742	0.84574	0.87284	0.93574	0.91219	0.92327
wood	0.96882	0.71628	0.79921	0.80222	0.93177	0.86209
drywall	0.88888	0.82131	0.85376	0.86518	0.90542	0.88484
masonry	0.70809	0.90818	0.79575	0.86500	0.80053	0.83152
Average	0.85133	0.82005	0.82533	0.89198	0.87481	0.88051

5.2 Construction Object Classification

5.2.1 Hierarchical deep learning-based object

Similar to the material classification process, object classification also utilizes PointNet++ with expanded input channels. The object classification process feeds material information labeled in the material classification process rather than intensity values. The indexed material information can provide useful contextual clues for classifying construction objects because particular building objects have specific or unique materials; for example, an H-beam is mostly composed of steel. This study, therefore, establishes *Hypothesis 2* as follow:

- *Hypothesis 2*: construction object classification performance can be improved by adding material information to input parameters for feature learning.

By adding the material information to the input, the n-point input for the PointNet++ architecture has (n × 7) dimensions (XYZ+RGB+Material index #), as seen in Figure 11. With the seven channels of input data, PointNet++ performs semantic segmentation, and the segmented points are then clustered into individual object units with the Euclidean-distance based clustering method (Rusu and Cousins 2011). The Euclidean cluster extraction method clusters neighboring points with an individual object level based on the Euclidean geometry, similar to the region growing segmentation method. The steps of the algorithm are as follows:

1. generate a kd-tree interpretation for the input dataset P;
2. set up a blank list of clusters C, and a queue of the points that need to be verified Q;

3. conduct the following steps with every point $p_i \in P$,:
    a. add p_i to the current queue Q;
    b. for all points $p_i \in Q$ perform:
        i. find for the set P_i^k of neighbor points of p_i in a sphere of radius $r < d_{th}$;
        ii. for all neighbor points $p_i^k \in P_i^k$, check if the point has been processed and if it is not processed, add it to Q;
    c. if the all point lists in Q has been processed, add Q to the list of the cluster C, and change Q to a blank list
4. If all points satisfying $p_i \in P$ have been processed and are part of the list of point clusters C, the roof is closed.

After the clustering process, the module assigns a unique index, 'objId,' to each clustered object. The training and validation datasets for the object classification include seven types of building objects, as shown in Table 7.

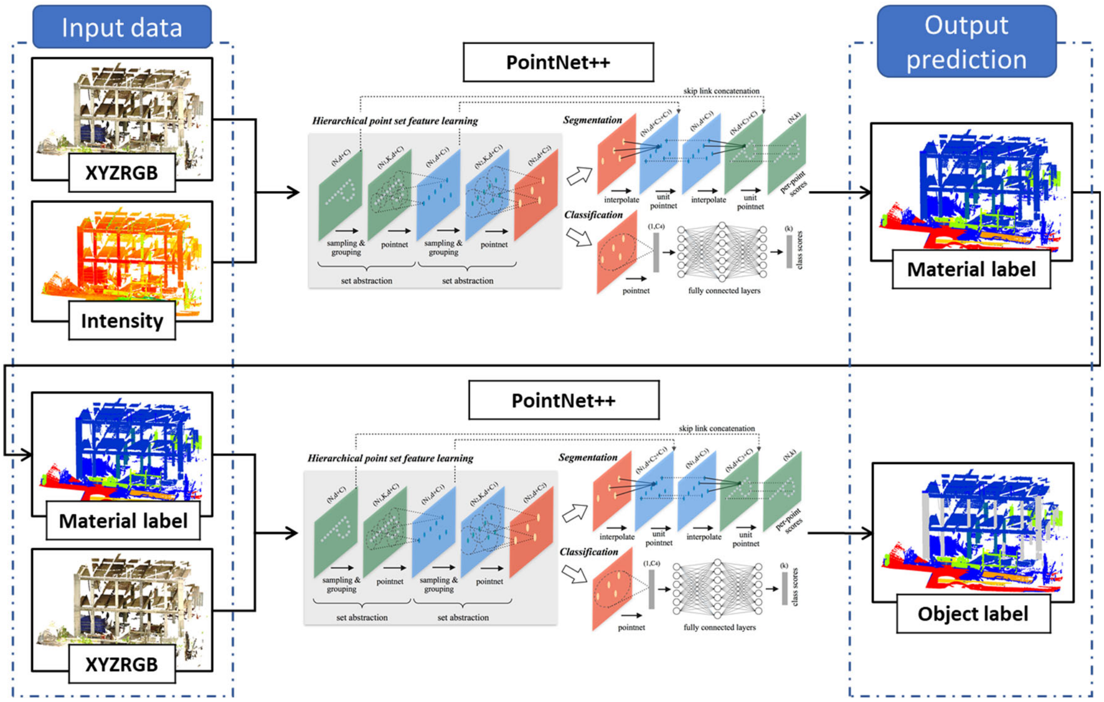


Figure 11. Hierarchical deep learning-based object classification

5.2.2 Object classification performance evaluation

To test construction object classification performance, the evaluation is conducted with three outdoor point clouds from construction sites. Figure 12 shows the construction object classification results from laser-scanned point cloud data using material information. A total of eight construction objects were detected in the evaluation datasets.

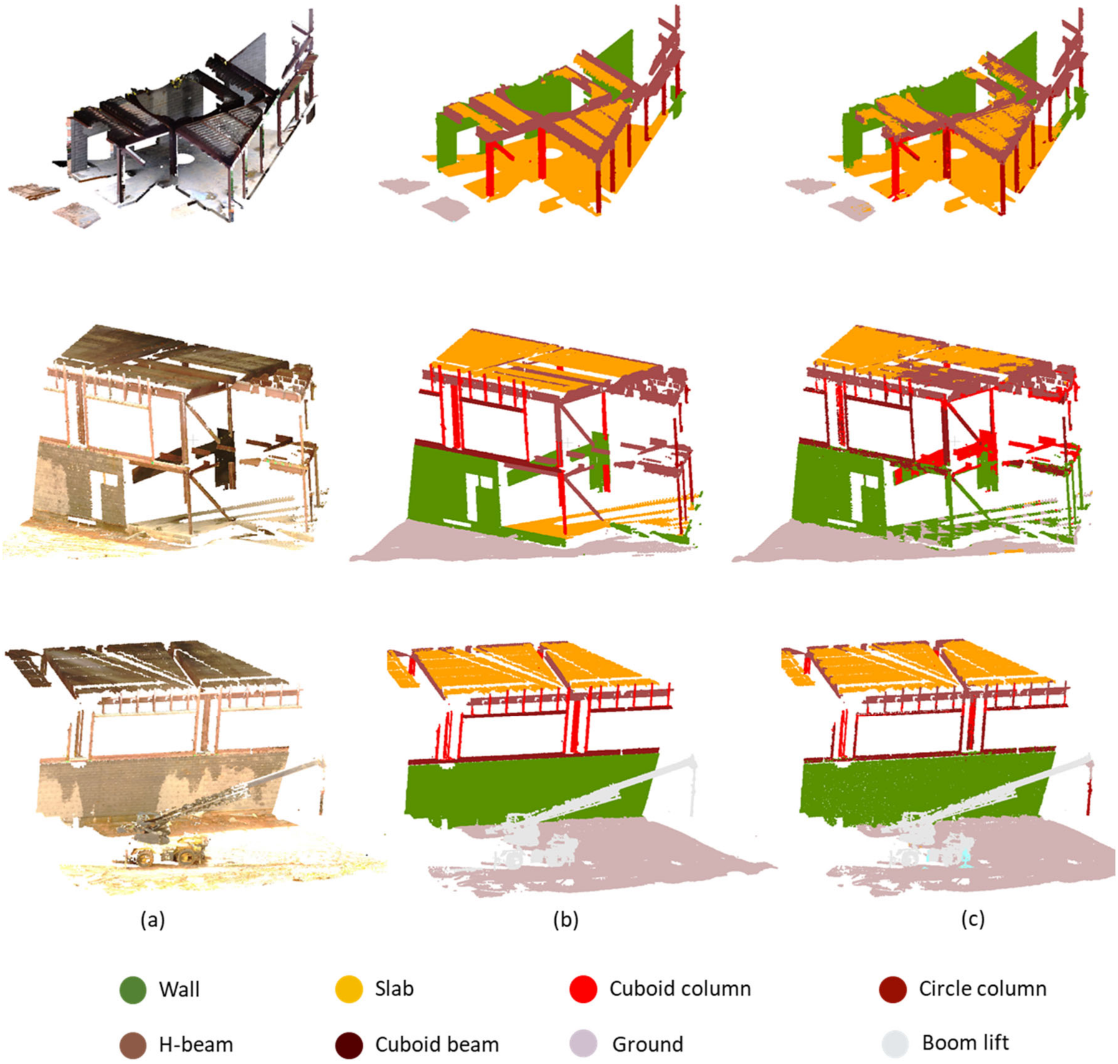


Figure 12. Construction object classification results. (a): the original point cloud, (b): the ground truth labeled point cloud, and (c): segmentation results

To validate *Hypothesis 2*, "using material information as an input parameter for architecture learning can improve the construction object classification performance," this study compared the performance of construction object classification in point cloud data with and without material information. Similar to material classification, the comparative analysis is conducted by comparing the F1 score, and the F1 scores are calculated in point level. Table 9 shows the results of the comparative analysis. The F1 score increased by up to 7% when using the material index. Based on these results, the hypothesis that using material information for an input parameter of neural network architecture training can improve the object classification in point cloud data is verified.

Table 9. Comparison of the construction object classification performance

	Without material			With material		
Material type	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Wall	85.5481	86.7940	86.16655	84.17940	92.55000	88.16647
Slab	86.1007	86.1111	86.10590	88.87360	89.26340	89.06807
Cuboid column	80.5099	88.8421	84.47103	91.37080	90.88510	91.12730
Circle column	81.4085	85.2744	83.29662	84.09450	88.38290	86.18539
Cuboid beam	80.8764	83.0203	81.93433	87.10920	87.80620	87.45631
H-beam	89.8702	89.4426	89.65589	86.22500	88.83930	87.51263
Ground	82.4527	91.6011	86.78648	86.94260	85.15620	86.04013
Boomlift	90.0274	81.9480	85.79791	91.36100	88.74080	90.03184
Average	84.5992	86.6292	85.5268	87.5195	88.9530	88.1985

5.3 Spatial Reasoning

After the point cloud segmentation processes, the PCIM framework carries out spatial reasoning with point density histogram approaches. The spatial reasoning module first determines building boundary proposal with XY-histogram and then finds main floor proposals with Z-histogram. Figure 13 is a flowchart representing the spatial reasoning module.

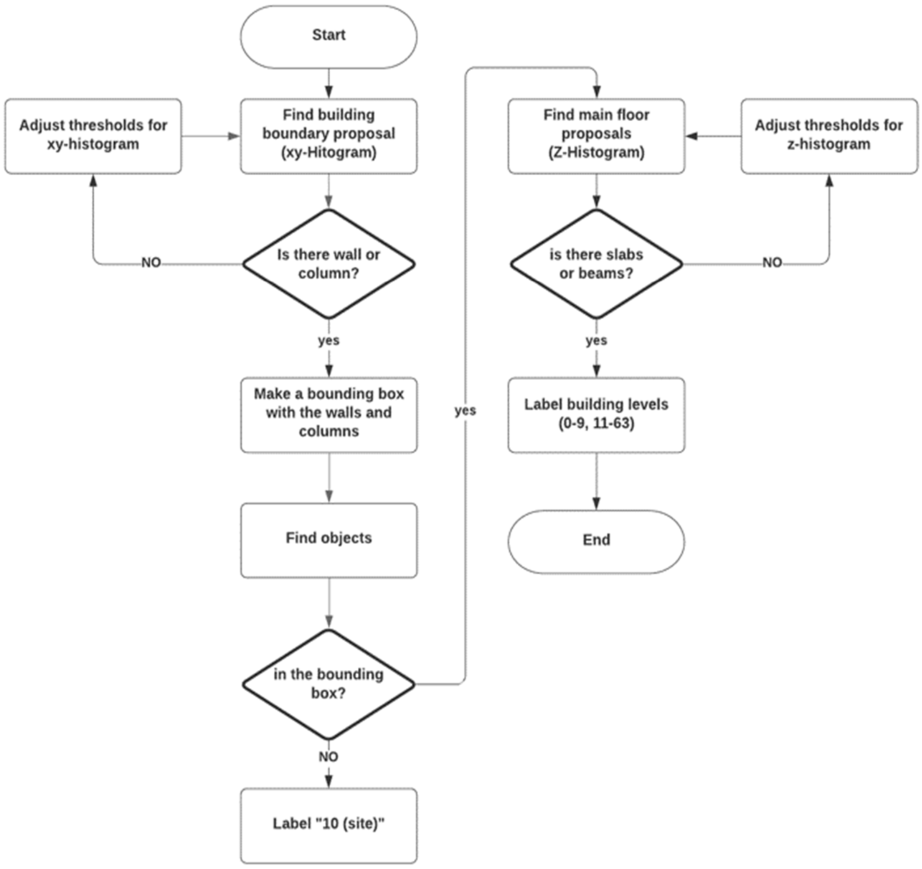


Figure 13. Flowchart of the spatial reasoning module in PCIM

5.3.1 XY-histogram based building boundary prediction

This method is derived from the premise that the building boundary is made of vertical components such as walls and columns, causing a high density of points in the x-y plane. The module determines building boundary proposals from the XY-histogram by searching the outermost peaks with a peak finding algorithm, as seen in Figure 14.

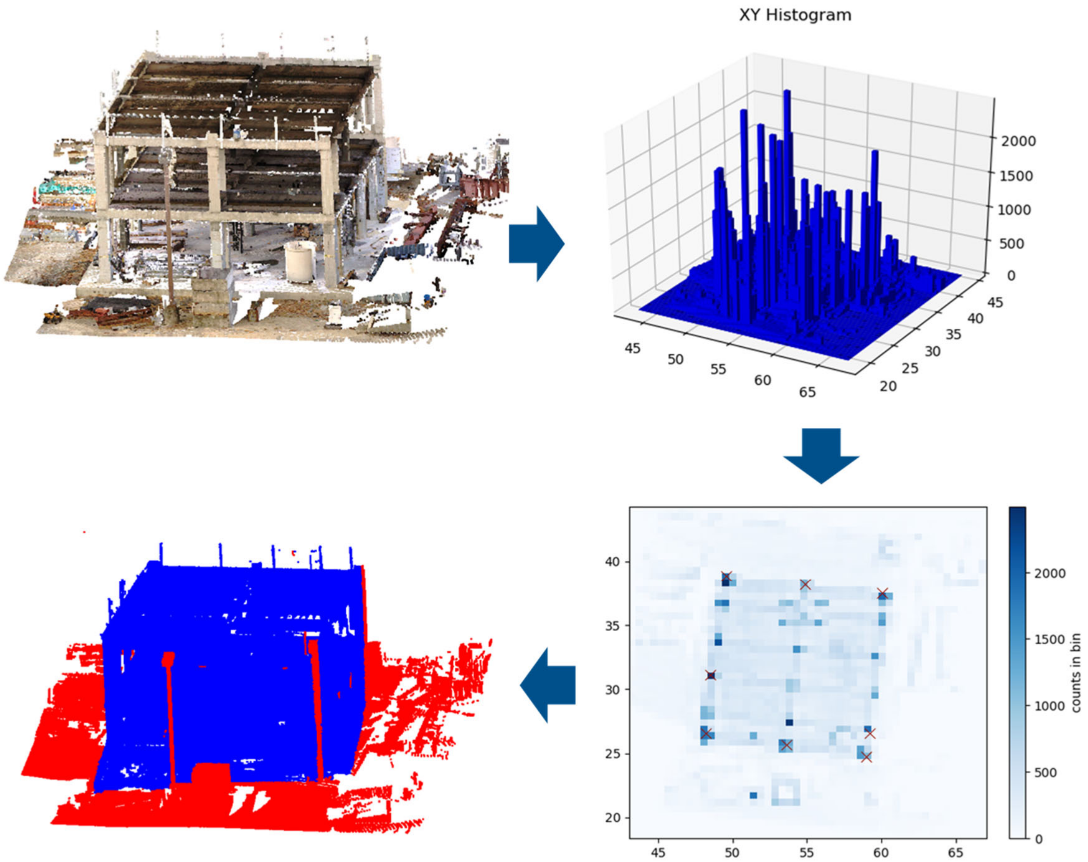


Figure 14. XY-histogram-based building boundary recognition

To find the multiple peaks in bivariate data, this research employs the kernel density estimator. The kernel estimator for the cumulative distribution function (cdf), for any real values of x, is given by Equation 6 and 7.

$$F_h(x) = \frac{1}{n}\sum_{i=1}^{n} G\left(\frac{x - x_i}{h}\right) \tag{6}$$

$$G(x) = \int_{-\infty}^{x} K(t)dt \tag{7}$$

where, $x_1, x_2, \cdots, x_n$ are random samples from an unknown distribution, n is the sample size, h is the bandwidth, and $K(t)$ is the kernel smoothing fuction. Once the peaks are estimated, the module finds objects in the peaks. If walls and columns classified at the classification module exist in the building boundary proposal, the module makes a bounding box with the walls and columns. And then, if the objects are not in the boundary line, this framework labels 'Site (#10)' to the objects.

5.3.2 Z-histogram based building level prediction

Regarding the points in the building boundary line, this framework finds the main floor proposal with a z-histogram method. Similar to the x-y histogram, the z-histogram method determines the main floor proposal based on the base concept that the point density in the z-axis where the main floor is located should be higher than elsewhere, as shown in

Figure 15. Based on the main floor proposals, the spatial reasoning module assigned building level indexes to each object, e.g., ‘11’ or ’12,’ which means 1F and 2F, respectively.

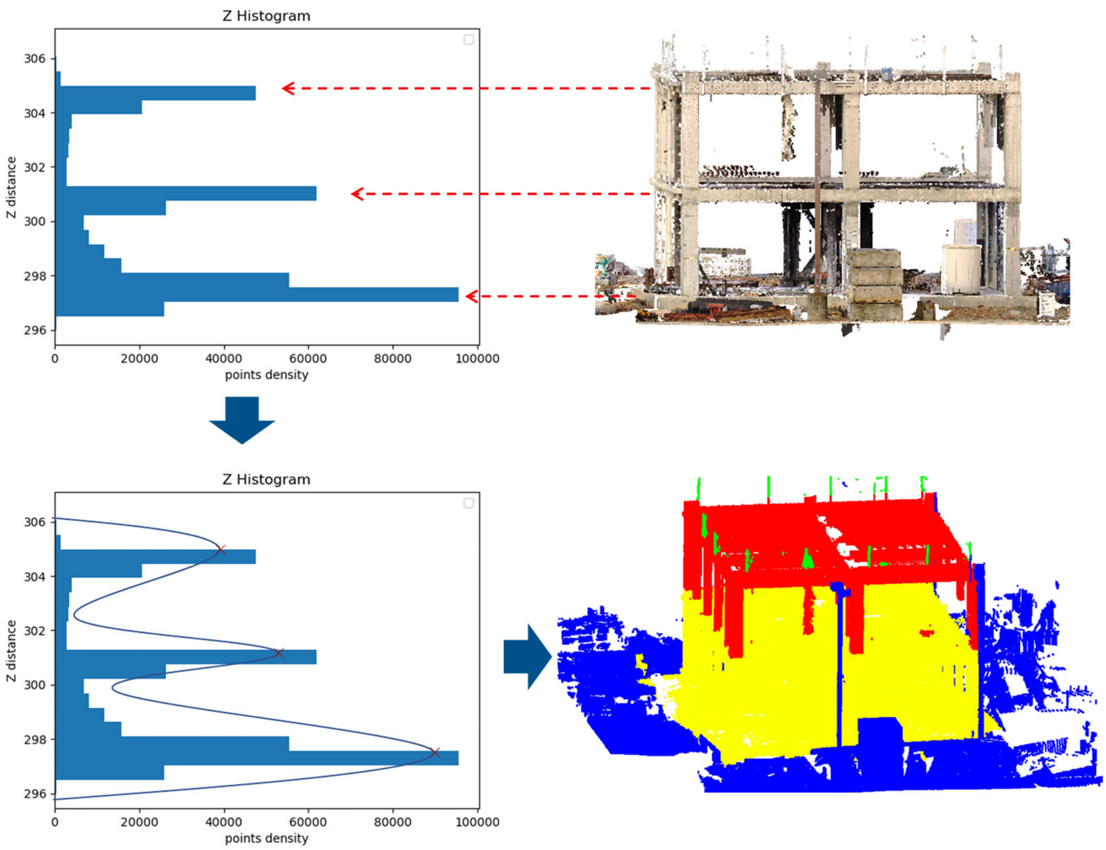


Figure 15. Z-histogram-based building level (floors) recognition

5.3.3 Point Cloud labeling

At the end of all the classification processes, a point cloud data finally generated with all the information labeled, as shown in Figure 16. The classified information is stored in each column at an object unit with an integer format, and individual objects are given its own IDs to distinguish and count them. The intensity values were used only for material classification, and it is eliminated after the material classification to reduce the file size of the point cloud data because it is no longer meaningful data for object classification and spatial reasoning. With the labeled point cloud data, PCIM parser organizes the data with an object-oriented data structure based on pre-defined PCIM data schema. The PCIM data schema and data parsing processes are demonstrated in CHAPTER 6.

PointId	x	y	z	R	G	B	ObjectId	object	material	shape	location
1	26.494417	7.443369	6.181198	188	170	146	0	31	5	8	12
2	26.441339	7.585955	6.138519	119	93	56	0	31	5	8	12
3	26.427677	7.571283	6.310195	188	153	121	0	31	5	8	12
4	26.605112	7.570482	6.317565	163	125	89	0	31	5	8	12
5	27.026114	7.557001	6.156677	185	172	138	0	31	5	8	12
6	27.103458	7.639324	6.11885	219	202	159	0	31	5	8	12
7	27.048269	7.580845	6.145996	182	167	134	0	31	5	8	12
8	27.069572	7.60342	6.183868	178	163	134	0	31	5	8	12
9	27.102451	7.638956	6.167816	178	163	134	0	31	5	8	12
10	26.998362	7.527403	6.239471	145	126	94	0	31	5	8	12
11	27.00362	7.532156	6.248672	172	157	128	0	31	5	8	12
12	27.013996	7.544271	6.227157	173	159	130	0	31	5	8	12
13	27.002108	7.531206	6.330581	164	151	107	0	31	5	8	12
14	26.996977	7.52486	6.345367	145	130	91	0	31	5	8	12
15	27.068666	7.602467	6.240936	177	162	133	0	31	5	8	12
16	27.091311	7.627207	6.197738	173	162	130	0	31	5	8	12
17	26.915012	7.597761	6.38356	122	104	66	0	31	5	8	12
18	27.056644	7.589697	6.303573	176	161	128	0	31	5	8	12
19	27.067045	7.601286	6.3311	180	164	128	0	31	5	8	12
20	27.100046	7.635221	6.339218	198	185	150	0	31	5	8	12
21	27.033672	7.565565	6.355407	210	194	160	0	31	5	8	12
22	27.099478	7.634809	6.372222	203	188	155	0	31	5	8	12
23	26.336365	8.136248	6.210693	125	82	47	0	31	5	8	12
24	26.334009	8.138901	6.254928	144	94	61	0	31	5	8	12

Figure 16. Information labeled point cloud after classification processes

CHAPTER 6. PCIM DATA SCHEMA

This chapter presents a prototypical design of PCIM data schema and validates the integrated PCIM framework as follows:

1. Data Schema Development
    a. Entity inheritance
    b. Data parser and viewer
2. Validation of PCIM framework from a Case Study
    a. Test set-up
    b. Classification module
    c. Data parsing module
    d. PCIM viewer
    e. Comparison between Scan-to-BIM and PCIM

Each of these steps is discussed in the following sections.

6.1 Data schema development

6.1.1 Entity inheritance

The PCIM data schema is designed to be compatible with IFC in terms of expressing building information by inheriting major data structures from IFC. However, since the primary purpose of PCIM is to represent the as-is condition of construction sites with reality measurements, the entities related to parametric modeling in IFC are excluded.

In addition, as PCIM is an automated information modeling framework in the current scope of the study, it only represents automatically detectable and recognizable entities. Figure 17 depicts the inheritance, relationship, and cardinality between entities in the PCIM data schema.

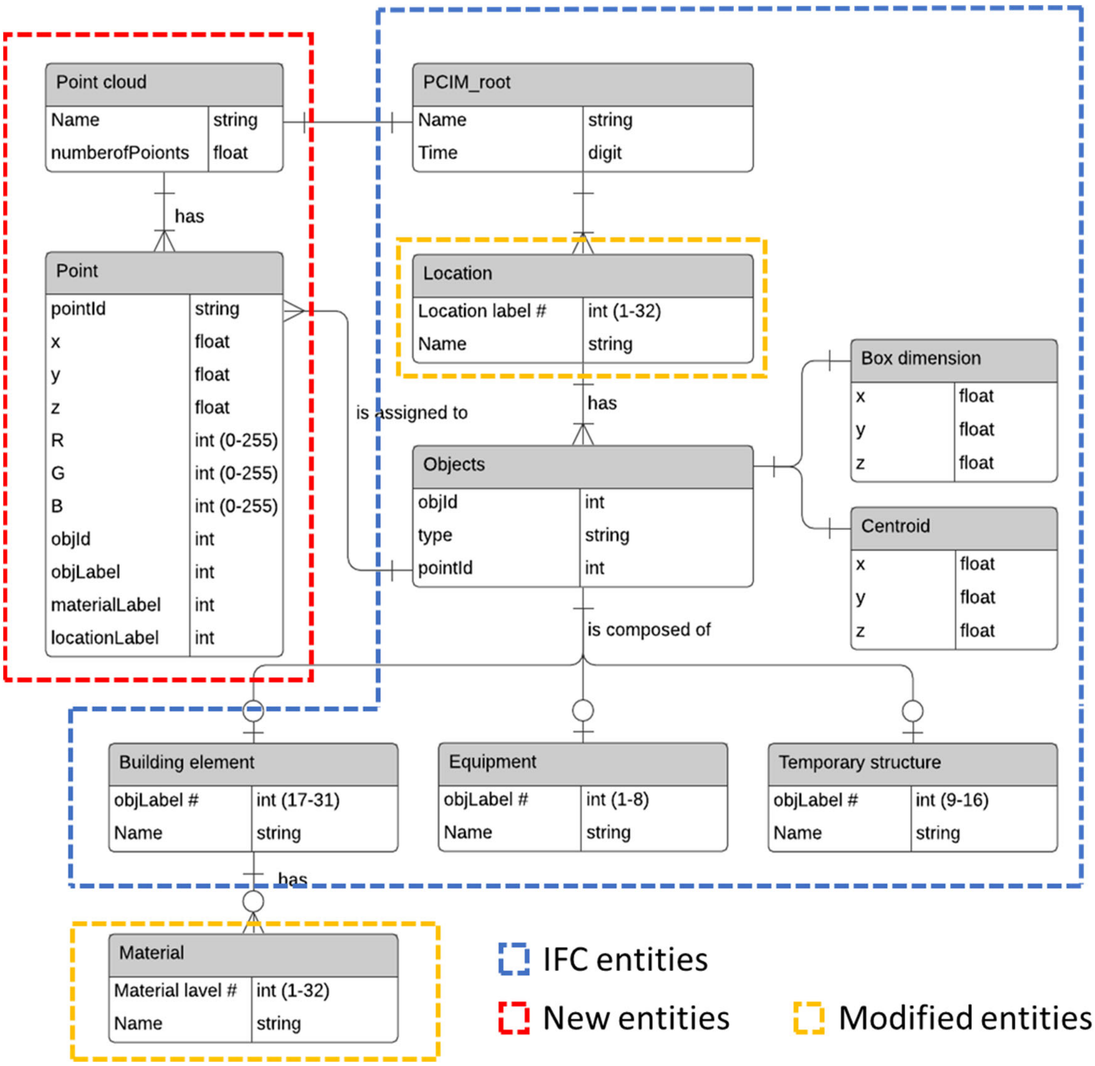


Figure 17. Entity-relationship diagram (ERD) of the PCIM data schema

PCIM data schema begins with PCIM_root describing the general information of the PCIM file. In the entity hierarchy, the 'Location' entity has one or more 'Object' entities. The 'Object' entity is composed of 'Building element,' 'Equipment,' and 'Temporary structure,' and among them, only the 'Building element' entity has a 'Material' entity. Moreover, one 'Point cloud' entity has millions of 'Point' entities containing all labeled information, and the 'Object' entity only brings the 'pointId' attribute from the 'Point' entities. The PCIM data schema also defines object and material label numbers as 'Objlabel Enum' and 'Material Enum.' The PCIM data schema is encoded by XSD (XML Schema Definition).

6.1.1.1 Elements in PCIM.xsd

As aforementioned, the first version of PCIM data schema is encoded by XSD, named PCIM.xsd. Figure 18 shows the 'object' element in PCIM.xsd. The 'object' element is similar to 'IfcElement' entity in terms of representing a tangible object. However, the 'object' element does not have child entities for parametric modeling and type declaration by the user because the element reflects the as-is status of objects. The types and shapes of the objects in PCIM are determined as detected in the raw point cloud data. Therefore, PCIM data schema just stores the 'pointId' from the segmented point cloud data rather than representing the surface shapes of the object. In this way, the PCIM data schema can be much lighter and simpler than IFC. Figure 19 is the definition of 'IfcElement' in IFC4.3 RC1. The definition of other elements are listed in APPENDIX A

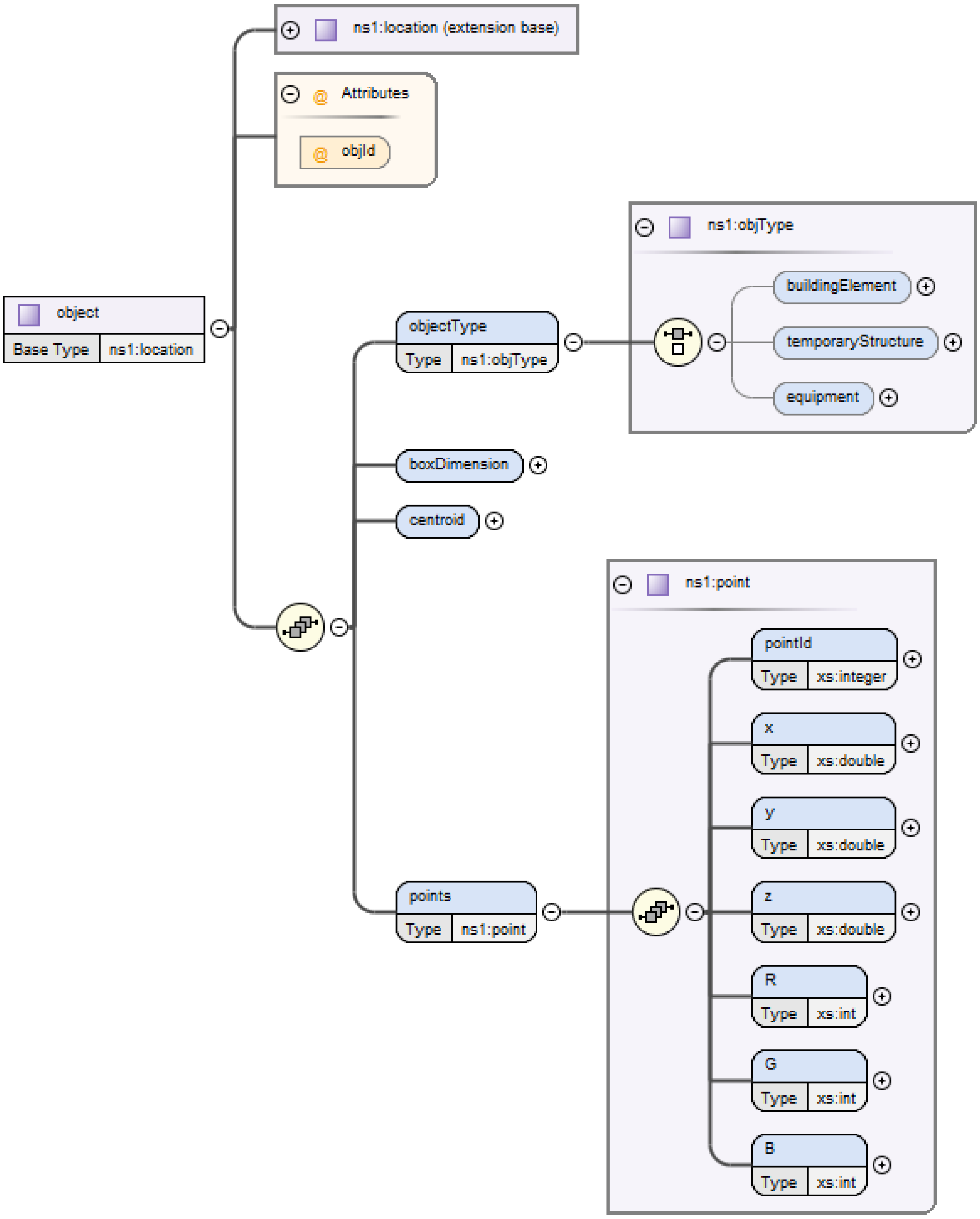


Figure 18. Definition of 'object' element in PCIM.xsd

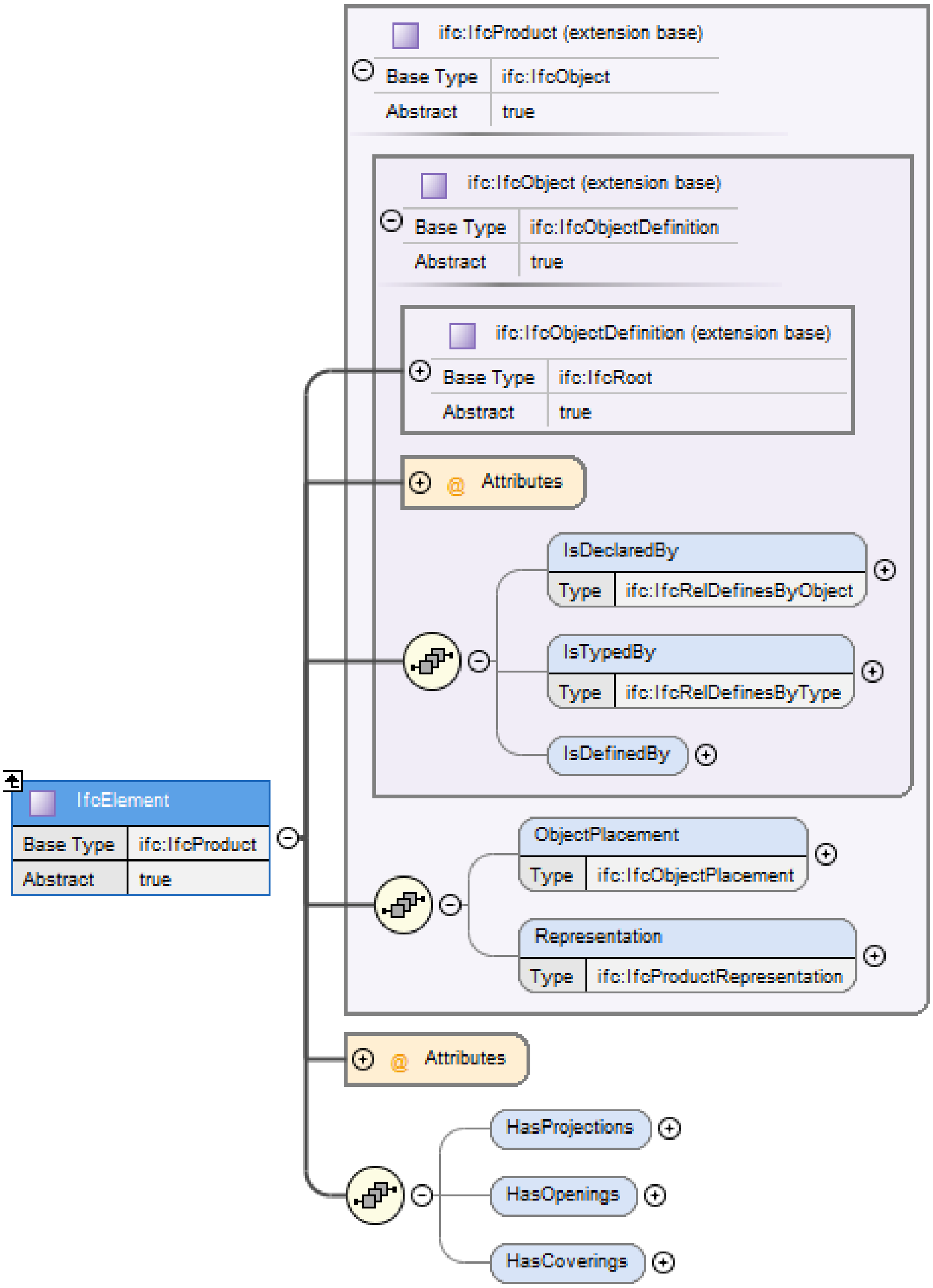


Figure 19. Definition of 'IfcElement' element in IFC4.3 RC1

6.1.2 PCIM data parser and viewer

This study implements a PCIM parser with the xml.etree.ElementTree module in Python providing functions to convert data sheets to XML files with a hierarchical structure. The PCIM parser categorizes raw data into object levels and generates bounding boxes. After that, the sub-modules of the PCIM parser compute the box-dimension and centroid of the bounding boxes and represent them with a 3D coordinate (x, y, z). The data parser finally converts the segmented point cloud data to an XML file based on the PCIM data schema. This study also develops a point cloud data viewer, named PCIM_viewer, implemented with the Point Processing Toolkit (pptk), an open-source Python library for visualizing and processing point cloud data with a simple user interface. The PCIM viewer enables to highlight only selected objects in the database (DB) tree. The viewer can also present the chosen object's properties, such as material, box dimensions, centroids, and color.

6.2 Validation of the Integrated PCIM Framework from a Case Study

6.2.1 Test set-up

To validate the feasibility of PCIM, this study conducted a case study with an actual building construction project on the campus of Georgia Institute of Technology. Figure 20 shows the field test site. The building consisted of various types of materials and building elements. There were two types of construction equipment, excavators and boom lifts, in the site. A total of six point clouds were collected by a terrestrial laser scanner (TLS) from the site. With the obtained point clouds, this study verified the feasibility of the integrated PCIM framework from the classification to the data parsing.

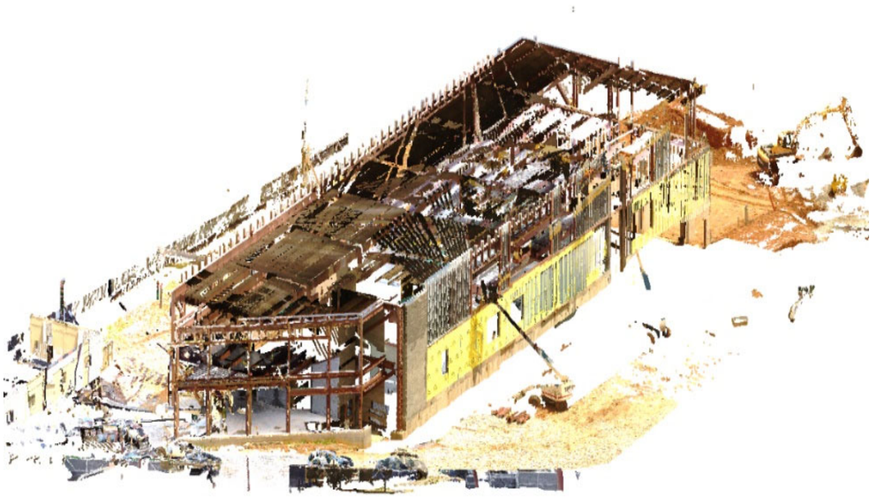


Figure 20. A registered point cloud model of a building under construction for PCIM framework validation

6.2.2 Classification module

The proposed PCIM framework conducted material classification first with the individual point cloud. Since the description of the classification processes and their classification performance is described in CHAPTER 5, this chapter only mentions the classification results and overall accuracy. Figure 21 depicts the material classification results using laser intensity. After the material classification process, the material labeled point clouds were registered, and with the registered point cloud, construction object classification was conducted.

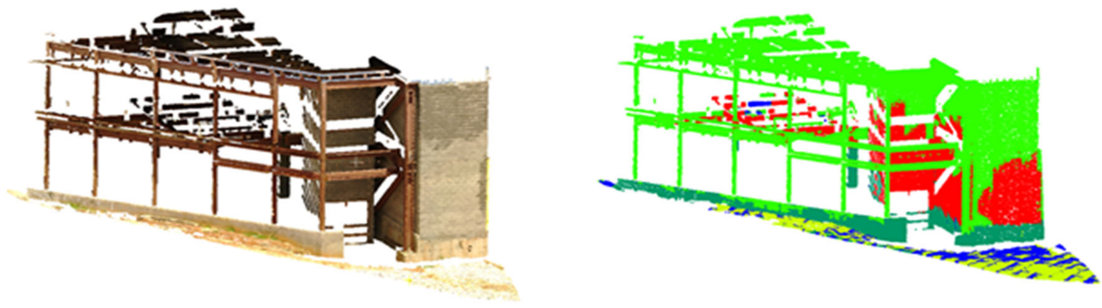


Figure 21. Material classification results for the test site

With the labeled material indexes, the PCIM framework performed object classification with a registered point cloud. Figure 22 and Table 10 describes the object classification results using the material index. The overall classification accuracy was about 88.71%. The accuracy was calculated for each point using Equation 8.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

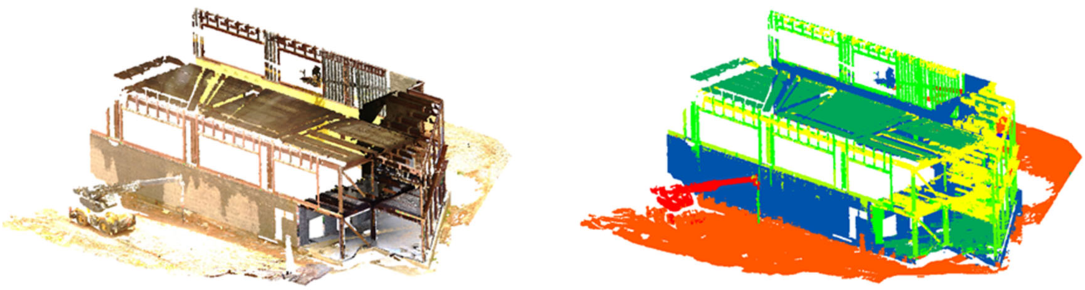


Figure 22. The results of the object classification using the material indexes

Table 10. Confusion matrix of the construction object classification for the case study

	Wall	Slab	Cuboid Column	H-Beam	Ground	Boom lift	Precision	Recall
Wall	0.909	0.051	0.000	0.026	0.015	0.000	0.909	0.998
Slab	0.000	0.991	0.000	0.000	0.009	0.000	0.991	0.922
Cuboid Column	0.000	0.000	0.610	0.367	0.000	0.023	0.610	0.837
H-Beam	0.001	0.017	0.035	0.947	0.000	0.000	0.947	0.882
Ground	0.000	0.004	0.000	0.000	0.993	0.002	0.993	0.993
Boom lift	0.000	0.000	0.000	0.000	0.016	0.984	0.984	0.963

After the sequential deep learning-based material and object classification, PCIM framework conducted the spatial reasoning process. Figure 23 and Figure 24 show the XY-histogram and Z-histogram, respectively. The module search columns and walls in the peak grids in XY-histogram and labeled “10”, which means “site,” to the objects found outside of the bounding box. Only “ground” and “boom lift” were found at the out of boundary line in this test. With the Z-histogram, the module found peak points with a kernel density estimator and determined building levels with the slabs and beams located in the peak points. The objects on the first floor were labeled “11” and objects on the second floor were labeled “12”.

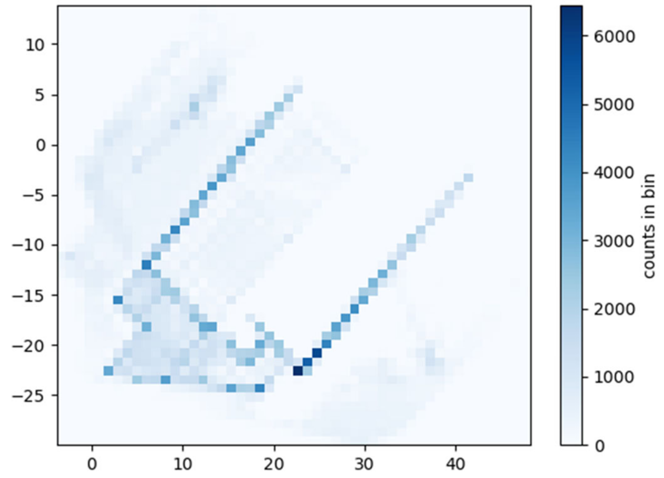


Figure 23. XY-histogram of the field test site

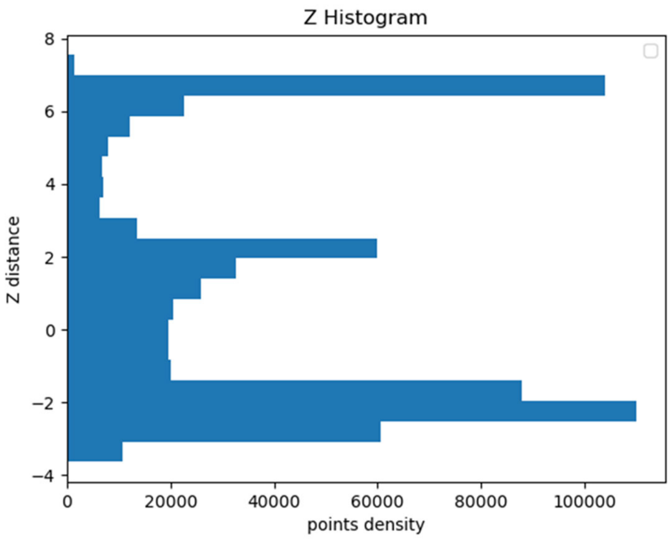


Figure 24. Z-histogram of the field test site

6.2.3 Data Parsing module

With the labeled point cloud data, PCIM parser clustered the semantically segmented point cloud into object levels and stored their properties such as location, material, box dimension, and centroid, and exported a human-readable XML file based on PCIM data schema (PCIM.xsd). Figure 26 and Figure 26 show a part of the XML file representing a wall at the test site.

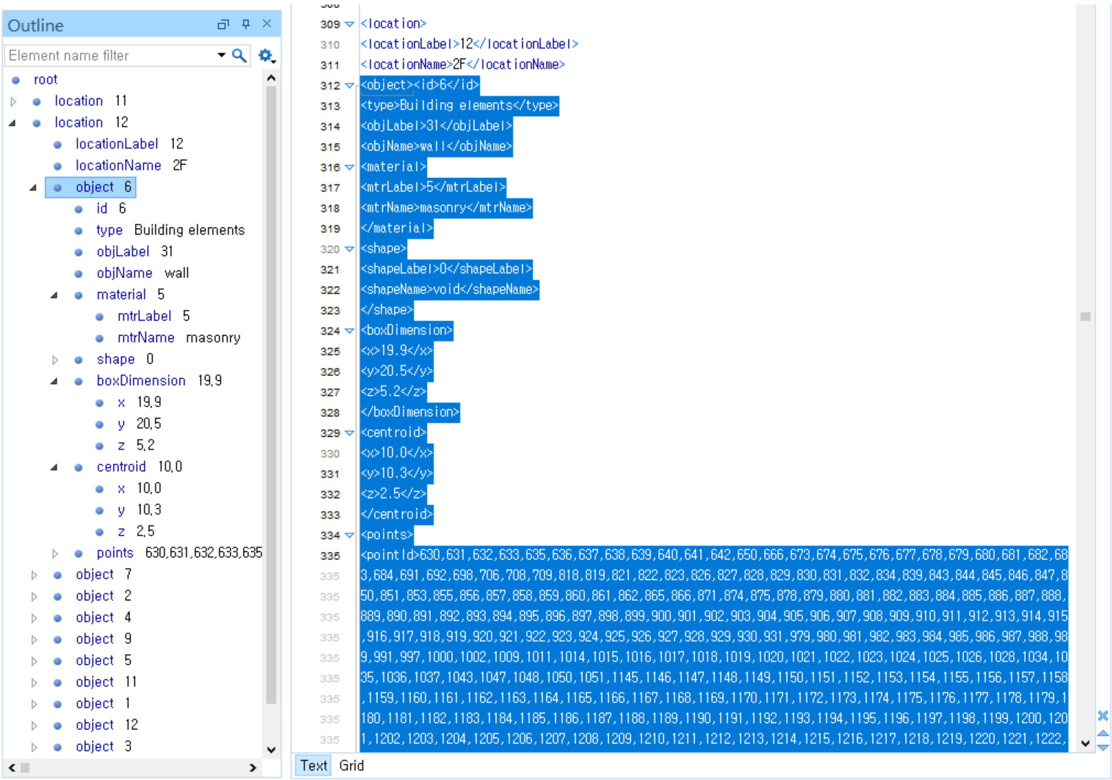
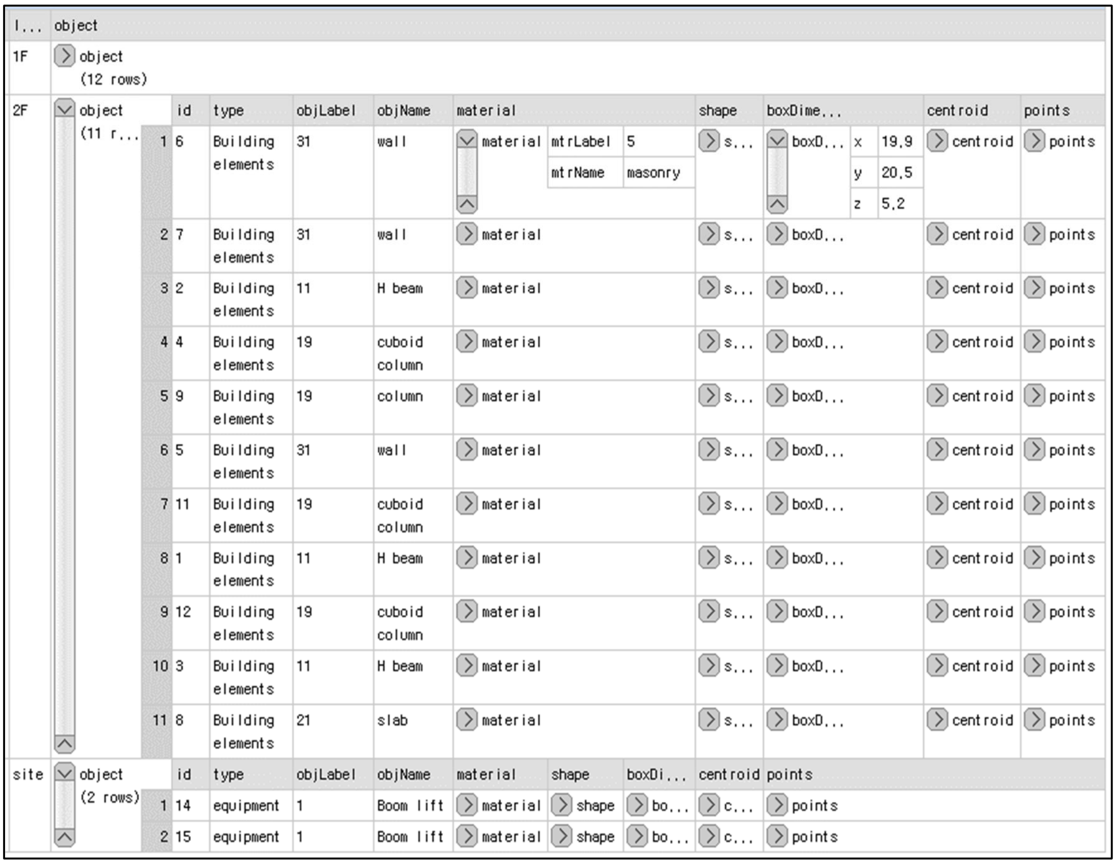


Figure 25. A part of the sample XML file generated by the PCIM framework for test site (Text view @ OxygenXML Viewer)



I...	object		id	type	objLabel	objName	material	shape	boxDime...	centroid	points
1F	object (12 rows)										
2F	object (11 r...	1	6	Building elements	31	wall	material: mtrLabel 5; mtrName masonry	s...	boxD...: x 19.9; y 20.5; z 5.2	centroid	points
		2	7	Building elements	31	wall	material	s...	boxD...	centroid	points
		3	2	Building elements	11	H beam	material	s...	boxD...	centroid	points
		4	4	Building elements	19	cuboid column	material	s...	boxD...	centroid	points
		5	9	Building elements	19	column	material	s...	boxD...	centroid	points
		6	5	Building elements	31	wall	material	s...	boxD...	centroid	points
		7	11	Building elements	19	cuboid column	material	s...	boxD...	centroid	points
		8	1	Building elements	11	H beam	material	s...	boxD...	centroid	points
		9	12	Building elements	19	cuboid column	material	s...	boxD...	centroid	points
		10	3	Building elements	11	H beam	material	s...	boxD...	centroid	points
		11	8	Building elements	21	slab	material	s...	boxD...	centroid	points
site	object (2 rows)	1	14	equipment	1	Boom lift	material	shape	bo...	c...	points
		2	15	equipment	1	Boom lift	material	shape	bo...	c...	points

Figure 26. A part of the sample XML file generated by the PCIM framework for the test site (Grid view @ OxygenXML Viewer)

6.2.4 PCIM viewer

The PCIM viewer implemented with the point processing tool kit (PPTK) library in Python visualizes the segmented point cloud with a DB tree that can choose hierarchically classified objects and highlight the selected items. Figure 27 depicts the PCIM viewer showing a chosen wall in the DB tree. The viewer also described the properties of the selected objects. The viewer not only visualizes the segmented point clouds but also is capable of adding a function to export the selected objects with .xml or other formats.

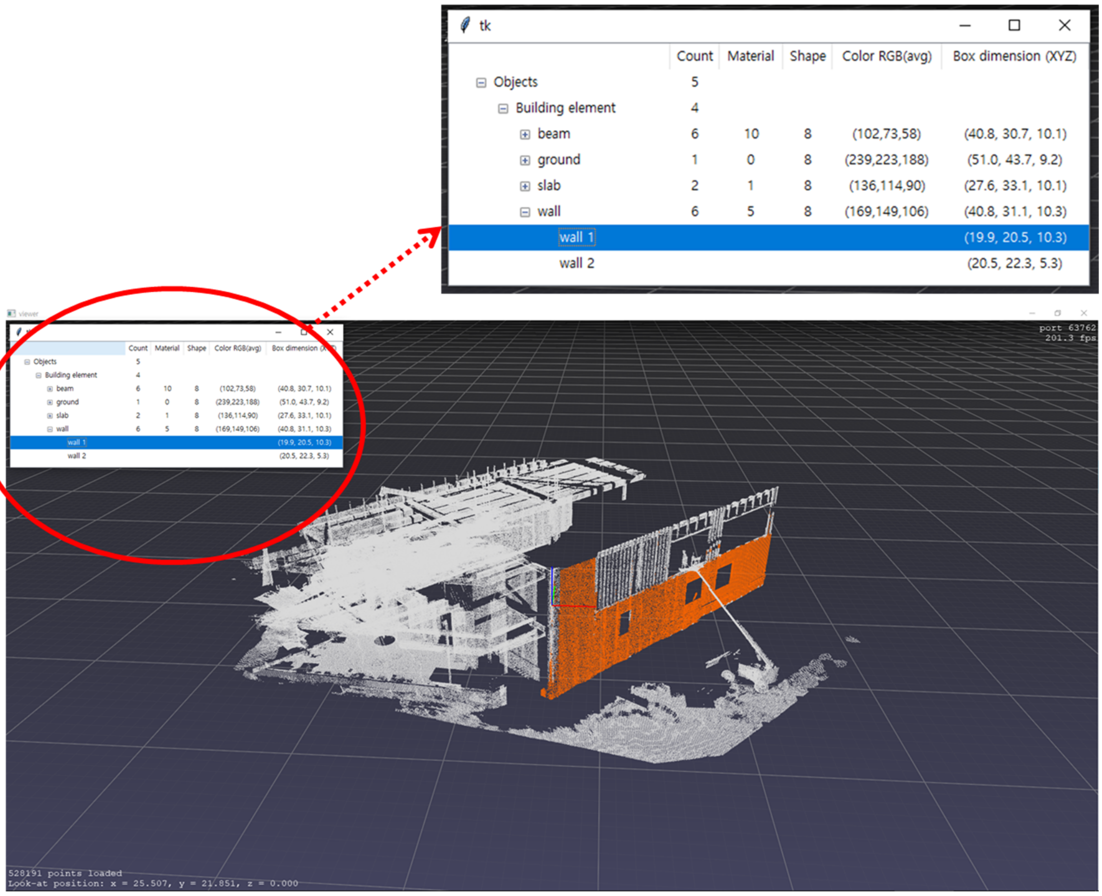


Figure 27. The PCIM viewer highlighting a selected wall in a DB tree

6.2.5 Comparison between Scan-to-BIM and PCIM

In this chapter, this study compares the PCIM framework to existing Scan-to-BIM practices. The comparison results are described in Table 11. Most of the currently commercialized Scan-to-BIM programs are implemented as semi-automation. In those programs, users find planar or cylindrical objects manually; then, the programs generate 3D models of the selected objects as seen in Figure 28. In most cases, region-growing-based plane detection algorithms were used to extract surfaces. Moreover, all the Scan-to-BIM softwares that are currently in service could not automatically classify the object's material and location, although some offer the ability to select material. Because of the manual processes in the commercial Scan-to-BIM programs, the processing time for as-built modeling depends on the user's expertise.

On the other hand, since the proposed PCIM framework is fully automated from classification to data parsing, users do not need to find the objects and input their properties manually. With the proposed hierarchical classification approach, PCIM can automatically detect objects, material, and locations and represent them in a human-readable format. Furthermore, since PCIM does not solidify the point cloud data, it can maintain the original point clouds' reality, such as true color, actual dimensions, or damages. As a result, the PCIM can be a more useful tool than Scan-to-BIM to understand and monitor the as-is conditions of construction sites.

Table 11. Comparison between PCIM and Scan-to-BIM

Category		**PCIM**	**Scan-to-BIM**	
			Scan to BIM (Revit Add-in)	**As-Built Modeler (Faro)**
Automation		Fully automatic	Semi-automatic	Semi-automatic
Solid modeling		X	O	O
Classification method		Deep learning	Region growing	N/A
Classification	**Building elements**	O (8 types)	Δ (only wall)	Δ (only wall)
	Temporary structure	O	X	X
	Equipment	O	X	X
	MEP	X	O	O
	Material	O	Manual declaration	X
	Location	O	Manual declaration	X
Processing Time		classification: 8 min Spatial reasoning: 2 min Data parsing: 2 min	Depend on user's expertise	Depend on user's expertise
Final product		XML file	Revit model	CAD model

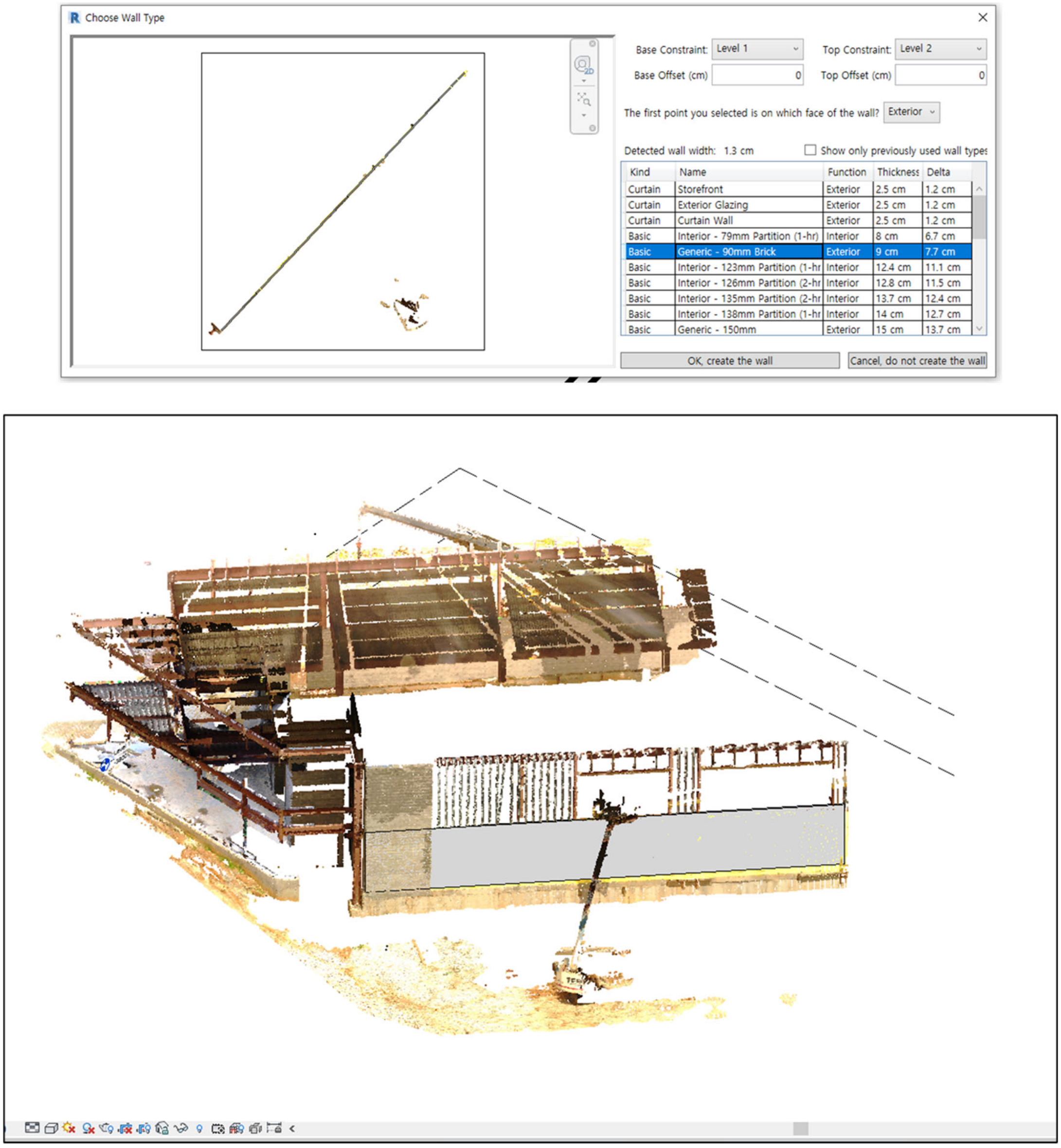


Figure 28. 3D model of a wall generated by Scan-to-BIM @ Revit

CHAPTER 7. CONCLUSION AND DISCUSSION

This chapter summarizes the research findings and discusses their potential contribution. The discussion of the limitations of the current research and future research follows.

7.1 Summary

The main goal of this dissertation is to develop artificial intelligence (AI)-driven information modeling framework for point cloud data. To achieve this goal, this study established two research questions and conducted several tasks to answer the questions as follows:

Research Question 1: *How can the construction objects and their properties be automatically extracted from the point cloud data?*

To seek the answer to the question, PointNet++, a state of the art approach for point cloud segmentation, was applied in this research. In addition, this study used laser intensity as an additional input parameter for training the deep neural network to improve the material classification. With the classified material information, the proposed framework performed object classification. The classification performance for both material and object could be enhanced by leveraging additional input parameters. Finally, the framework carried out spatial reasoning to identify the classified objects' location. For the spatial reasoning, this study proposed density histogram-based approaches. Consequently, the proposed hierarchical methods could classify construction objects and their properties simultaneously.

Research Question 2: *How can the classified semantic data be effectively managed and represented in terms of data interoperability?*

To manage the detected semantic information from the classification processes, this study developed an object-oriented data schema for point cloud data, named PCIM schema. In the PCIM data schema, all recognizable entities in point cloud data are pre-defined. Several definitions and relationships of entities in the PCIM data schema was derived from that of IFC to promote interoperability, but PCIM data schema included an entity to store point cloud data. This study also implemented PCIM data parser to convert the labeled point cloud data to a human-readable XML format and PCIM viewer to visualize the PCIM data with a hierarchical structure. The integrated PCIM framework was validated by a case study conducted at an actual building construction site. The proposed framework could classify the semantic information from point cloud data and store the data based on the pre-defined PCIM data schema automatically. To evaluate the effectiveness, this study compared the proposed PCIM framework to existing Scan-to-BIM software. As a result, this study identified that PCIM could be a more useful tool than Scan-to-BIM to understand and monitor the as-is conditions of construction sites.

7.2 Research Contribution and Impact

In the construction industry, PCIM contributes to increasing the utilization of point clouds by providing a PCIM data schema to store and represent the semantic information of point cloud data with an object-oriented data structure. Based on the advantage of point cloud, which represent the current state of construction sites, PCIM will be able to replace BIM in some construction tasks such as construction quality management and construction inspection. Moreover, this research can fill the gaps in current studies on semantic segmentation of 3D point cloud by presenting a sequential deep learning method leveraging additional input parameters. This methodological approach may be able to answer the question of simultaneously detecting the object type and its property information that could not be solved. As this research presents the prototype of PCIM, various studies related to this are expected to be additionally derived. Thus, new approaches to bring technical advances of PCIM may be invented.

7.3 Limitations and Future Research

Under the PCIM framework, the information on construction objects and their properties were automatically classified and stored with a hierarchical data structure. In this way, PCIM can enrich the information of point cloud data while maintaining the real measurements. However, since the research on PCIM is in its early phase, several limitations are found as follows:

1. The current classification module in PCIM extracted limited semantic information because of the lack of training datasets. To classify non-structural

elements such as temporary structure or MEP, more diverse learning datasets collected from construction sites are required.

2. Current PCIM schema only represent general information, therefore, in the future work, PCIM schema extension is needed to represent other specific information such as defects or openings in the surface
3. Since the point clouds collected from real construction sites have some noise data derived from moving objects and wet surfaces, additional data pre-processing techniques to remove the noise data are required. This additional information and functionalities will be implemented in the next version of PCIM.

REFERENCES

Adán, Antonio, Blanca Quintana, Samuel A. Prieto, and Frédéric Bosché. 2018. "Scan-to-BIM for 'Secondary' Building Components." *Advanced Engineering Informatics* 37:119–38.

Anil, Engin, Burcu Akinci, and Daniel Huber. 2011. "Representation Requirements of As-Is Building Information Models Generated from Laser Scanned Point Cloud Data." in *Proceedings of the 28th ISARC*.

Armeni, Iro, Sasha Sax, Amir R. Zamir, and Silvio Savarese. 2017. "Joint 2D-3D-Semantic Data for Indoor Scene Understanding." *ArXiv E-Prints* arXiv:1702.01105.

Armeni, Iro, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 2016. "3D Semantic Parsing of Large-Scale Indoor Spaces." Pp. 1534–43 in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Barnea, Shahar, and Sagi Filin. 2012. "Extraction of Objects from Terrestrial Laser Scans by Integrating Geometry Image and Intensity Data with Demonstration on Trees." *Remote Sensing* 4(1).

Bell, Sean, Paul Upchurch, Noah Snavely, and Kavita Bala. 2015. "Material Recognition in the Wild with the Materials in Context Database." Pp. 3479–87 in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Bosché, Frédéric, Mahmoud Ahmed, Yelda Turkan, Carl T. Haas, and Ralph Haas. 2015.

"The Value of Integrating Scan-to-BIM and Scan-vs-BIM Techniques for Construction Monitoring Using Laser Scanning and BIM: The Case of Cylindrical MEP Components." *Automation in Construction* 49:201–13.

Bosché, Frédéric, Mahmoud Ahmed, Yelda Turkan, Carl T Haas, and Ralph Haas. 2015. "The Value of Integrating Scan-to-BIM and Scan-vs-BIM Techniques for Construction Monitoring Using Laser Scanning and BIM: The Case of Cylindrical MEP Components." *Automation in Construction* 49:201–13.

Boukamp, Frank, and Burcu Akinci. 2007. "Automated Processing of Construction Specifications to Support Inspection and Quality Control." *Automation in Construction* 17(1):90–106.

Boulch, A., B. Le Saux, and N. Audebert. 2017. "Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks." Pp. 17–24 in *Proceedings of the Workshop on 3D Object Retrieval*, *3Dor '17*. Goslar, DEU: Eurographics Association.

buildingSMART. 2018. "Industry Foundation Classes." Retrieved (https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/link/object-definition.htm).

Chen, Jingdao, and Yong K. Cho. 2018. "Point-to-Point Comparison Method for Automated Scan-vs-BIM Deviation Detection." in *17th International Conference on Computing in Civil and Building Engineering*. Tampere, Finland.

Chen, Jingdao, Zsolt Kira, and Yong K. Cho. 2019. "Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction." *Journal of*

Computing in Civil Engineering 33(4):4019027.

Dai, Angela, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes." *ArXiv E-Prints* arXiv:1702.04405.

Dimitrov, Andrey, and Mani Golparvar-Fard. 2014a. "Vision-Based Material Recognition for Automated Monitoring of Construction Progress and Generating Building Information Modeling from Unordered Site Image Collections." *Advanced Engineering Informatics* 28(1):37–49.

Dimitrov, Andrey, and Mani Golparvar-Fard. 2014b. "Vision-Based Material Recognition for Automated Monitoring of Construction Progress and Generating Building Information Modeling from Unordered Site Image Collections." *Advanced Engineering Informatics* 28(1):37–49.

Eastman, Chuck, Paul Teicholz, Rafael Sacks, and Kathleen Liston. 2008. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley Publishing.

Fehr, Duc, William J. Beksi, Dimitris Zermas, and Nikolaos Papanikolopoulos. 2016. "Covariance Based Point Cloud Descriptors for Object Detection and Recognition." *Computer Vision and Image Understanding* 142:80–93.

Fischler, Martin a., and Robert C. Bolles. 1981. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." *Communications of the ACM* 24(6):381–95.

Graham, Benjamin, Martin Engelcke, and Laurens van der Maaten. 2017. "3D Semantic Segmentation with Submanifold Sparse Convolutional Networks." *ArXiv E-Prints* arXiv:1710.07563.

Gross, H., B. Jutzi, and U. Thoennessen. 2008. "Intensity Normalization By Incidence Angle and Range of Full-Waveform Lidar Data." Pp. 405–12 in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Beijing, China.

Guldur, Burcu, and Jerome F. Hajjar. 2014. "Damage Detection on Structures Using Texture Mapped Laser Point Clouds." Pp. 324–34 in *Structures Congress, Proceedings*. Boston, MA, USA.

Guo, Yulan, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. 2019. "Deep Learning for 3D Point Clouds: A Survey." *ArXiv E-Prints* arXiv:1912.12033.

Hackel, Timo, Nikolay Savinov, Lubor Ladicky, Jan D. Wegner, Konrad Schindler, and Marc Pollefeys. 2017. "Semantic3d. Net: A New Large-Scale Point Cloud Classification Benchmark." *ArXiv Preprint ArXiv:1704.03847*.

Hamledari, Hesam, Ehsan Rezazadeh Azar, and Brenda McCabe. 2018. "IFC-Based Development of As-Built and As-Is BIMs Using Construction and Facility Inspection Data: Site-to-BIM Data Transfer Automation." *Journal of Computing in Civil Engineering* 32(2):4017075.

Han, Kevin K., and Mani Golparvar-Fard. 2015. "Appearance-Based Material

Classification for Monitoring of Operation-Level Construction Progress Using 4D BIM and Site Photologs.” *Automation in Construction* 53:44–57.

Han, Kevin K, and Mani Golparvar-Fard. 2015. “Appearance-Based Material Classification for Monitoring of Operation-Level Construction Progress Using 4D BIM and Site Photologs.” *Automation in Construction* 53:44–57.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition.” Pp. 770–78 in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, Diane, Liefeng Bo, and Xiaofeng Ren. 2011. “Toward Robust Material Recognition for Everyday Objects.” Pp. 1–11 in *BMVC 2011 - Proceedings of the British Machine Vision Conference 2011*.

Hyunjoo, Kim, and Ahn Hongseob. 2011. “Temporary Facility Planning of a Construction Project Using BIM (Building Information Modeling).” Pp. 627–34 in *Computing in Civil Engineering*, *Proceedings*. Miami, Florida.

Ioannidou, Anastasia, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. 2017. “Deep Learning Advances in Computer Vision with 3D Data: A Survey.” *ACM Computing Surveys* 50(2).

Järemo Lawin, Felix, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. 2017. “Deep Projective 3D Semantic Segmentation.” *ArXiv E-Prints* arXiv:1705.03428.

Jolliffe, Ian T., and Jorge Cadima. 2016. "Principal Component Analysis: A Review and Recent Developments." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374(2065):20150202.

Kashani, Alireza G., and Andrew J. Graettinger. 2015. "Cluster-Based Roof Covering Damage Detection in Ground-Based Lidar Data." *Automation in Construction* 58:19–27.

Kim, Pileun, Jisoo Park, Yong K. Cho, and Junsuk Kang. 2019. "UAV-Assisted Autonomous Mobile Robot Navigation for as-Is 3D Data Collection and Registration in Cluttered Environments." *Automation in Construction* 106:102918.

Krijnen, Thomas, and Jakob Beetz. 2017. "An IFC Schema Extension and Binary Serialization Format to Efficiently Integrate Point Cloud Data into Building Models." *Advanced Engineering Informatics* 33:473–90.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM*, 84–90.

Kyungki, Kim, Cho Yong, and Kim Kinam. 2018. "BIM-Driven Automated Decision Support System for Safety Planning of Temporary Structures." *Journal of Construction Engineering and Management* 144(8).

Li, Lin, Fan Yang, Haihong Zhu, Dalin Li, You Li, and Lei Tang. 2017. "An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells." *Remote Sensing* 9(5).

Lichti, D., and B. Harvey. 2002. “THE EFFECTS OF REFLECTING SURFACE MATERIAL PROPERTIES ON TIME-OF-FLIGHT LASER SCANNER MEASUREMENTS.” in *ISPRS*.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2014. “Fully Convolutional Networks for Semantic Segmentation.” *ArXiv E-Prints* arXiv:1411.4038.

Macher, Hélène, Tania Landes, and Pierre Grussenmeyer. 2017. “From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings.” *Applied Sciences* 7(10).

Miettinen, Reijo, and Sami Paavola. 2014. “Beyond the BIM Utopia: Approaches to the Development and Implementation of Building Information Modeling.” *Automation in Construction* 43:84–91.

Moreno, Cristina, Svetlana Olbina, and Raja R. Issa. 2019. “BIM Use by Architecture, Engineering, and Construction (AEC) Industry in Educational Facility Projects.” *Advances in Civil Engineering* 2019:19.

Motawa, Ibrahim, and Abdulkareem Almarshad. 2013. “A Knowledge-Based BIM System for Building Maintenance.” *Automation in Construction* 29:173–82.

Mukupa, W., G. W. Roberts, C. M. Hancock, and K. Al-Manasir. 2017. “A Review of the Use of Terrestrial Laser Scanning Application for Change Detection and Deformation Monitoring of Structures.” *Survey Review* 49(353):99–116.

Park, Jisoo, Jing Dao Chen, and Yong K. Cho. 2020. “Point Cloud Information Modeling

(PCIM): An Innovative Framework for as-Is Information Modeling of Construction Sites.” in *Construction Research Congress*.

Park, Jisoo, Pileun Kim, Yong K. Cho, and Junsuk Kang. 2019. “Framework for Automated Registration of UAV and UGV Point Clouds Using Local Features in Images.” *Automation in Construction* 98:175–82.

Pătrăucean, Viorica, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, and Carl Haas. 2015. “State of Research in Automatic As-Built Modelling.” *Advanced Engineering Informatics* 29:162–71.

Pu, Shi, and George Vosselman. 2009. “Knowledge Based Reconstruction of Building Models from Terrestrial Laser Scanning Data.” *ISPRS Journal of Photogrammetry and Remote Sensing* 64:575–84.

Qi, Charles R., Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation.” Pp. 77–85 in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*.

Qi, Charles R., Li Yi, Hao Su, and Leonidas J. Guibas. 2017. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.” *Advances in Neural Information Processing Systems* 5100–5109.

Rashidi, Abbas, Mohamad Hoseyn Sigari, Marcel Maghiar, and David Citrin. 2016. “An Analogy between Various Machine-Learning Techniques for Detecting Construction Materials in Digital Images.” *KSCE Journal of Civil Engineering* 20(4):1178–88.

Rusu, R. B., N. Blodow, and M. Beetz. 2009. “Fast Point Feature Histograms (FPFH) for 3D Registration.” Pp. 3212–17 in *2009 IEEE International Conference on Robotics and Automation*.

Rusu, Radu Bogdan, and Steve Cousins. 2011. “3D Is Here: Point Cloud Library (PCL).” in *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China.

Sharan, Lavanya, Ce Liu, Ruth Rosenholtz, and Edward H. Adelson. 2013. “Recognizing Materials Using Perceptually Inspired Features.” *International Journal of Computer Vision* 103(3):348–71.

Son, Hyojoo, Changmin Kim, Nahyae Hwang, Changwan Kim, and Youngcheol Kang. 2014. “Classification of Major Construction Materials in Construction Environments Using Ensemble Classifiers.” *Advanced Engineering Informatics* 28(1):1–10.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. “Going Deeper with Convolutions.” Pp. 1–9 in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Tan Qu, and Wei Sun. 2015. “Usage of 3D Point Cloud Data in BIM (Building Information Modelling): Current Applications and Challenges.” *Journal of Civil Engineering and Architecture* 9(11):1269–78.

Tang, Pingbo, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. 2010. “Automatic Reconstruction of As-Built Building Information Models from Laser-

Scanned Point Clouds: A Review of Related Techniques." *Automation in Construction* 19(7):829–43.

Tarsha-Kurdi, Fayez, Tania Landes, and Pierre Grussenmeyer. 2007. "Extended Ransac Algorithm for Automatic Detection of Building Roof Planes From Lidar Data." Pp. 407–12 in *ISPRS Workshop on Laser Scanning*.

Tchapmi, Lyne P., Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. 2017. "SEGCloud: Semantic Segmentation of 3D Point Clouds." *ArXiv E-Prints* arXiv:1710.07563.

Vo, Anh-Vu, Debra F. Laefer, and Michela Bertolotto. 2016. "Airborne Laser Scanning Data Storage and Indexing: State-of-the-Art Review." *International Journal of Remote Sensing* 37(24):6187–6204.

Wang, Chao, Yong K. Cho, and Changwan Kim. 2015. "Automatic BIM Component Extraction from Point Clouds of Existing Buildings for Sustainability Applications." *Automation in Construction* 56:1–13.

Wang, Cheng, Chenglu Wen, Yudi Dai, Shangshu Yu, and Minghao Liu. 2020. "Urban 3D Modeling with Mobile Laser Scanning: A Review." *Virtual Reality & Intelligent Hardware* 2(3):175–212.

Wing, Brian M., Martin W. Ritchie, Kevin Boston, Warren B. Cohen, and Michael J. Olsen. 2015. "Individual Snag Detection Using Neighborhood Attribute Filtered Airborne Lidar Data." *Remote Sensing of Environment* 163:165–79.

Yu, Qunzhou, Kaiman Li, and Hanbin Luo. 2016. “A BIM-Based Dynamic Model for Site Material Supply.” *Procedia Engineering* 164:526–33.

Zhang, Chengyi, and David Arditi. 2020. “Advanced Progress Control of Infrastructure Construction Projects Using Terrestrial Laser Scanning Technology.” *Infrastructures* 5(10).