Integrating Inverse Photogrammetry and a Deep Learning–Based Point Cloud Segmentation Approach for Automated Generation of BIM Models

Zhongming Xiang, Ph.D., Aff.M.ASCE[1]; Abbas Rashidi, M.ASCE[2]; and Ge Ou, Aff.M.ASCE[3]

Abstract: Automatically converting three-dimensional (3D) point clouds into building information modeling (BIM) has been an active research area over the past few years. However, existing solutions in the literature have been suffering the limitations of covering all different design scenarios (prior knowledge-based approach) or collecting sufficient point clouds as training data sets (3D deep learning–based approach). To tackle this issue, we propose a fused system to automatically develop as-built BIMs from photogrammetric point clouds. A series of images is captured to generate a high-quality point cloud, which is then preprocessed by removing noise and downsizing points. Meanwhile, a two-dimensional (2D) deep-learning method, DeepLab, is utilized to semantically segment elements (e.g., walls, slabs, and columns) from the collected images. Subsequently, an inverse photogrammetric pipeline is employed to recognize element categories in the point cloud by projecting the isolated 3D planes into 2D images and assigning the identified elements to the 3D planes. Finally, the industry foundation classes are devised to create as-built BIMs based on the segmented point clouds. In order to evaluate the performance of the proposed system, we selected six cases with various elements as the testbed. The prospective results reveal that (1) our system can provide a highly automated solution to develop as-built BIMs; and (2) 39 out of 45 elements in six different cases are successfully recognized in point clouds. **DOI: 10.1061/JCEMD4.COENG-13020.** *© 2023 American Society of Civil Engineers.*

Author keywords: Building information modeling (BIM); Point cloud; Two-dimensional (2D) deep learning; Inverse photogrammetry; Element identification; Industry foundation classes (IFC).

Introduction

Building information modeling (BIM) plays a major role in the architecture, engineering, and construction (AEC) industry by increasing efficiency and shrinking project life-cycles through automation, visualization, and productivity. Despite the expected reduction of the global economy in the next few years, the BIM-related market is projected to have stable growth and experience a twofold increase compared with 2020, as estimated by Petra Gartzen (2022). In the AEC industry, the current practice of developing as-built BIM involves scanning existing buildings with a laser scanner to generate point clouds, followed by counterdrawing elements from point clouds operated by a three-dimensional (3D) draftsman using commercial software. This process is labor-intensive, error-prone, and tedious. To accelerate the process of as-built BIM development, a number of vendors have recently launched customized programs to semiautomatically detect and generate components from point clouds. For instance, Autodesk Plant 3D can generate pipes by manually selecting two points (Autodesk 2021). However, there is a lack of well-established software that is fully automatic and capable of generating high-quality as-built BIMs without manual intervention. As a result, this is still an ongoing area of research, and several studies have been conducted in the last few years on this specific topic.

Generally speaking, the major tasks for as-built BIM development are (1) identifying geometrical information, (2) defining element types, and (3) visualizing models in digital format. For the first task, the geometrical information is normally derived from point clouds, which can be developed using two approaches: image-based and laser scanning (Brilakis et al. 2011; Rashidi et al. 2011, 2013). Generating point clouds using either approach is now common practice, and research in this area is mainly focused on improving the quality of points (Xu et al. 2021) and/or accurately deriving absolute scales (Rashidi et al. 2015).

For the second task of defining element types in point clouds, the existing solutions are mainly prior knowledge-based and 3D deep-learning-based. The prior knowledge focuses either on element features [e.g., beams are horizontally placed (Kim et al. 2018)] or geometrical relationships [e.g., piers are placed underneath slabs in bridges (Lu et al. 2019)]. The 3D deep-learning-based method directly predicts element categories by using a network trained from a large number of semantically segmented point clouds. In addition to these two approaches, other methods include ensemble learning (Chen et al. 2020) and density clustering (Aljumaily et al. 2017). Considering the complexity of the issue, this is still an ongoing area of research. To date, this problem does not have a comprehensive solution.

For the third task of visualizing models in a digital format, several commercial software packages already exist, and the major barrier is exchanging data between those platforms (Lai et al. 2019). To tackle this issue, industry foundation classes (IFC), a standard data-exchange format, has been introduced by buildingSMART (BuildingSmart 2021; Tang et al. 2020).

[1] Transmission Line Engineer, Stantec, Salt Lake City, UT 84121 (corresponding author). Email: gary.xiang@stantec.com

[2] Assistant Professor, Dept. of Civil and Environmental Engineering, Univ. of Utah, Salt Lake City, UT 84112. Email: abbas.rashidi@utah.edu

[3] Assistant Professor, Dept. of Civil and Coastal Engineering, Univ. of Florida, Gainesville, FL 32603. Email: gou@ufl.edu

Note. This manuscript was submitted on August 11, 2022; approved on March 31, 2023; published online on June 22, 2023. Discussion period open until November 22, 2023; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Construction Engineering and Management***, © ASCE, ISSN 0733-9364.

Although several studies have been conducted to develop as-built BIMs by following the aforementioned three steps, one significant drawback in the workflow is identifying elements from point clouds since the existing solutions lack generalization. For one, the prior knowledge-based approach is designed for specific buildings and fails to deal with universal cases; however, the 3D deep-learning-based approach is highly dependent upon the quality and size of training point clouds, which are difficult to be prepared. Thus, a more reliable and robust solution is needed.

To address the need, this research proposes an automated system to generate as-built BIMs with a two-dimensional (2D) deep-learning-based approach. The building is scanned from various angles and positions to capture a set of images, which are then fed into a photogrammetric pipeline to develop a dense point cloud of the scene. Preprocessing procedures are then applied to remove outliers and downsize the generated point cloud. Meanwhile, a 2D deep-learning method, DeepLab, is utilized to implement semantic segmentation to identify element categories (e.g., column, shear wall, slab) within the captured images. Subsequently, an inverse photogrammetry method utilizes the projection relationship between 2D images and the 3D model to project the identified elements from corresponding images into the point cloud. Finally, the IFC schema, a data model developed by buildingSMART, is adopted to convert the point cloud with identified elements into a BIM model, which is interoperable with various platforms.

This paper is organized as follows: the state-of-the-art studies related to this research are briefly reviewed in the following section. In the section "Problem Statement, Objectives, and the Scope of Work," we discuss gaps in current studies and the objectives and scope of this research. The section "Research Methodology" presents an in-depth formulation of the proposed methodology. Then, the section "Experiments and Results" includes the implementation of our system, the performance, corresponding result analysis, and comparison with existing studies. Finally, the "Limitations" section summarizes the conclusions, discussions, and directions for potential future research.

Background

Developing as-built BIMs usually involves three stages: (1) data acquisition; (2) point cloud segmentation; and (3) BIM development from point clouds. This section will elaborate on and discuss existing studies comprehensively.

Data Acquisition

Generating point cloud data is considered fundamental for BIM development because it provides all necessary information, such as geometrical data and element categories. Depending on how spatial points are generated, the data-acquisition approaches are classified as either image-based or laser scanning (Bae et al. 2015; Rashidi et al. 2013; Ma and Liu 2018; Wang and Kim 2019). The image-based approach reconstructs point clouds by using a series of digital images or successive video frames and can be further divided into photogrammetry and/or videogrammetry based on the selected devices (Brilakis et al. 2011; Dai et al. 2013). Recent studies (Xu et al. 2021; Rebolj et al. 2017; Xiao et al. 2018; De Geyter et al. 2022) have analyzed the requirements of point clouds' quality for developing a promising BIM. Converting images to point clouds requires a set of complex algorithms, including camera calibration, feature extraction and matching, and multiview triangulation (Golparvar-Fard et al. 2011). In the step of feature extraction and matching, thousands of representative points (such as dramatic gradient changes of pixel values) are isolated from every single image to determine whether or not two images are matched by comparing the similarities of extracted features. Multiview triangulation is also a time-consuming step that determines point depths based on geometrical relationships. However, this approach's equipment cost is acceptable and normally less than $2,000 US for the most popular devices (e.g., Nikon D-80, Canon EOS 10D, and Rollei 6006) according to Wang et al. (2020).

Laser scanning is another favorable approach, which can directly identify the spatial locations of objects based on a laser beam's two-way travel time (time-of-flight-based) or the phase difference of emitted and received signals (phase-shift-based). According to data-acquisition platforms, laser scanning can be classified as terrestrial laser scanning (TLS), airborne laser scanning (ALS), and mobile laser scanning (MLS), where a laser scanner is mounted over the ground, on an aircraft, or on a mobile device, respectively. Each of these techniques has been widely used to facilitate BIM development based on different application scenarios: Yang et al. (2020) scanned a steel structure on a large scale with TLS and generated a BIM with detailed parameters; Sun et al. (2020) used ALS as a data-acquisition source to update the BIM models within a city model; Nikoohemat et al. (2020) implemented MLS to collect indoor point clouds of a multistory building and developed a model to help disaster management. In general, the laser scanning method has demonstrated its superiority due to its fast survey duration without postprocessing and promising measurement accuracy, which can be less than 1 mm (Trimble 2022; Sanhudo et al. 2020). However, the widespread use of the laser scanning method is limited by a higher cost when compared with the image-based method. From the summary by Wang et al. (2020), the costs of laser scanners (such as Trimble GX, FARO TLS, and RIEGL TLS) are usually more than $50,000 US, which is much higher than the equipment costs of the image-based approach. In addition, operators need to be specifically trained to deploy the data-acquisition task (Teizer and Kahlmann 2007).

Point Cloud Segmentation

One major challenge of converting point clouds into as-built BIMs is that point clouds are unstructured data sources, and it is difficult to process them and automatically extract desired information (e.g., information about different object categories such as walls, slabs, and columns). Segmenting point clouds is an efficient approach toward converting them into a more structured format. There are several types of segmentation methods, which can be classified into two major categories: (1) spatial-relationship-based (such as Farid and Sammut 2012; Wang 2019); and (2) machine-learning-based approaches (such as Chen et al. 2019; Kazi et al. 2020; Xue et al. 2020). For the spatial-relationship-based group of algorithms, the spatial distribution of points, such as distances between different points or the orientation of each point, is the basis used to divide point clouds into several organized groups. An example of this method is the region-growing plane-segmentation algorithm (Farid and Sammut 2012), which uses the smoothness factor to generate plane surfaces from point clouds. A point with the smallest curvature value is selected as the seed, and the normal vectors' angles between the seed and its neighbors are calculated and compared with a predefined threshold to determine if the neighbors are within the same plane as the seed. As a variant of RANSAC, m-estimator sample and consensus (MSAC) (Torr and Zisserman 1997) has been used to extract planes from point clouds by calculating the distances of points to an initial plane, which is fitted from a small portion of points randomly selected in point clouds (Wang 2019). In addition to these two algorithms, there are similar methods (Hichri et al. 2013), such as measuring the

Table 1. List of existing machine learning studies for segmenting point clouds

Category	Method	Targets	Data-acquisition technique	References
Supervised learning	SVM	Urban areas	Airborne LiDAR	Chen et al. (2019)
		Urban areas	Airborne LiDAR	Zhang et al. (2013)
	Random forests	Buildings	LiDAR	Bassier et al. (2019)
		Urban areas	LiDAR	Ni et al. (2017)
	Markov network	Urban areas	LiDAR	Munoz et al. (2008)
Unsupervised learning	K-means	Buildings	—	Kazi et al. (2020)
		Buildings	Airborne LiDAR	Kong et al. (2013)
	Hierarchical clustering	Urban areas	LiDAR	Xue et al. (2020)
		Buildings	LiDAR	Li et al. (2017)
Deep learning	MVCNN	Indoor objects	LiDAR	Zhang et al. (2018)
	SnapNet	Urban areas	RGB-D	Boulch et al. (2018)
	VoxNet	Indoor objects	RGB-D	Shen and Stamos (2021)
		Urban areas, indoor objects	LiDAR, RGB-D, CAD	Maturana and Scherer (2015)

distance between plane surfaces (Dorninger and Nothegger 2007), detecting element shapes (Ning et al. 2009), and fusing color data and spatial information (Zhan et al. 2009).

For machine learning (ML)-based algorithms, the underlying premise is using the machine to automatically learn the distribution rules of points and then predict the point groups. Numerous representative studies in this area are summarized in Table 1. ML-based methods could be divided into three subcategories: supervised learning; unsupervised learning; and deep learning.

The supervised learning-based methods provide a set of segmented point clouds and predefined features related to these clouds to make algorithms learn the rules, which are then applied to predict other unlabeled point clouds. Zhang et al. (2013) is a typical study of supervised learning that uses a support vector machine (SVM) to segment airborne LiDAR point clouds. They trained an SVM by using point clouds collected from a large area based on geometry, radiometry, topology, and so forth. Unsupervised learning algorithms are based on an automated procedure for learning different categories. A popular method in this category is the *k*-means clustering algorithm, which randomly selects a portion of points as the initial center and then updates the center based on the distances of the remaining points to the initial center (Kazi et al. 2020; Kong et al. 2013). This process is iterated until it meets a specific predefined criterion (e.g., a predefined iteration number or a minimum error variation). Hierarchical clustering is another unsupervised learning approach to segment point clouds (Xue et al. 2020).

Regarding deep-learning-based methods, researchers have proposed several advanced techniques with help from the incredible enhancement of computing ability in the last few years. Prior to applications in segmenting point clouds, deep-learning methods were typically used to process 2D images, so the straightforward approach is to transform point clouds into 2D images. The multiview convolutional neural network is a common method that projects point clouds from different angles into various 2D views, which are then predicted by a traditional convolutional neural network (Zhang et al. 2018). SnapNet processes point clouds in a similar way (Boulch et al. 2018) by transforming 3D models into several 2D images. Additionally, VoxNet is a method that uses voxels to uniformly group points into small grids, which are then processed by a 3D convolutional neural network (CNN) (Shen and Stamos 2021; Maturana and Scherer 2015). By summarizing all the previous studies, those methods can be classified as indirect segmentation because they deploy deep-learning methods to predict preprocessed point clouds (Zhang et al. 2019).

BIM Development from Point Cloud Data (Scan-to-BIM)

Seeking an efficient and automated approach to convert point clouds into as-built BIMs is another challenge for BIM development. As mentioned, the current AEC industry practice includes manually generating BIM from point clouds by using commercial software packages, which is time-consuming, error-prone, and tedious due to a series of manual operations, i.e., removing undesired points, inserting point clouds into commercial platforms (such as Autodesk Revit), setting drawing levels, and modeling elements based on the point cloud profiles (United-BIM 2022; TEJJY 2021). To address this issue, some products have been launched (such as Plant 3D, Leica CloudWorx, and Kubit PointSense Building), but automation improvement is limited since manual intervention is still required in some scenarios. For instance, PointSense Building can develop walls, floors, and ceilings from point clouds, but other objects, such as windows and doors, need to be manually created (FARO 2021). To address shortcomings in the current practice, a number of studies have been conducted to develop automated approaches for converting point clouds into as-built BIMs. Tables 2 and 3 show summaries of recent studies, which can be classified as a prior knowledge-based approach and as a 3D deep-learning-based method.

As illustrated in Table 2, the prior knowledge-based approach uses object attributes or geometrical relationships between elements to generate as-built BIMs from point clouds: (1) building elements normally have specific attributes, such as walls being vertical surfaces, columns are also vertical and have rectangular or circular cross-sections, and slabs are horizontal elements. Some researchers, such as Macher et al. (2021), Bassier and Vergauwen (2020), and Paiva et al. (2020), have utilized these object attributes to identify element categories; and (2) geometrical relationships between elements are typically consistent among different buildings and can be used to determine element categories from point clouds. For example, (1) the top and base constraints of walls are ceilings and floors, respectively (Jung et al. 2018); (2) windows and doors are openings in walls (Wang et al. 2015); and (3) slabs, piers, and girders in bridges are placed from top to bottom (Lu et al. 2019). In addition to independently implementing rules of object attributes or geometrical relationships, some researchers have integrated these two rules in converting point clouds to BIMs. For example, Anagnostopoulos et al. (2016) detected walls based on the fact that walls are vertically placed with large surfaces (object attributes) and have top and bottom boundaries constrained by floors and ceilings (geometrical relationships).

Table 2. Representative studies of BIM development from point clouds (prior knowledge-based approach)

Reference	Highlight	Details	Testbed	Data	Objects	Performance[a]	Efficiency
Romero-Jarén and Arranz (2021)	Geometrical relationships + object attributes	Floor and ceiling are horizontal. The parallel gap in floor and ceiling is the column. Irregular objects were not included.	Parking lots and office	Scanned by the authors	Walls, floors, ceilings, and columns	Global precision is around 90%.	—
Díaz-Vilariño et al. (2015)	Geometrical relationships + object attributes	Columns are vertical; and circular/rectangular columns had specific cross-sections.	Building foundation parking lots	Scanned by the authors	Columns	Precision: 53%–100% Recall: 68%–95%	—
Lu et al. (2019)	Geometrical relationships	Slabs, piers, pier caps, and girders are placed from top to down.	Bridges	Scanned by the authors	Slabs, piers, and girders	Precision: 92.0% (pie caps)–100% (girders)	~8.02 ± 3.02 min for each bridge
Anagnostopoulos et al. (2016)	Geometrical relationships + object attributes	Floors and ceilings are horizontal elements; ceilings are located higher than floors; walls are vertical and placed between floors and ceilings.	Office buildings	Scanned by the authors	Walls, slabs, and ceilings	Precision: 86% (exterior walls)–100% (floors, ceilings)	—
Wang et al. (2015)	Geometrical relationships + object attributes	Walls are vertical surfaces; windows and doors are openings within the walls; windows are located higher than doors; and roofs are vertical elements above the walls.	Residential and industrial buildings	Scanned by the authors	Walls, roofs, windows, and doors	Accuracy: 95% (windows)–100% (walls, doors, roofs)	—
Ochmann et al. (2019)	Geometrical relationships	Connections between walls, slabs, and rooms were used to separate point clouds. But doors and windows were ignored.	Office buildings	Scanned by the authors, Stanford 2D–3D-S Data set (public)	Room spaces	—	1–10 min
Bassier and Vergauwen (2020)	Object attributes	Walls were identified based on the feature of vertical and large plan surfaces, but other objects were ignored.	Office buildings	Stanford 2D–3D-S Data set (public)	Walls	Accuracy is greater than 95%	Less than 40 s for each building
Son et al. (2015)	Object attributes	Pipes, valves, and pumps were recognized based on the outlines. It assumed no noise.	Industrial buildings	Scanned by the authors	MEP	Recall and precision are both 100%	6.5 h from extracting point clouds to developing models of all 17 sets.
Jung et al. (2018)	Geometrical relationships + object attributes	Floor and ceiling were identified at first; floor-wall boundary was used to separate rooms; and windows and doors were identified at last.	School and apartment	Scanned by the authors	Floors, walls, ceilings, windows, and doors	RMSEs 0.089 m for school 0.074 m for apartment	140.79 s for school 58.5 s for apartment
Xue et al. (2019)	Geometrical relationships	The multi-modal optimization method was developed to address object detection and filtering in a complex environment.	Office buildings	Stanford 2D–3D-S Data set (public)	Theater chairs	Precision 99.3% Recall 98.0%	1,155.0 s
Macher et al. (2021)	Object attributes	Color, intensity, and geometry information of windows were unique and used to identify them from point clouds.	Heritage buildings	Scanned by the authors	Windows	Accuracy: 100%	—
Paiva et al. (2020)	Object attributes	Watershed transform and curvature analysis were combined to segment point clouds.	Heritage buildings	CHAS (public)	Roofs, walls, windows, floors, etc.	Accuracy: 84.10%–97.68%	—

[a] IoU is the intersection between the actual and derived labels divided by the union; precision is the ratio of correct positive observations to all positive observations; recall is the ratio of correct positive observations to all positive observations; accuracy is the ratio of correct observations to the total observations; and RMSE is the root mean square error.

Table 3. Representative studies of BIM development from point clouds (3D deep-learning-based approach)

Reference	Highlight	Details	Testbed	Data	Objects	Performance[a]	Efficiency
Perez-Perez et al. (2021)	Proposed deep-learning method (Scan2BIM-NET)	Two CNN and one RNN were combined to generate ScanToBIM-Net; the network was tested on 83 rooms.	Industrial buildings	Raamac Lab data set (public)	Beams, ceilings, columns, floors, pipes, and walls	Average accuracy is 86.13%	—
Wang et al. (2022b)	Existing deep learning method (DCNN)	RGB-D camera was used to collect point clouds; 2D semantic segmentation was conducted and projected to 3D maps; semantic 3D maps were transferred into LIDAR point clouds.	Industrial buildings	Scanned by the authors	MEP	Recall 91.0% Precision 95.6%	412 min (from data collection to BIM generation for four regular-size rooms)
Yin et al. (2021)	Proposed deep learning method (ResPointNet++)	Network was developed based on PointNet++.	Office buildings	Scanned by the authors	MEP	Recall 88.4% Precision 91.8%	—
Lee et al. (2021)	Proposed deep learning method (hierarchical DGCNN)	Setting of an extended range of neighbors is introduced in DGCNN to identify railway bridge components; and it has less training time than DGCNN.	Bridge	Scanned by the authors	Piers, decks, abutments, poles, and clutter	Recall 0.91 Precision 0.90	—
Pierdicca et al. (2020)	Proposed deep learning method (modified DGCNN)	Data set with cultural heritage buildings was created. They modified DGCNN to make it suitable for heritage point clouds. Half of the building was used as a training dataset and the rest was a testing data set.	Heritage buildings	ArCH (public)	Arcs, floors, doors, walls, stairs, vaults, roofs, etc.	Recall: 0.107 (arcs)–0.94 (roofs) Precision: 0.288 (arcs)–0.959 (roofs)	—
Tang et al. (2022)	Existing deep learning method (RandLA-Net)	Deep learning was used to identify objects and a morphological approach was used to separate volumetric spaces. It overcomes the limitation of missing doors/windows.	Office buildings	Stanford 2D–3D-S Data set (public)	Rooms, floors, ceilings, beams, walls, stairs, windows, doors, etc.	Precision: 53.2%	—
Kim and Kim (2021)	Existing deep-learning method (DGCNN)	Deep learning was used to identify elements; and missing data points were completed and modeled.	Office buildings	Synthetic point clouds, scanned by the authors	Floors, ceilings, columns, walls, and beams	IoU: 26.53% (walls)–97.22% (ceilings)	—
Xiong and Wang (2021)	Existing deep-learning method (PointNet)	Deep learning was used to identify elements; BIM instances were generated by using Dynamo.	Office buildings	Stanford 2D–3D-S Data set (public)	Floors, ceilings, beams, walls, stairs, windows, doors, etc.	Accuracy 95%	—
Park et al. (2022)	Existing deep-learning method (PointNet)	PointNet was used to predict the objects; boundary boxes of objects were defined and adjusted based on the spatial.	Office buildings	Stanford 2D–3D-S Data set (public)	Floors, ceilings, beams, walls, stairs, windows, doors, etc.	Accuracy: 71.46%	—
Wang et al. (2022a)	Proposed deep-learning method (ModelNet40)	Point-level features were extracted, and point feature distance maps were designed in the network.	Industrial buildings	Scanned by the authors	MEP	Accuracy 96.44%	—

Table 3. (*Continued.*)

Reference	Highlight	Details	Testbed	Data	Objects	Performance[a]	Efficiency
Xu et al. (2021)	Proposed deep learning method (CorDet)	A new network was created. A 3D bounding box was defined. Object-level and corner-level features were learned.	Office buildings	Stanford 2D–3D-S Data set (public)	Floors, ceilings, beams, walls, stairs, windows, doors, etc.	IoU: 81.6% (doors)–94.0% (columns)	0.53 s per room
Ma et al. (2020)	Existing deep learning method (PointNet, DGCNN)	They used BIM to generate point clouds for the training network.	Office buildings	Synthetic data set generated from BIM models	Floors, ceilings, beams, walls, stairs, windows, doors, etc.	IoU is 7.1% higher than trained on real data alone.	—

[a] IoU is the intersection between the actual and derived labels divided by the union; precision is the ratio of correct positive observations to all positive observations; recall is the ratio of correct positive observations to all positive observations; accuracy is the ratio of correct observations to the total observations; and RMSE is the root mean square error.

As seen in Table 3, the 3D deep-learning-based approach is another major area that uses advanced neural networks to directly learn features from labeled point clouds and then predict elements from unlabeled point clouds. Some networks have been developed based on different applications. The most popular network in this area is PointNet (Qi et al. 2017a), which obtains each point's feature with a multilayer perceptron and generates a global feature by integrating all point features. Nevertheless, it is only suitable for smaller scale-point clouds since the relationship between points is not considered. To address this issue, several networks [e.g., PointNet++ (Qi et al. 2017b), ResPointNet++ (Yin et al. 2021)] have been designed by modifying the architecture of PointNet or introducing additional features. A dynamic graph-based convolutional neural network (DGCNN) is another popular 3D deep-learning method, which includes an outstanding feature of adaptive convolutional filters for dynamic structures of local regions inside graphs (Phan et al. 2018). Some variants have been developed based on DGCNN to fit various application scenarios. For instance, the hierarchical DGCNN was designed by Lee et al. (2021) to consider slender piers of bridges by introducing an expanded range of neighboring points during feature extraction. One big issue of the 3D deep-learning-based approach is the difficulty of collecting sufficient point clouds and manually labeling every single point for training networks. To alleviate this situation, researchers have created and published some data sets, such as the Stanford 2D–3D-Semantics data set (Armeni et al. 2017), PartNet data set (Mo et al. 2019), ScanNet data set (Dai et al. 2017), and Architectural Cultural Heritage data set (Matrone et al. 2020). However, building types vary significantly and cannot be adequately represented by a few public point clouds.

Problem Statement, Objectives, and the Scope of Work

As discussed, the current automated methods for converting point clouds into BIM are prior knowledge-based and 3D deep-learning-based. The prior knowledge-based method is straightforward due to the adoption of object attributes or geometrical relationships, which can obtain promising performance on specific cases but may fail when a case slightly varies. For instance, Romero-Jarén and Arranz (2021) classified point clouds into columns based on the rule that ceilings and floors are the top and the bottom boundaries of columns; however, one typical design of columns in office buildings is that there is no ceiling, which exposes column-beam connections (see Case 2 in this study). In addition, this approach might fail to handle diverse cases, such as slanting columns that cannot be identified, as by Romero-Jarén and Arranz (2021), as well.

Thus, with the rapid development of computational techniques and a good understanding of neural networks, researchers have recently proposed a more robust approach based on advanced 3D deep-learning and gained encouraging results on various cases (e.g., regular office buildings, complex MEP rooms, and heritage buildings) by directly interpreting point clouds. This method successfully overcomes the aforementioned shortages of the prior knowledge-based approach due to the easy deployment of including various situations in training data sets. However, this approach has a significant defect in that dramatic effort is needed to collect sufficient point clouds and label these point clouds for training purposes. Although some public data sets (such as Stanford 2D–3D-Semantics data set, PartNet data set, ScanNet data set, and Architectural Cultural Heritage data set) are available and have been adopted by some studies (e.g., Park et al. 2022; Pierdicca et al. 2020), the generalization is still limited and cannot be applied

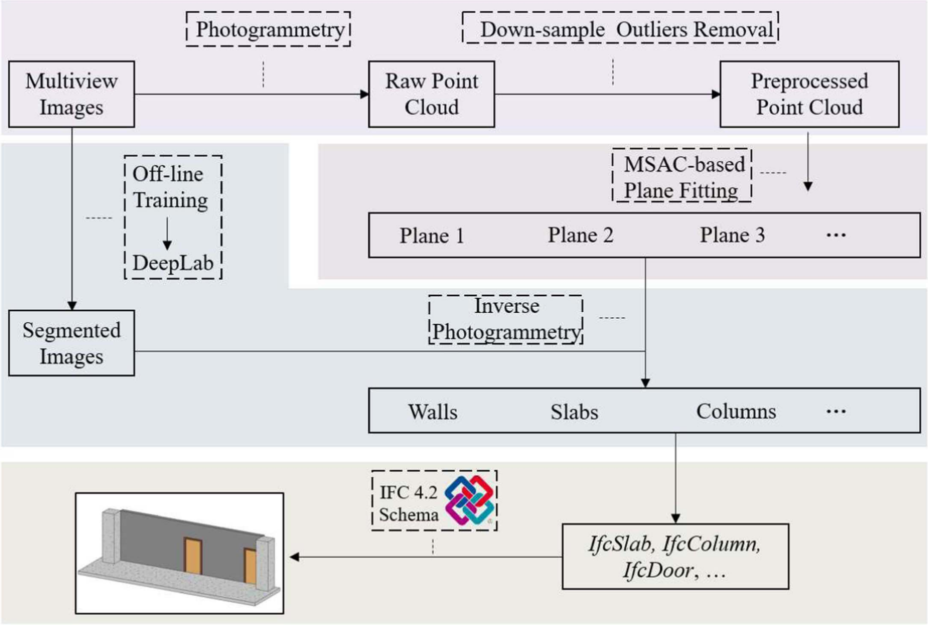


Fig. 1. Overall workflow of the proposed system.

on different cases (see details in the "Comparison Analysis" section). Moreover, as a new technique that emerged in the last few years, 3D deep-learning methods are far less promising than 2D deep-learning methods in terms of computational cost, training efficiency, and prediction performance (Guo et al. 2020; Minaee et al. 2021; Wang et al. 2022b).

To tackle such issues, this research proposes an automated system to develop as-built BIMs from point clouds by using a 2D deep-learning method to conduct 2D semantic segmentation and projecting segmented results from images to point clouds. The widely applied method, DeepLab, is selected as the 2D deep-learning method for this research. Meanwhile, to implement projecting from images to point clouds, we introduce an inverse photogrammetric method, which has demonstrated its feasibility in Braun and Borrmann (2019). Technical details of this system will be discussed in the following sections. The focus of this research is on common building elements, including walls, slabs, columns, doors, and windows, which have been frequently used in other studies (Tables 1–3). It is worth noting that we selected the 2D deep-learning method as the basis for identifying elements, and it is possible to generalize the results by training the system with more diverse data sets.

Research Methodology

The proposed system consists of four major components: (1) preprocessing point clouds to remove outliers and downsize the data set scale; (2) isolating planes from preprocessed point clouds; (3) recognizing element categories based on isolated planes; and (4) using IFC to generate BIM models. These four components are outlined in Fig. 1.

Preprocessing Point Cloud

As a well-established technique, photogrammetry has exhibited substantial advantages (e.g., low cost, high reliability, and ease of deployment) for the 3D reconstruction of civil infrastructure. The workflow of photogrammetry includes five major steps: (1) capturing a sequence of 2D images of targets from different positions, orientations, and angles; (2) extracting feature descriptors (e.g., SURF, SIFT, and KAZE) from the images; (3) matching correspondences between images based on the similarities of the descriptors; (4) utilizing random sample consensus (RANSAC) to remove outlier correspondences; and (5) generating a 3D model according to the geometrical relationships between the matched correspondences. Currently, there are several commercially available photogrammetric software packages such as Meshroom, ContextCapture, 3D Zephyr, and Pix4D. In order to expedite the process and avoid developing everything from scratch, this study directly implements ContextCapture, a reliable software launched by Bentley, for 3D reconstruction purposes.

The generated point clouds through the photogrammetric pipeline need to be further preprocessed due to (1) large scales of data sets, which would take up massive computational resources, and (2) the existence of outliers, which would cause errors and deviations during the element identification step. As a result, the first task of preprocessing is to downsize the generated point clouds. To accomplish this goal, nonoverlapping voxel grids are uniformly designed to separate points into several groups, and the centroid point of each grid is extracted to replace points in the corresponding grid (Fig. 2). The quality of a point cloud is controlled by the grid size since smaller grid sizes could generate more dense point clouds with higher quality while using larger size grids could result in generating sparse but lower quality point clouds. To determine a suitable grid size, this study conducted a series of experiments. Fig. 3 illustrates point clouds generated under different grid sizes (between 0.05 and 0.4 with an increment of 0.05). Meanwhile, Fig. 4 demonstrates changes in the decrease ratio of point numbers by applying various sizes of voxel grids. In this research, we selected 0.2 as the standard grid size due to (1) the decrease ratio of the original point cloud's point number to the new point cloud's point number is more than 90%, which can significantly reduce the computational cost, and (2) the point cloud quality is acceptable [Fig. 3(d)], as each object's texture is reserved to maintain the basic profile. By contrast, Fig. 3(h) is an example of a low-quality point cloud because of missing texture details.

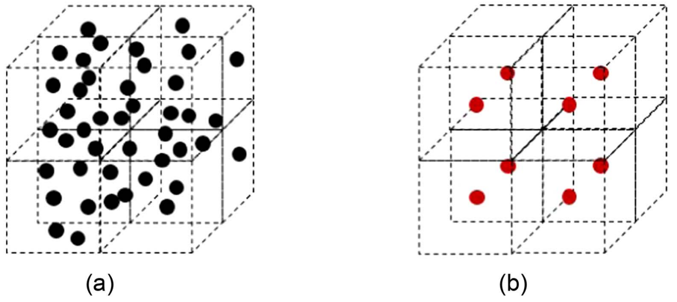


Fig. 2. Voxel grid-based downsizing: (a) raw 3D points with grids; and (b) centroid point in each grid.

The second step of preprocessing is to remove outliers caused by objects partially modeled in point clouds. Typically, outliers are located far away from desired points, which means a clustering method, such as the density-based spatial clustering application with noise (DBSCAN) (Ester et al. 1996), could be utilized to extract and remove them. The DBSCAN algorithm simply clusters spatial points into several groups based on distances; the groups with small numbers of points are considered outliers. DBSCAN is an efficient method to handle point clouds with consistent and uniform densities (Park et al. 2022). This is a suitable algorithm for this research since, by implementing voxel grids, the densities of the generated point clouds are consistent.

DBSCAN could be calibrated based on two major parameters: (1) point number n to determine the minimum number of points within the clustered group; and (2) search radius r, which is used as the largest distance between the points belonging to one group. In this research, we set n as a constant value equal to 1, which can identify and remove each singular point. Therefore, the noise-removal performance of DBSCAN is dependent on the selection of search radius r, which is a result of the voxel grid, as it has uniformly changed the distances between points. The ideal value of r should be slightly greater than the grid size to ensure that it is large enough to involve adjacent points as a group and small enough to differentiate outliers from the original points within the point cloud. As illustrated in Fig. 5, outliers could be successfully removed if $r = 0.3$ [Fig. 5(c)], but some or all points are misrecognized as outliers if r is not greater than the grid size [Figs. 5(a and b)], and no outlier could be identified when r is extremely large [Fig. 5(d)].

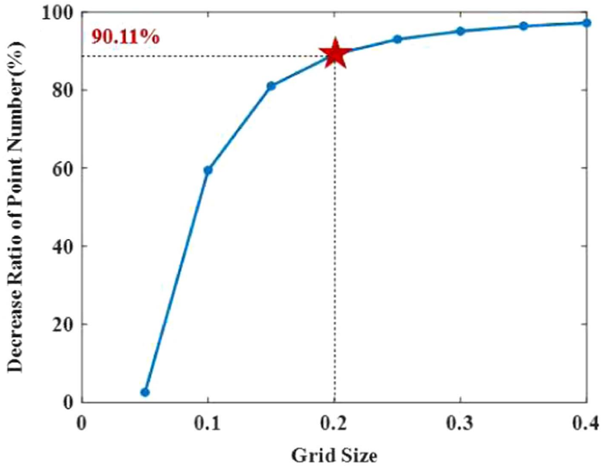


Fig. 4. Decrease ratios of point numbers under different sizes of voxel grids.

Point Cloud Segmentation

After preprocessing the original point clouds, the following step is to segment them into several groups. As stated in the "Background" section, several deep-learning methods have been proposed for segmentation in the last few years, and these methods need to be trained by a large number of point clouds. It is not always possible to collect and process a sufficient number of point clouds that are fine-tuned to train these methods. Thus, this research uses a classical method, MSAC (Torr and Zisserman 1997), to fit planes through the point clouds. As a reliable method, MSAC has been widely used to segment point clouds with a high level of accuracy and low computational costs (Gené-Mola et al. 2021; Pleansamai and Chaiyasarn 2019). Similar to RANSAC, MSAC randomly selects a few points to fit an initial plane, which is then used to calculate the consensus cost for the inliers and outliers based on the distances from these points to the fitted plane. Then, another set of points will be randomly selected to fit a new plane if the consensus cost is greater than a predefined threshold. This procedure will be iterated until the cost is smaller than the threshold, or when the iteration times reach a maximum defined limit. There is one possible issue with using the original RANSAC: the algorithm is sensitive to the value of the threshold. It does not assign penalties to inliers, while each outlier receives a constant penalty. As a result,

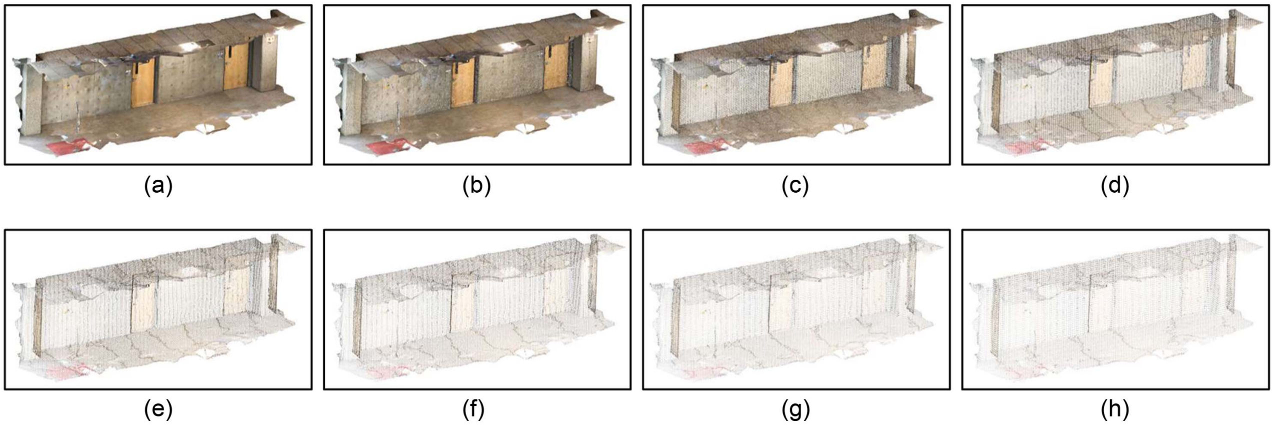


Fig. 3. Point cloud quality under different sizes of voxel grids: (a) 0.05; (b) 0.10; (c) 0.15; (d) 0.20; (e) 0.25; (f) 0.30; (g) 0.35; and (h) 0.40.

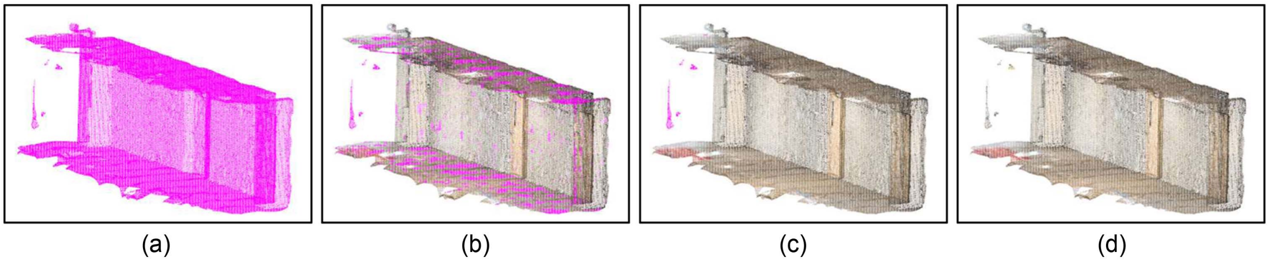


Fig. 5. Noise removal performance under different search radius: (a) $r = 0.1$; (b) $r = 0.2$; (c) $r = 0.3$; and (d) $r = 3$.

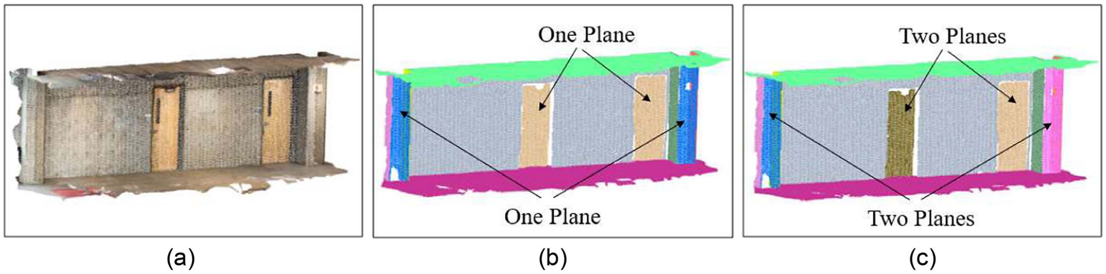


Fig. 6. MSAC-based plane fitting: (a) original point cloud; (b) initial planes; and (c) separated planes.

the algorithm might fail under a large threshold and might become unstable under a small one. To solve this issue, MSAC assigns a penalty to each inlier based on the distance between the inlier and the fitted plane (Subbarao and Meer 2006). In addition to the point cloud parameters, there are two input parameters in MSAC: the distance tolerance to determine each fitted plan's maximum thickness; and a threshold to be compared with consensus costs. In this research, and similar to the selection of the search radius r in DBSCAN, the fitted plan's distance tolerance is set as 0.3. A larger value might assign points from different objects into one group, while a smaller value might separate points from one object into various groups. The reason for this setting is that point clouds have been preprocessed by noise removal with a search radius of 0.3. The cost threshold has been tested several times to obtain promising results [as shown in Fig. 6(a)] and then can be used for other cases with the same data-acquisition settings. It should be noted that parameter values of these variables (e.g., search radius, distance tolerance, and cost threshold) in this research are dependent on the image capture device solution and need to be fine-tuned for each new device.

The MSAC algorithm might also suffer another problem: points located on the surfaces of different objects may be segmented into the same plane [see doors and columns in Fig. 6(b)]. To address this problem, the described DBSCAN algorithm is implemented again. Since point clouds have been preprocessed by voxel grid-based downsampling and DBSCAN-based noise removal, the distance between points has been restricted as a constant value, i.e., the size of voxel grids. That means a search radius in DBSCAN that is greater than the grid size can separate the surfaces in the same plane, so this research sets the search radius as 0.3.

Element Identification from Point Clouds

To semantically segment point clouds, Landrieu and Simonovsky (2018) proposed a superpoint graphs method to extract planes from point clouds and then used a deep-learning method to detect the categories of 3D planes. This method considers planes but not individual pixels as targets to conduct recognition, but it still needs to be trained on a data set with a sheer number of point clouds. To address this issue, we design a procedure to train the network on images, which are much more accessible and easier to be collected compared with point clouds. There are two steps in the proposed method: (1) semantically segmenting elements in images with a 2D deep-learning method; and (2) projecting segmentation results from images to point clouds with an inverse photogrammetry method.

Semantic Segmentation in Images with 2D Deep-Learning Method

A 2D semantic segmentation method, DeepLab, is introduced in this research to identify elements throughout images. DeepLab is an efficient deep-learning method and has shown promising performance in semantic segmentation (Chen et al. 2017). It is designed based on VGG16 and has an architecture as shown in Fig. 7. Compared with VGG16, the most powerful and unique feature of DeepLab is that it combines Atrous Convolution to enlarge the field view of filters without increasing the scale of parameters. By integrating Atrous Convolution with different sizes, the Atrous Spatial Pyramid Pooling is designed and finally used in DeepLab to obtain data information in multiscales during pooling operations. In addition to the feature of Atrous Convolution, there are modifications to improve the ability of semantic segmentation: (1) replacing the fully connected layer with a convolutional layer; (2) changing the stride size to 1 for the last two pooling layers; (3) using cross-entropy as the loss function; and (4) combining the output of VGG16 with a fully connected conditional random field. More technical details about DeepLab could be found in Chen et al. (2017).

To train and test the DeepLab, a set of images was collected and manually segmented into eight different categories: slabs; columns; walls; beams; ceilings; doors; windows, and clutter. In particular, the clutter category includes all other objects (e.g., chairs, tables, and plumbing) beyond the major seven elements listed in this text. Five examples of segmented images are presented in Fig. 8. Meanwhile, to demonstrate that DeepLab is the optimum network for our research, two well-known deep learning methods fully

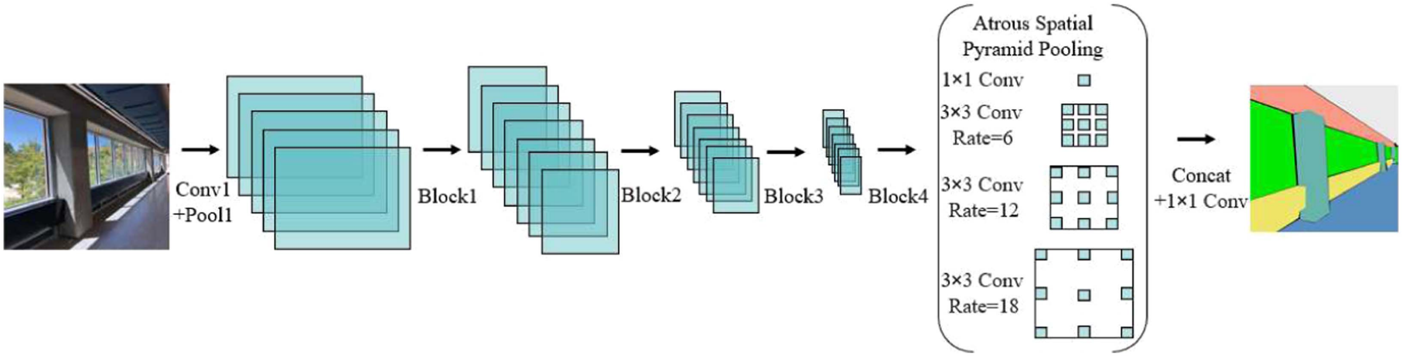


Fig. 7. Architecture of DeepLab. (Image by Zhongming Xiang.)

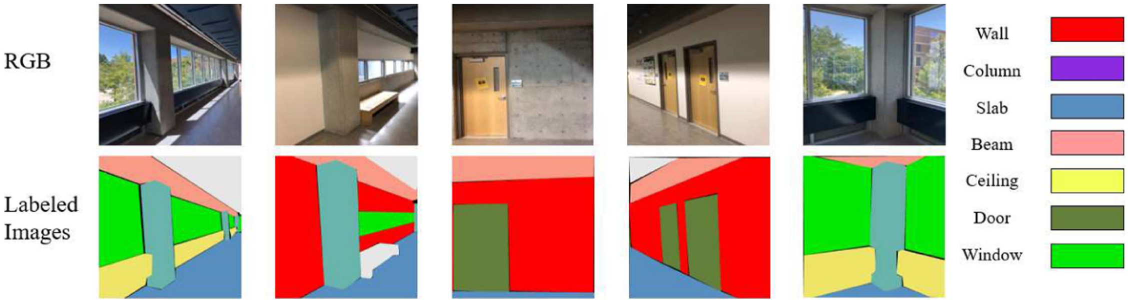


Fig. 8. Examples of segmented images for training DeepLab. (Images by Zhongming Xiang.)

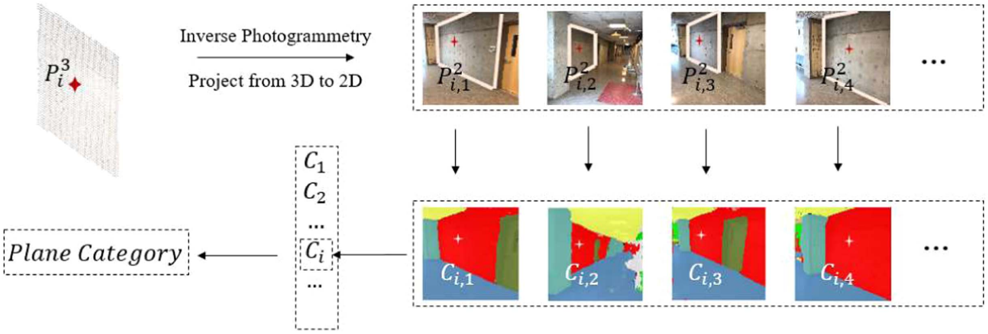


Fig. 9. Element identification by projecting 3D model into 2D images. (Images by Zhongming Xiang.)

convolutional networks (FCN) (Long et al. 2015) and U-Net (Ronneberger et al. 2015) have been selected to compare the performance of 2D semantic segmentation.

Element Identification in Point Cloud with Inverse Photogrammetry

Once the identified 2D images and the segmented point clouds are obtained, the element categories in point clouds can be determined by using the inverse photogrammetry method to project 3D points back to images. The category of each 2D point is assigned to the corresponding 3D point. Braun and Borrmann (2019) utilized inverse photogrammetry to label construction site images for neural networks based on segmented point clouds. The authors projected identified elements in point clouds into images to segment elements in 2D. Inspired by their work, we extend inverse photogrammetry to identify elements in point clouds based on segmented images.

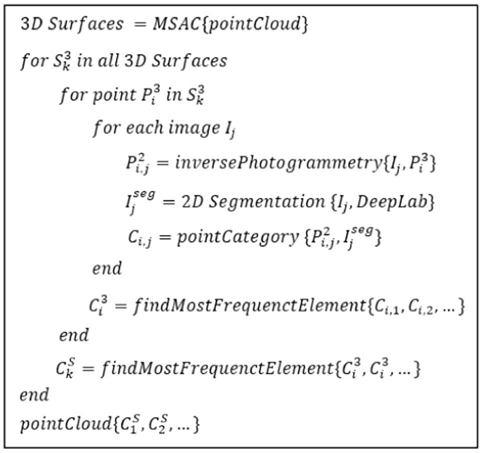
The workflow of the inverse photogrammetry-based element identification is illustrated in Fig. 9, and the corresponding pseudocode is presented in Fig. 10. To project 3D points back to images, three camera parameters are needed: the calibration matrix K; the rotation matrix R; and the translation matrix T, which are described in Eqs. (1)–(3)

$$K = [F_x \ 0 \ x_0 \ 0 \ F_y \ y_0 \ 0 \ 0 \ 1] \tag{1}$$

$$R = [R_{11} \ R_{12} \ R_{13} \ R_{21} \ R_{22} \ R_{23} \ R_{31} \ R_{32} \ R_{33}] \tag{2}$$

$$T = [T_1 \ T_2 \ T_3]' \tag{3}$$

where $[F_x, F_y]$ = the focal length; $[x_0, y_0]$ = the principal point; and R_{ij} and T_i = the extrinsic parameters for coordinate system transformation from 3D world coordinates to 3D camera coordinates.



```
$3D\ Surfaces = MSAC\{pointCloud\}$
$for\ S_k^3\ in\ all\ 3D\ Surfaces$
    $for\ point\ P_i^3\ in\ S_k^3$
        $for\ each\ image\ I_j$
            $P_{i,j}^2 = inversePhotogrammetry\{I_j, P_i^3\}$
            $I_j^{seg} = 2D\ Segmentation\ \{I_j, DeepLab\}$
            $C_{i,j} = pointCategory\ \{P_{i,j}^2, I_j^{seg}\}$
        $end$
        $C_i^3 = findMostFrequenctElement\{C_{i,1}, C_{i,2}, \ldots\}$
    $end$
    $C_k^S = findMostFrequenctElement\{C_i^3, C_i^3, \ldots\}$
$end$
$pointCloud\{C_1^S, C_2^S, \ldots\}$
```

Fig. 10. Pseudocode of element category determination in point clouds.

Assuming P_i^3 is a 3D point, the corresponding 2D point $P_{i,j}^2$ in image j can be calculated via Eqs. (4) and (5)

$$[m_1\ m_2\ m_3] = K \times [R\ T] \times P_i^3 \tag{4}$$

$$P_{i,j}^2 = \left[\frac{m_1}{m_3}\ \frac{m_2}{m_3}\right]' \tag{5}$$

Subsequently, a series of 2D points $\{P_{i,1}^2, P_{i,2}^2, \ldots P_{i,j}^2, \ldots\}$ corresponding to image sequence $\{I_1, I_2, \ldots, I_j, \ldots\}$ is derived for the 3D point P_i^3 in the studied plane. Based on the element recognition results in images, a set of categories $\{C_{i,1}, C_{i,2}, \ldots, C_{i,j}, \ldots\}$ is consequently determined for P_i^3. Considering that some 2D projected points might be located outside images, only the projected points located within images are considered effective points. Then, the category with the greatest occurrence among the effective points is indicated as the eventual category C_i^3 of the point P_i^3. As the result, a set of categories $\{C_1^3, C_2^3, \ldots, C_i^3, \ldots\}$ can be determined for all points in the studied plane. Similarly, the category corresponding to the highest frequency is indicated as the plane category. Fig. 11 shows the segmentation result of a sample.

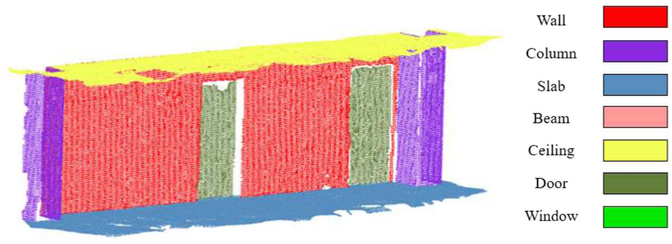


Fig. 11. Segmentation results for a point cloud.

Even though the previous procedures can efficiently identify element categories from point clouds, there remains an issue: surfaces of each column are separated as isolated planes [Fig. 12(a)]. To address the issue, an extra step is added within the system to group planes belonging to the same column together based on the distances between these planes [Fig. 12(b)].

IFC-Based BIM Development

The purpose of BIM development is to automatically manage building information based on visualized models. Thus, the last step of the proposed system is visualization, which designs an operational model. To achieve this goal, the IFC, providing efficient tools for object-based transactions between different BIM applications, has been adopted in this research. Various studies (Hamledari et al. 2018; Kim et al. 2016; Lu and Brilakis 2019) have demonstrated the feasibility of IFC being used to generate reliable BIM models. IFC could convert identified point clouds into BIM models that are interoperable in various programs, such as Revit, ArchiCAD, and Vectorworks. In this project, the IFC4.0 schema is adopted to generate as-built BIMs from point clouds. Since some case studies do not have enough wall supports, ceilings are not considered during the IFC-based BIM development step. Thus, we select walls, columns, slabs, beams, doors, and windows as the study targets.

To develop BIM models, this research adopts the instances of *IfcWallStandardCase* class for walls, *IfcColumn* class for columns, *IfcSlab* class for slabs, *IfcBeam* class for beams, *IfcDoor* for doors, and *IfcWindow* for windows. The relationships among different instances are demonstrated in Fig. 13, which shows that walls, columns, beams, and slabs can be created independently, while doors and windows rely on walls. A predefined script is designed to include all possible elements (e.g., windows, walls, columns, slabs, doors, etc.) and utilize the element numbers and the parameters of each element as variables. Take a wall instance as an example: If there is no wall element, the corresponding section of defining walls in the script will be skipped; otherwise, the script will be executed to define wall instances with dimension values. A file in IFC format for cross-platform applications will then be generated. The workflow of generating IFC models is presented in Fig. 14.

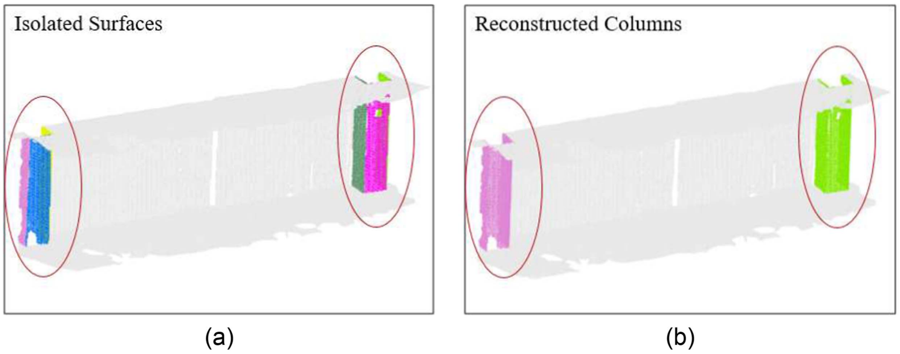


Fig. 12. Further processing to reconstruct columns from separated planes.

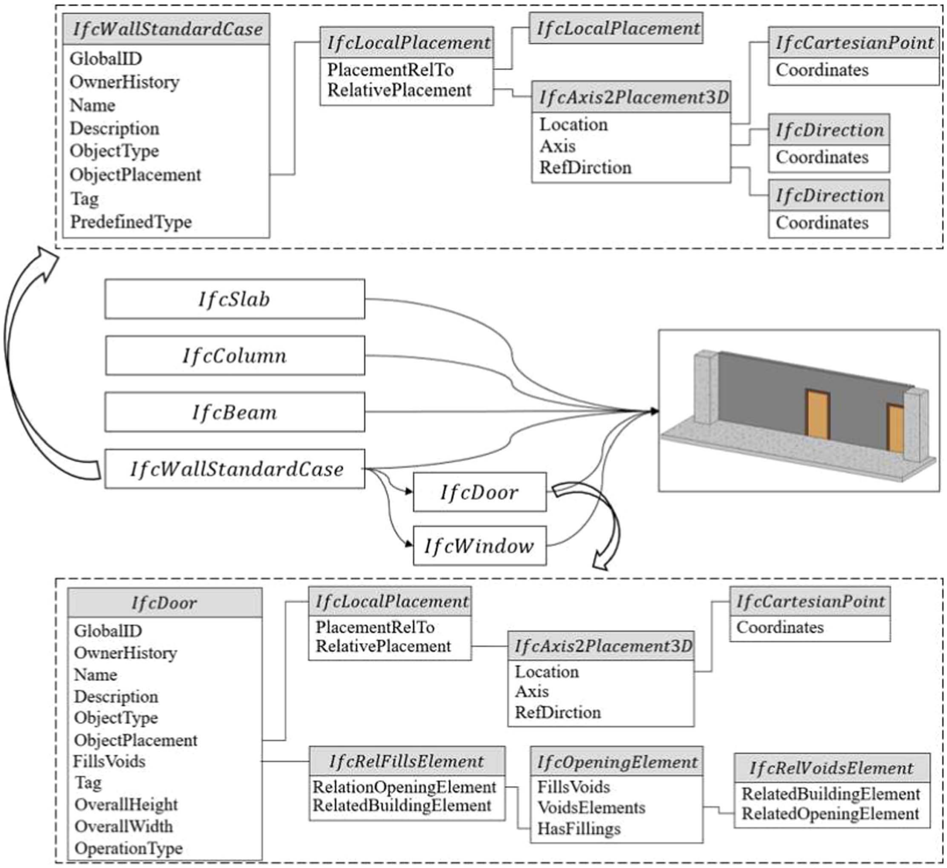


Fig. 13. Geometry retrieval chart of IFC schema used in this research.

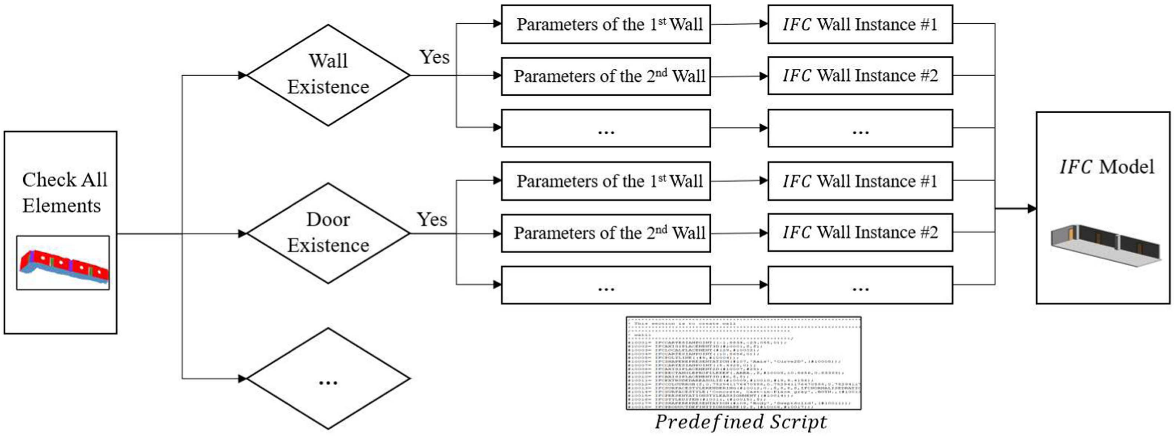


Fig. 14. Workflow of developing IFC models.

To generate a prospective BIM model with IFC, details need to be determined and clarified as follows:

1. During BIM development, dealing with invisible elements is a major challenge. Normally, invisible elements fall into two categories: subelements embedded in other elements (such as rebar), and elements that cannot be fully captured by cameras (such as partially invisible columns attached to walls). To handle embedded elements, other nondestructive sensing tools are needed to identify these elements' locations and sizes, which are then transformed into BIM models. We have previously dug into this area by using ground penetrating radar (GPR) with several methodologies (Xiang et al. 2020, 2021) and plan to integrate these methods with BIM development. Regarding the incomplete elements, the deduction is implemented since BIM models can be generated with only partial critical data rather than an entire model. For example, to generate a column, the following parameters are necessary: coordinates of bottom face center $[x, y, z]$, width w, length l, and height h. That means only two perpendicular vertical surfaces can build the model of a column since all of these six coordinates can be derived from these two surfaces.
2. Since the models in this research are generated based on the photogrammetry method, the dimensions are based on the pixel sizes of collected images. Manual dimensional measurements in

the real world are needed to proportionally rescale models. Nevertheless, this research takes three cases as examples to rescale the BIM model by involving a manually measured distance from the real world.

3. It is necessary to mention that materials are also critical features that should be identified when creating IFC data, but this step is beyond the scope of this research, and we simply applied a set of predefined materials to walls, columns, slabs, beams, doors, and windows.

Experiments and Results

To evaluate the feasibility of implementing the proposed system, six case studies are selected. The system performances in terms of generating point clouds, identifying elements in images and point clouds, and developing as-built BIMs are comprehensively discussed in this section.

Data Collection

As illustrated in Table 4, six cases from five different concrete buildings are used to evaluate the performance of the proposed system. A camera with a resolution of 3,024 × 4,032 pixels and a focal length of 4 mm is utilized to capture the necessary images. The image numbers vary according to different case sizes, i.e., the case on a small scale needs fewer images while the large-scale case needs more images.

Besides images for 3D reconstruction, another 300 images are captured from different concrete buildings to train the DeepLab described in previous sections. Meanwhile, data augmentation is implemented to increase the data size. Each image is randomly scaled, rotated, moved, and flipped, respectively; finally, 1,500 images are collected to train the DeepLab. These images are then manually segmented according to the aforementioned categories (walls, columns, slabs, beams, ceilings, doors, windows, and clutter).

Results and Discussion

As the primary step for evaluating the system's overall performance, we first assess the performance of DeepLab. As indicated earlier, we collected 1,500 images, of which 60% (900 images) are used as the training data set, 20% (300 images) are used as the validation data set, and the rest (300 images) are used for the testing data set. The performances of DeepLab and the two compared methods, FCN and U-Net, on the testing data set are illustrated in Table 5. The comparison study justifies the superiority of DeepLab over the other two methods: IoUs of most categories predicted by DeepLab are higher than the other two networks, except the IoUs of column and beam, which are slightly lower than U-Net. This table shows that promising results could be achieved by using DeepLab, as accuracies and IoUs of most elements (walls, slabs, columns, ceilings, doors, and windows) are greater than 0.9. However, IoUs of beams and clutter are around 0.87. For beams, the potential cause for this outcome is the lack of a corresponding category in the training data set. For the case of clutter, this research defines other items that are not listed (e.g., plumbing, chairs, trash cans) as clutter, which raises the difficulty to extract a uniform rule about spatial distributions or materials by using DeepLab.

As the next step, and by applying the proposed system to all six cases, the results, including the reconstructed point clouds, preprocessed results, segmented elements, and the corresponding BIMs, are obtained and presented as in Figs. 15–18. For point cloud generation, photogrammetry has been demonstrated as an efficient method since all six cases' point clouds are of high quality (Fig. 15). As mentioned, these point clouds need to be preprocessed due to the sheer number of points and the existence of outliers. Fig. 16 shows the preprocessed results, which verify that (1) the outliers are completely removed, and (2) point numbers are significantly decreased without causing any inverse impact on the quality of point clouds (Table 6).

The performance of semantic segmentation for point clouds is demonstrated in Fig. 17. This figure shows that the proposed approach is robust, as 39 out of 45 elements in point clouds are accurately identified. However, there are still some elements that cannot be recognized from point clouds. For instance, the glass door in Case 5 and the standalone column in Case 6 are identified as clutter. Further, the four windows in Case 5 are detected as larger than the actual sizes since the wall surfaces underneath the windows are located on the same planes as the windows. Additionally, the two columns attached to walls in Case 6 are identified as walls, caused by the fact that both columns are too small to be differentiated from walls by using DeepLab.

Finally, Fig. 18 shows the as-built BIMs generated based on IFC4.0 schema. All elements in point clouds are successfully generated except doors in Case 3. As mentioned, door instances rely on walls when creating BIM models with IFC4.0, but the dependent wall in Case 3 is misrecognized as clutter. In addition, the two walls in Case 4 are different from the original scanned objects due to the boundary setting in this research, which determines boundaries based on the extreme value of objects in the same direction. Take the left wall in Case 4 as an example, the minimum z value (assuming the vertical direction) is selected as the smaller one between the bottom points of two vertical boundaries. In this case, the left wall in Case 4 is larger in the model than in the real world.

Table 4. Details of the selected six cases

Category	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Building name and code	MCE	MCE	LSND	FMAB	CRCC	BPLS
Image number	205	711	477	324	658	590

Table 5. Numerical results for evaluating DeepLab for semantic segmentation of 2D images

IoU	Wall	Column	Slab	Beam	Celling	Door	Window	Clutter
DeepLab	0.95	0.90	0.95	0.88	0.94	0.92	0.95	0.88
FCN	0.92	0.86	0.87	0.81	0.91	0.91	0.86	0.82
U-Net	0.93	0.91	0.92	0.89	0.90	0.92	0.89	0.84

Note: IoU = intersection-over-union between the predicted and the ground-truth bounding boxes.

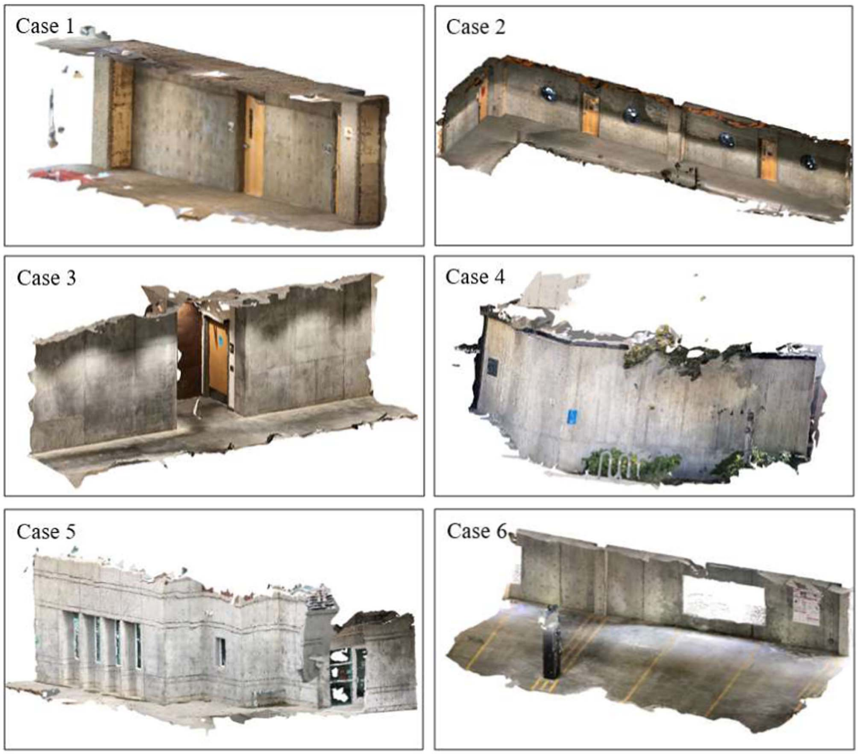


Fig. 15. Raw point clouds of six cases.

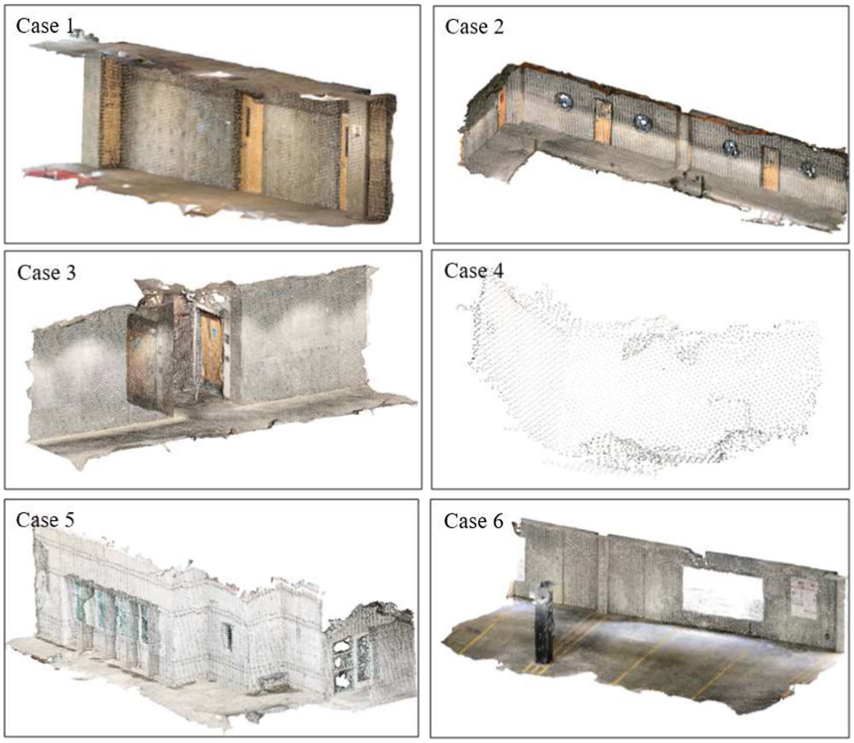


Fig. 16. Preprocessed point clouds of six cases.

After measuring at least one dimension of the actual building, the whole model could be scaled up. We take Case 1 as an example to illustrate the procedure of scaling up BIM models: (1) measuring dimensions $(a_1, a_2, a_3, \dots)$ from the real-world cases; (2) measuring the corresponding dimensions in pixels $(p_1, p_2, p_3, \dots)$ from the point clouds; (3) using the first dimension as a reference to obtain a ratio (a_1/p_1) of the real world to the point cloud; and (4) calculating the scaled dimensions $[a1p1 * (p1, p2, p3, \dots)]$ and applying these dimensions to BIM models. As shown in Fig. 19, several segments from the six cases are selected to discuss the

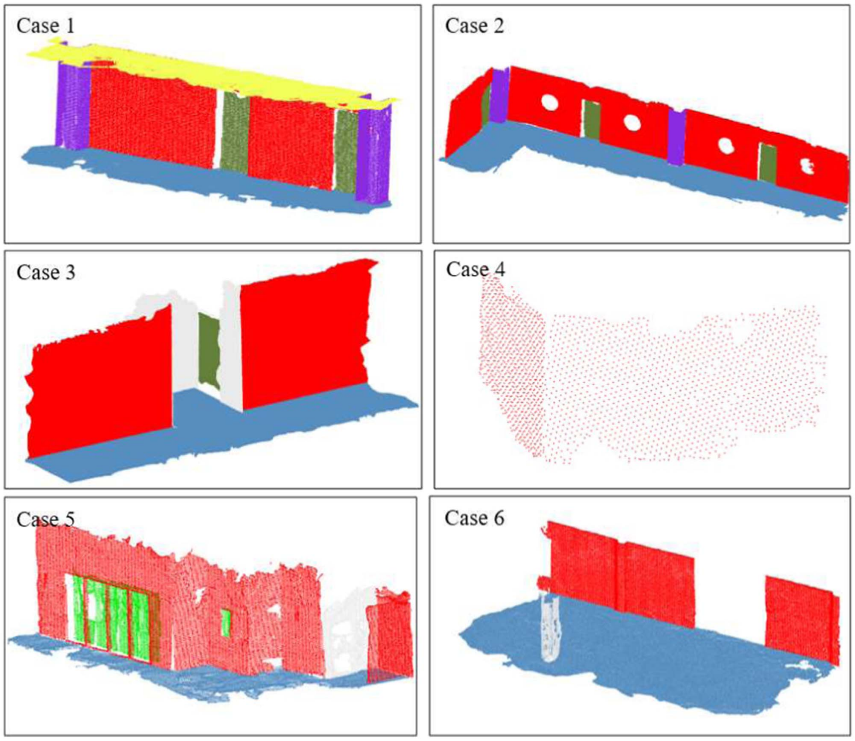


Fig. 17. Segmentation results of six cases.

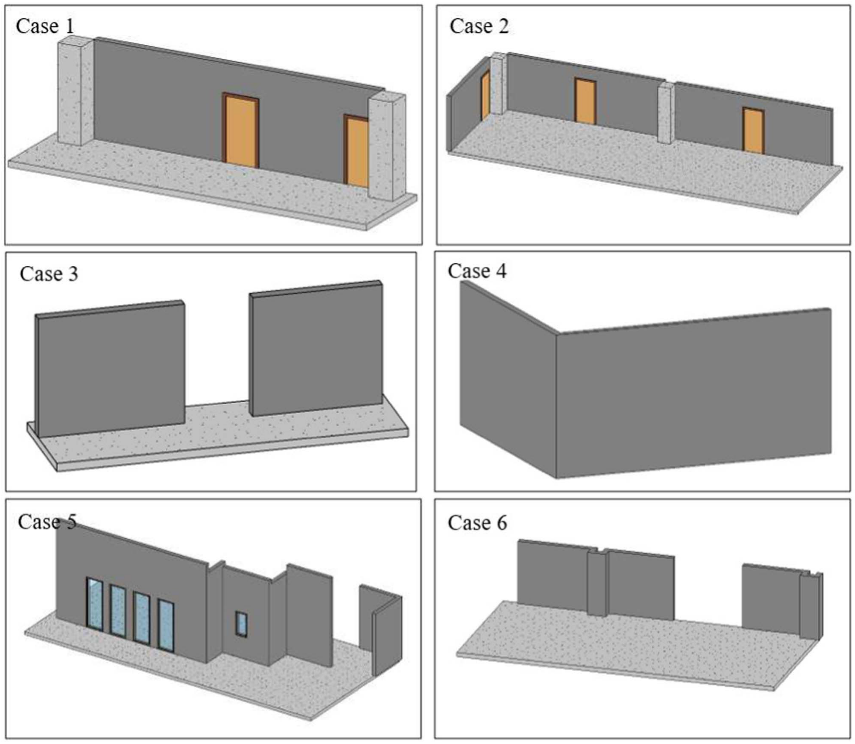


Fig. 18. IFC-based BIMs of all cases.

Table 6. Point numbers after preprocessing for all cases

Category	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Point number (raw point cloud)	580,890	1,280,202	1,781,752	628,933	406,944	652,928
Point number (preprocessed)	62,978	98,087	55,975	3,869	44,232	72,347
Decrease rate	89.16%	92.34%	96.86%	99.38%	89.13%	88.92%

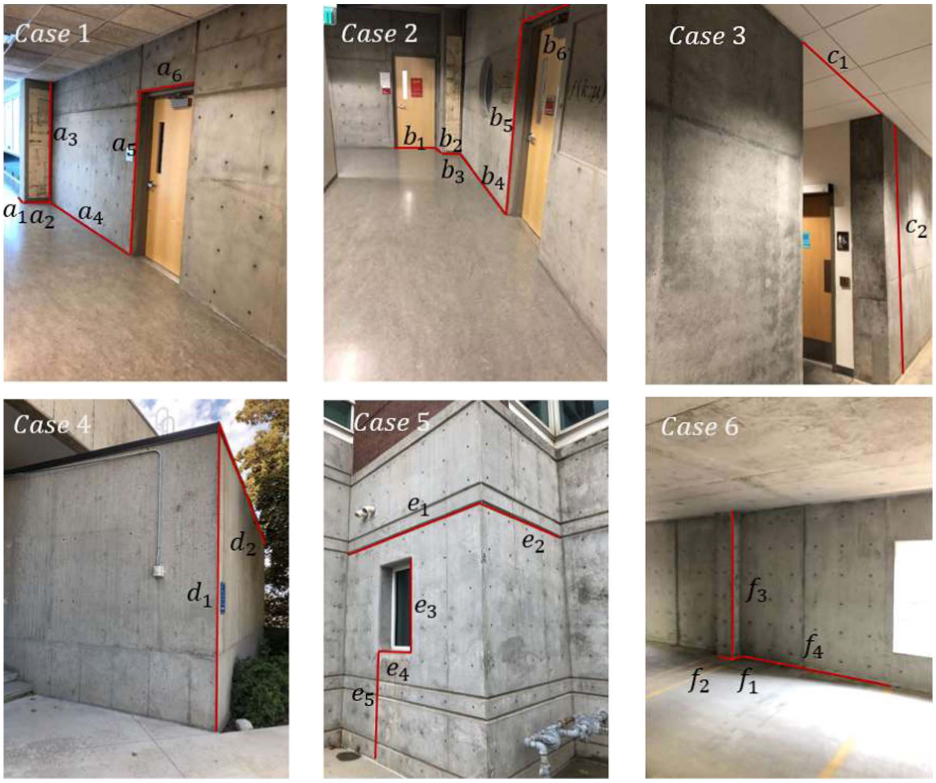


Fig. 19. Selected measured elements for dimensional accuracy evaluation. (Images by Zhongming Xiang.)

performance of determining dimensions. The absolute errors, calculated as the absolute value of the ratio of the difference between BIM distance and the real-world distance to the real-world distance, are determined for all selected segments and presented in Table 7. The results reveal that our models are correctly developed for each dimension due to the small absolute errors. Some values are even less than 1%. However, there are still some large deviations between the real-world cases and BIM models since absolute errors in some cases are larger than 5%, especially for d_2 in Case 4, which has an absolute error of 16.27%. The causes of these errors are primarily from the three steps: (1) point cloud generation that distorts element sizes; (2) 3D plane segmentation that cannot completely group all points from one plane together; and (3) strategy of determining element boundaries that loses element sizes' true details.

Sensitivity Analysis

The number of images for 3D reconstruction is a major factor that has a significant impact on the quality of point clouds since a large number of images could generate a high-quality point cloud, while a small number of images might produce a lower-quality and more sparse point cloud. Thus, this research analyzes the sensitivity of point clouds respective to the number of implemented images. Case studies 1 and 2 are selected as examples to implement the analysis. Images are captured from various angles and positions to generate data sets with different image numbers. There are four sets for Case 1 with image numbers 50, 100, 150, and 200 and four sets for Case 2 with image numbers 100, 200, 300, and 400, respectively. Fig. 20 shows the changes in point numbers and the computing times, along with the increase in image numbers (computation times are calculated based on a computer with an Intel i7-8700 processor, 64 GB memory, and NVIDIA GeForce RTX 2080 Ti graphics card). As illustrated in this figure, there is a trade-off between the quality of point clouds and the cost of computing resources when selecting image numbers. Meanwhile, the relationship between point clouds and image numbers depends on the scale of targets as well. As shown in Fig. 21, 200 images can generate a decent point cloud for Case 1, but the point cloud for Case 2 generated from the same number of images is missing several important details such as partial walls, as shown in Fig. 21(a).

Comparison Analysis

As mentioned in the "Background" section, the existing scan-to-BIM solutions can be classified as prior knowledge-based approaches and 3D deep-learning-based approaches, which are compared with the proposed method. Many researchers have proposed various prior knowledge-based approaches according to different application scenarios, and two of the most frequently cited works (Romero-Jarén and Arranz 2021; Wang et al. 2015) are compared because of the high degree of similarities between their study cases and our selected buildings. Both works include the following major steps: data collection; noise removal; downsampling; plane segmentation; and knowledge-based component classification. However, we assume that all the steps are perfectly implemented with the exception of the last step. Regarding the 3D deep-learning-based approach, the compared network is selected as PointNet, which has been widely used recently (e.g., Xiong and Wang 2021; Park et al. 2022; Ma et al. 2020) to convert point clouds into BIMs. The Stanford 2D–3D-Semantics data set (Armeni et al. 2017), a popular set of point clouds collected from six buildings, is adopted to train the PointNet. It should be noted that this public data set involves several objects (e.g., chairs, tables, boards, bookcases, sofas, stairs, etc.), which are beyond the scope of our research; thus, only the components mentioned in the previous sections are selected to train PointNet. In addition, before testing

Table 7. Distances measured from the real world and the point cloud

Case 1						
Category	a_1	a_2	a_3	a_4	a_5	a_6
Real world (cm)	61.00	61.00	271.00	382.00	217.00	102.00
BIM (cm)	61.00	61.38	276.03	389.07	218.08	102.75
Absolute error	0	0.63%	1.85%	1.85%	0.50%	0.73%
Case 2						
Category	b_1	b_2	b_3	b_4	b_5	b_6
Real world (cm)	102	60	49	375	217	102
BIM (cm)	102.00	59.92	50.34	403.11	221.46	92.37
Absolute error	0	0.13%	2.74%	7.50%	2.06%	9.44%
Case 3						
Category	c_1	c_2	—	—	—	—
Real world (cm)	152.00	275.00	—	—	—	—
BIM (cm)	152.00	265.49	—	—	—	—
Absolute error	0	3.46%	—	—	—	—
Case 4						
Category	d_1	d_2	—	—	—	—
Real world (cm)	351.50	748.00	—	—	—	—
BIM (cm)	351.50	869.74	—	—	—	—
Absolute error	0	16.27%	—	—	—	—
Case 5						
Category	e_1	e_2	e_3	e_4	e_5	—
Real world (cm)	298.00	178.00	127.00	66.00	148.00	—
BIM (cm)	298.00	167.01	128.53	62.21	148.94	—
Absolute error	0	6.17%	1.21%	5.74%	0.64%	—
Case 6						
Category	f_1	f_2	f_3	f_4	—	—
Real world (cm)	12.00	27.00	163.00	171.00	—	—
BIM (cm)	12.00	28.80	176.52	157.97	—	—
Absolute error	0	6.65%	8.29%	7.62%	—	—

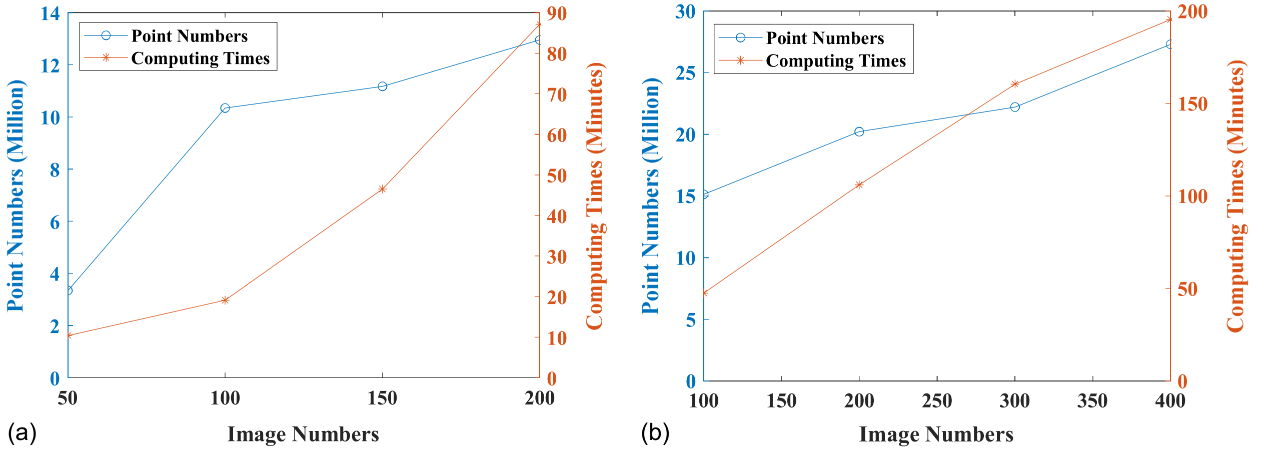


Fig. 20. Changes of point clouds' point numbers and computing times along with image numbers: (a) Case 1; and (b) Case 2.

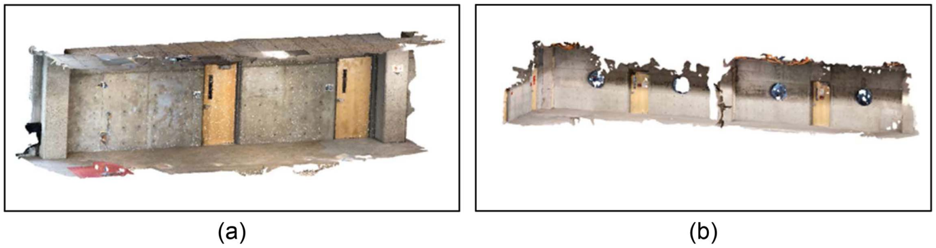


Fig. 21. Point clouds generated by 200 images: (a) Case 1; and (b) Case 2.

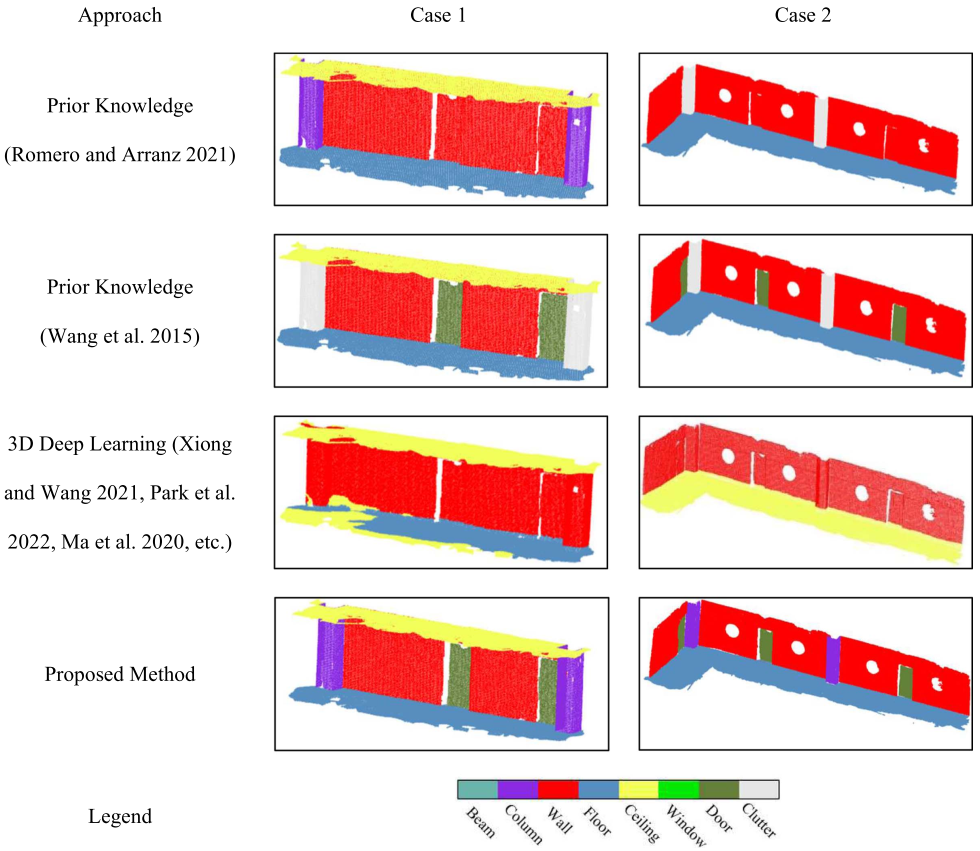


Fig. 22. Scan-to-BIM performance of the prior knowledge-based approach, 3D deep-learning-based approach, and our proposed method on Case 1 and Case 2.

Table 8. Element prediction performances of different methods

		Case 1					Case 2			
Method	Performance	Wall	Column	Slab	Ceiling	Door	Wall	Column	Slab	Door
Prior knowledge (Romero-Jarén and Arranz 2021)	Accuracy[a]	0.71	1.00	1.00	1.00	0.71	0.67	0.80	1.00	0.67
	Runtime[b]			<5.0 s				<5.0 s		
Prior knowledge (Wang et al. 2015)	Accuracy[a]	1.00	0.71	1.00	1.00	1.00	1.00	0.78	1.00	1.00
	Runtime[b]			<5.0 s				<5.0 s		
3D deep learning	IoU	0.54	0	0.66	0.69	0	0.74	0	0	0
	Runtime[b]			~60 s				~60 s		
Proposed method	Accuracy[a]	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	IoU	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Runtime[b]			~20 s				~20 s		

Note: Processing times are based on using a computer with an Intel i7-8700 processor, 64 GB memory, and NVIDIA GeForce RTX 2080 Ti graphics card.

[a] Accuracy = (true positives + true negatives)/(true positives + true negatives + false positives + false negatives).

[b] Runtime in this table does not include the required time for 3D reconstruction.

our point clouds through the trained PointNet, we preprocessed the data by removing all possible noise. Fig. 22 summarizes the comparison results of the three methods based on Case 1 and Case 2.

Table 8 summarizes the identification performances of all methods. As mentioned, the results of the two prior knowledge methods are determined by the categories of isolated 3D planes, while the 3D deep-learning method identifies categories of every single point; thus, we use identification accuracies for the two prior knowledge-based methods and use IoU for the 3D deep-learning method. The comparison results demonstrate that our method is robust since the accuracies and IoUs of all components can reach 100%, while the other methods might fail to identify some components. For the first knowledge-based method (Romero-Jarén and Arranz 2021), all doors in both cases are recognized as walls, and columns in Case 2 are not recognized. These mispredictions occur for two main reasons: (1) it doesn't include the rule of how to segment doors; and (2 columns are identified based on the existence of ceilings and floors, but Case 2 doesn't have a ceiling. The second knowledge-based method (Wang et al. 2015) mispredicts all columns because it doesn't define the rule of column identification. Regarding the 3D deep learning method, some components are recognized but with low IoUs. For instance, the highest IoU among all components is 0.74 for the wall in Case 2. The IoUs of some components are even zero. The major cause of the poor performance is the lack of generalization, and the training point clouds are far different from the point clouds collected in our study. This drawback is a common issue for all deep-learning methods, but it has less impact on our approach since it is much easier to collect images for training DeepLab than collecting point clouds for training PointNet. In addition, some components are predicted as more than one category (e.g., the floor in Case 1 is partially classified as the ceiling), which is caused by the fact that the 3D deep-learning-based method does not include the plane segmentation step.

In terms of computational costs, the two prior knowledge-based methods have significant advantages compared with our method and the 3D deep-learning method because these two methods do not have the time-consuming procedure of processing data through trained networks. However, our method still consumes less time than the 3D deep-learning method due to the simplicity of 2D compared with 3D networks. It should be noted that the listed run times do not include the required time for 3D reconstruction. Nevertheless, all the selected methods were developed based on LiDAR data, which took longer times to be collected and processed than photogrammetry. Finally, when it comes to equipment costs, these selected methods are more expensive than ours since they use LiDAR scanners, while the necessary images for our method could be collected using available smartphones and digital cameras. According to Wang et al. (2020), the average cost of LiDAR scanners could be more than $50,000 USD while the associated costs of photogrammetry are usually less than $2,000 US.

Limitations

Despite the proposed method's promising performance, it does have some limitations: (1) The generalization of results is dependent on the selected 2D deep-learning method. Although DeepLab is a promising tool to conduct 2D semantic segmentation, the detection results are primarily determined based on the training data set. If one scenario (e.g., a cylindrical column) is not fully covered within the data set, it will not be correctly detected by the deep-learning method. Collecting a comprehensive data set that involves all possible scenarios (e.g., element type, shape, and geometry) with a sufficient number of images could be a tedious and time-consuming task. (2) The current practice in this research includes using the extreme value in each direction to define each object's boundary, which causes a difference between models and actual scanned objects. Boundary refinement will be one of the future works of this research. (3) A consequent issue is that the automation level decreases when scaling up dimensions in BIM models. Similar to the majority of the existing photogrammetric methods, manual measurement of at least one actual dimension is required to project distances from real-world cases to BIM models. Nevertheless, compared with the limitations of the existing methods (the prior knowledge-based method and 3D deep-learning-based method) discussed in the "Background" section, the minor work is still acceptable.

Concluding Remarks

In order to develop a robust system of generating as-built BIMs in various circumstances, this paper proposed an automated procedure based on a 2D deep-learning approach. Images were captured from various buildings to develop point clouds, which were then preprocessed to remove outliers and reduce data set size. As the next step, planes were isolated from point clouds by using the MSAC algorithm. A 2D deep learning method, DeepLab, was implemented for the semantic segmentation of images, and the results were mapped to isolated 3D planes based on the inverse photogrammetry method. The IFC4.0 schema was then utilized to generate as-built BIMs from point clouds with identified elements. Six case studies were selected to validate the feasibility of the proposed system. According to the results, three major lessons have been gained from this study:

- The proposed system is highly automated without any manual intervention.
- The proposed segmentation method for point clouds is very efficient. The DeepLab is promising for semantic segmentation in images since the accuracies of all categories on the test data set were greater than 96%, and the mean IoU was greater than 90%; however, combined with the inverse photogrammetry method, 39 out of 45 elements in point clouds have been correctly recognized.
- The IFC is an effective tool to generate as-built BIMs. All elements could be successfully identified and created based on the information extracted from point clouds. Meanwhile, the generated BIM models are compatible and interpretable through various BIM software platforms.

In summary, this paper contributes to the body of knowledge by introducing a 2D deep-learning-based method for automating the processes of BIM development. Our method is adaptive and can be applied to other scenarios with various components by enlarging the training data set and fine-tuning parameters. The study provides the AEC industry with an intelligent system to identify elements efficiently from point clouds and to generate as-built BIMs that are compatible with various BIM applications.

Future works involve four aspects: (1) The authors intend to create an image data set of buildings, including various elements. Although the proposed 2D deep-learning-based approach is more easily deployed than the 3D deep-learning-based approach in terms of collecting training data sets, significant effort is still required to cover many types of elements to make the proposed method as applicable as possible. (2) The second portion of future work will consider invisible elements, especially embedded rebar in concrete elements, during BIM development. Since the authors have conducted some works to identify rebar by using ground-penetrating radar and obtained promising results (Xiang et al. 2021), we plan to convert rebar locations into as-built BIMs by integrating the proposed workflow with rebar identification methods. (3) The third piece of future work will focus on upgrading the method of boundary determination when converting point clouds into BIM models. The current solution in our work is using extreme coordinates of each element to determine boundaries, which might lose geometrical details in BIM models. Wang et al. (2022b) fitted circles to retrieve lost points of pipes, which provides a potential solution to refine our point clouds' boundaries. We plan to explore this area in our future research. (4) The last aspect of future work is to extend our work to other applications (e.g., MEP, furniture). We are now focusing on building elements, but there are other facilities with important components during building operation and management. Our method will be fine-tuned to be adapted for new scenarios.

Data Availability Statement

Data generated or analyzed during the study are available from the corresponding author by request.

References

Aljumaily, H., D. F. Laefer, and D. Cuadra. 2017. "Urban point cloud mining based on density clustering and MapReduce." *J. Comput. Civ. Eng.* 31 (5): 04017021. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000674.

Anagnostopoulos, I., V. Pătrăucean, I. Brilakis, and P. Vela. 2016. "Detection of walls, floors, and ceilings in point cloud data." In *Proc., Construction Research Congress 2016*, 2302–2311. Mayagüez, Puerto Rico: Univ. of Puerto Rico-Mayagüez.

Armeni, I., S. Sax, A. R. Zamir, and S. Savarese. 2017. "Joint 2D-3D-semantic data for indoor scene understanding." Preprint, submitted Feburary 3, 2017. http://arxiv.org/abs/1702.01105.

Autodesk. 2021. "The overview of plant 3D." Accessed May 6, 2021. https://www.autodesk.com/products/autocad/included-toolsets/autocad-plant-3d.

Bae, H., M. Golparvar-Fard, and J. White. 2015. "Image-based localization and content authoring in structure-from-motion point cloud models for real-time field reporting applications." *J. Comput. Civ. Eng.* 29 (4): B4014008. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000392.

Bassier, M., B. Van Genechten, and M. Vergauwen. 2019. "Classification of sensor independent point cloud data of building objects using random forests." *J. Build. Eng.* 21 (Jan): 468–477. https://doi.org/10.1016/j.jobe.2018.04.027.

Bassier, M., and M. Vergauwen. 2020. "Unsupervised reconstruction of building information modeling wall objects from point cloud data." *Autom. Constr.* 120 (Dec): 103338. https://doi.org/10.1016/j.autcon.2020.103338.

Boulch, A., J. Guerry, B. Le Saux, and N. Audebert. 2018. "SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks." *Comput. Graphics* 71 (Apr): 189–198. https://doi.org/10.1016/j.cag.2017.11.010.

Braun, A., and A. Borrmann. 2019. "Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning." *Autom. Constr.* 106 (Oct): 102879. https://doi.org/10.1016/j.autcon.2019.102879.

Brilakis, I., H. Fathi, and A. Rashidi. 2011. "Progressive 3D reconstruction of infrastructure with videogrammetry." *Autom. Constr.* 20 (7): 884–895. https://doi.org/10.1016/j.autcon.2011.03.005.

BuildingSmart. 2021. "Industry foundation classes (IFC): An introduction." Accessed May 6, 2021. https://technical.buildingsmart.org/standards/ifc/.

Chen, C., X. Li, A. N. Belkacem, Z. Qiao, E. Dong, W. Tan, and D. Shin. 2019. "The mixed kernel function SVM-based point cloud classification." *Int. J. Precis. Eng. Manuf.* 20 (5): 737–747. https://doi.org/10.1007/s12541-019-00102-3.

Chen, L. C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 2017. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (4): 834–848. https://doi.org/10.1109/TPAMI.2017.2699184.

Chen, M., A. Feng, K. McCullough, P. B. Prasad, R. McAlinden, and L. Soibelman. 2020. "3D photogrammetry point cloud segmentation using a model ensembling framework." *J. Comput. Civ. Eng.* 34 (6): 04020048. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000929.

Dai, A., A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. 2017. "Scannet: Richly-annotated 3D reconstructions of indoor scenes." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 5828–5839. Redmond, WA: Microsoft.

Dai, F., A. Rashidi, I. Brilakis, and P. Vela. 2013. "Comparison of image-based and time-of-flight-based technologies for three-dimensional reconstruction of infrastructure." *J. Constr. Eng. Manage.* 139 (1): 69–79. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000565.

De Geyter, S., J. Vermandere, H. De Winter, M. Bassier, and M. Vergauwen. 2022. "Point cloud validation: On the impact of laser scanning technologies on the semantic segmentation for BIM modeling and evaluation." *Remote Sens.* 14 (3): 582. https://doi.org/10.3390/rs14030582.

Díaz-Vilariño, L., B. Conde, S. Lagüela, and H. Lorenzo. 2015. "Automatic detection and segmentation of columns in as-built buildings from point clouds." *Remote Sens.* 7 (11): 15651–15667. https://doi.org/10.3390/rs71115651.

Dorninger, P., and C. Nothegger. 2007. "3D segmentation of unstructured point clouds for building modeling." *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* 35 (3/W49A): 191–196.

Ester, M., H. P. Kriegel, J. Sander, and X. Xu. 1996. "A density-based algorithm for discovering clusters in large spatial databases with noise." In Vol. 96 of *Proc., of the Second Int. Conf. on Knowledge Discovery and Data Mining*, 226–231. Washington, DC: Association for the Advancement of Artificial Intelligence.

Farid, R., and C. Sammut. 2012. "A relational approach to plane-based object categorization." In *Proc., RSS 2012 Workshop on RGB-D Cameras*. Canberra, ACT, Australia: Commonwealth Scientific and Industrial Research Organisation.

FARO. 2021. "User manual for pointsense building." Accessed May 7, 2021. https://knowledge.faro.com/Software/Legacy-Software/Legacy-PointSense_and_CAD_Plugins/PointSense/PointSense_Building/User_Manual_for_PointSense_Building.

Gené-Mola, J., R. Sanz-Cortiella, J. R. Rosell-Polo, A. Escola, and E. Gregorio. 2021. "In-field apple size estimation using photogrammetry-derived 3D point clouds: Comparison of 4 different methods considering fruit occlusions." *Comput. Electron. Agric.* 188 (Dec): 106343. https://doi.org/10.1016/j.compag.2021.106343.

Golparvar-Fard, M., J. Bohn, J. Teizer, S. Savarese, and F. Peña-Mora. 2011. "Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques." *Autom. Constr.* 20 (8): 1143–1155. https://doi.org/10.1016/j.autcon.2011.04.016.

Guo, Y., H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. 2020. "Deep learning for 3D point clouds: A survey." *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (12): 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434.

Hamledari, H., E. Rezazadeh Azar, and B. McCabe. 2018. "IFC-based development of as-built and as-is BIMs using construction and facility inspection data: Site-to-BIM data transfer automation." *J. Comput. Civ. Eng.* 32 (2): 04017075. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000727.

Hichri, N., C. Stefani, L. De Luca, P. Veron, and G. Hamon. 2013. "From point cloud to BIM: A survey of existing approaches." In *Proc., 24th Int. CIPA Symp*. Stockholm, Sweden: Hexagon.

Jung, J., C. Stachniss, S. Ju, and J. Heo. 2018. "Automated 3D volumetric reconstruction of multiple-room building interiors for as-built BIM." *Adv. Eng. Inf.* 38 (Oct): 811–825. https://doi.org/10.1016/j.aei.2018.10.007.

Kazi, A., A. Sausthanmath, S. M. Meena, S. V. Gurlahosur, and U. Kulkarni. 2020. "Detection of holes in 3D architectural models using shape classification based Bubblegum algorithm." *Procedia Comput. Sci.* 167: 1684–1695. https://doi.org/10.1016/j.procs.2020.03.379.

Kim, H., and C. Kim. 2021. "3D as-built modeling from incomplete point clouds using connectivity relations." *Autom. Constr.* 130 (Oct): 103855. https://doi.org/10.1016/j.autcon.2021.103855.

Kim, H., Z. Shen, I. Kim, K. Kim, A. Stumpf, and J. Yu. 2016. "BIM IFC information mapping to building energy analysis (BEA) model with manually extended material information." *Autom. Constr.* 68 (Aug): 183–193. https://doi.org/10.1016/j.autcon.2016.04.002.

Kim, P., J. Chen, and Y. K. Cho. 2018. "Automated point cloud registration using visual and planar features for construction environments." *J. Comput. Civ. Eng.* 32 (2): 04017076. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000720.

Kong, D., L. Xu, and X. Li. 2013. "A new method for building roof segmentation from airborne LiDAR point cloud data." *Meas. Sci. Technol.* 24 (9): 095402. https://doi.org/10.1088/0957-0233/24/9/095402.

Lai, H., X. Deng, and T. Y. P. Chang. 2019. "BIM-based platform for collaborative building design and project management." *J. Comput. Civ. Eng.* 33 (3): 05019001. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000830.

Landrieu, L., and M. Simonovsky. 2018. "Large-scale point cloud semantic segmentation with superpoint graphs." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 4558–4567. Mountain View, CA: Google.

Lee, J. S., J. Park, and Y. M. Ryu. 2021. "Semantic segmentation of bridge components based on hierarchical point cloud model." *Autom. Constr.* 130 (Oct): 103847. https://doi.org/10.1016/j.autcon.2021.103847.

Li, X., P. Wu, G. Q. Shen, X. Wang, and Y. Teng. 2017. "Mapping the knowledge domains of building information modeling (BIM): A bibliometric approach." *Autom. Constr.* 84 (Dec): 195–206. https://doi.org/10.1016/j.autcon.2017.09.011.

Long, J., E. Shelhamer, and T. Darrell. 2015. "Fully convolutional networks for semantic segmentation." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 3431–3440. Mountain View, CA: Google.

Lu, R., and I. Brilakis. 2019. "Digital twinning of existing reinforced concrete bridges from labeled point clusters." *Autom. Constr.* 105 (Sep): 102837. https://doi.org/10.1016/j.autcon.2019.102837.

Lu, R., I. Brilakis, and C. R. Middleton. 2019. "Detection of structural components in point clouds of existing RC bridges." *Comput.-Aided Civ. Infrastruct. Eng.* 34 (3): 191–212. https://doi.org/10.1111/mice.12407.

Ma, J. W., T. Czerniawski, and F. Leite. 2020. "Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds." *Autom. Constr.* 113 (May): 103144. https://doi.org/10.1016/j.autcon.2020.103144.

Ma, Z., and S. Liu. 2018. "A review of 3D reconstruction techniques in civil engineering and their applications." *Adv. Eng. Inf.* 37 (Aug): 163–174. https://doi.org/10.1016/j.aei.2018.05.005.

Macher, H., L. Roy, and T. Landes. 2021. "Automation of windows detection from geometric and radiometric information of point clouds in a scan-to-BIM process." In Vol. 43 of *Proc., 24th ISPRS Congress (2021 edition)*. Blagnac, France: AIRBUS.

Matrone, F., A. Lingua, R. Pierdicca, E. S. Malinverni, M. Paolanti, E. Grilli, F. Remondino, A. Murtiyoso, and T. Landes. 2020. "A benchmark for large-scale heritage point cloud semantic segmentation." *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLIII-B2-2020: 1419–1426. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1419-2020.

Maturana, D., and S. Scherer. 2015. "Voxnet: A 3D convolutional neural network for real-time object recognition." In *Proc., 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 922–928. New York: IEEE.

Minaee, S., Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos. 2021. "Image segmentation using deep learning: A survey." *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (7): 3523–3542. https://doi.org/10.1109/TPAMI.2021.3059968.

Mo, K., S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. 2019. "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding." In *Proc., IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 909–918. Redmond, WA: Microsoft.

Munoz, D., N. Vandapel, and M. Hebert. 2008. "Directional associative markov network for 3-d point cloud classification." In *Proc., 2008 4th Int. Symp. on 3D Data Processing, Visualization and Transmission*, 65–72. Alexandria, VA: National Science Foundation.

Ni, H., X. Lin, and J. Zhang. 2017. "Classification of ALS point cloud with improved point cloud segmentation and random forests." *Remote Sens.* 9 (3): 288. https://doi.org/10.3390/rs9030288.

Nikoohemat, S., A. A. Diakité, S. Zlatanova, and G. Vosselman. 2020. "Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management." *Autom. Constr.* 113 (May): 103109. https://doi.org/10.1016/j.autcon.2020.103109.

Ning, X., X. Zhang, Y. Wang, and M. Jaeger. 2009. "Segmentation of architecture shape information from 3D point cloud." In *Proc., 8th Int. Conf. on Virtual Reality Continuum and Its Applications in Industry*, 127–132. New York: Special Interest Group on Computer Graphics.

Ochmann, S., R. Vock, and R. Klein. 2019. "Automatic reconstruction of fully volumetric 3D building models from oriented point clouds." *ISPRS J. Photogramm. Remote Sens.* 151 (May): 251–262. https://doi.org/10.1016/j.isprsjprs.2019.03.017.

Paiva, P. V., C. K. Cogima, E. Dezen-Kempter, and M. A. Carvalho. 2020. "Historical building point cloud segmentation combining hierarchical watershed transform and curvature analysis." *Pattern Recognit. Lett.* 135 (Jul): 114–121. https://doi.org/10.1016/j.patrec.2020.04.010.

Park, J., J. Kim, D. Lee, K. Jeong, J. Lee, H. Kim, and T. Hong. 2022. "Deep learning–based automation of scan-to-BIM with modeling objects from occluded point clouds." *J. Manage. Eng.* 38 (4): 04022025. https://doi.org/10.1061/(ASCE)ME.1943-5479.0001055.

Perez-Perez, Y., M. Golparvar-Fard, and K. El-Rayes. 2021. "Scan2BIM-NET: Deep learning method for segmentation of point clouds for scan-to-BIM." *J. Constr. Eng. Manage.* 147 (9): 04021107. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002132.

Petra Gartzen. 2022. "BIM—Evolution from stone age to digital age." Accessed May 16, 2022. https://cambashi.com/wp-content/uploads/2022/02/BIM-%E2%80%93-evolution-from-Stone-Age-to-Digital-Age-1.pdf.

Phan, A. V., M. Le Nguyen, Y. L. H. Nguyen, and L. T. Bui. 2018. "DGCNN: A convolutional neural network over large-scale labeled graphs." *Neural Networks* 108 (Dec): 533–543. https://doi.org/10.1016/j.neunet.2018.09.001.

Pierdicca, R., M. Paolanti, F. Matrone, M. Martini, C. Morbidoni, E. S. Malinverni, E. Frontoni, and A. M. Lingua. 2020. "Point cloud semantic segmentation using a deep learning framework for cultural heritage." *Remote Sens.* 12 (6): 1005. https://doi.org/10.3390/rs12061005.

Pleansamai, K., and K. Chaiyasarn. 2019. "M-estimator sample consensus planar extraction from image-based 3D point cloud for building information modeling." *GEOMATE J.* 17 (63): 69–76. https://doi.org/10.21660/2019.63.09667.

Qi, C. R., H. Su, K. Mo, and L. J. Guibas. 2017a. "Pointnet: Deep learning on point sets for 3D classification and segmentation." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 652–660. Redmond, WA: Microsoft.

Qi, C. R., L. Yi, H. Su, and L. J. Guibas. 2017b. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Preprint, submitted June 7, 2017. http://arxiv.org/abs/1706.02413.

Rashidi, A., I. Brilakis, and P. Vela. 2015. "Generating absolute-scale point cloud data of built infrastructure scenes using a monocular camera setting." *J. Comput. Civ. Eng.* 29 (6): 04014089. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000414.

Rashidi, A., F. Dai, I. Brilakis, and P. Vela. 2013. "Optimized selection of key frames for monocular videogrammetric surveying of civil infrastructure." *Adv. Eng. Inf.* 27 (2): 270–282. https://doi.org/10.1016/j.aei.2013.01.002.

Rashidi, A., H. Fathi, and I. Brilakis. 2011. "Innovative stereo vision-based approach to generate dense depth maps of transportation infrastructure." *Transp. Res. Rec.* 2215 (1): 93–99. https://doi.org/10.3141/2215-10.

Rebolj, D., Z. Pučko, N. Č. Babič, M. Bizjak, and D. Mongus. 2017. "Point cloud quality requirements for Scan-vs-BIM based automated construction progress monitoring." *Autom. Constr.* 84 (Dec): 323–334. https://doi.org/10.1016/j.autcon.2017.09.021.

Romero-Jarén, R., and J. J. Arranz. 2021. "Automatic segmentation and classification of BIM elements from point clouds." *Autom. Constr.* 124 (Apr): 103576. https://doi.org/10.1016/j.autcon.2021.103576.

Ronneberger, O., P. Fischer, and T. Brox. 2015. "U-net: Convolutional networks for biomedical image segmentation." In *Proc., Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, 234–241. Cham, Switzerland: Springer.

Sanhudo, L., N. M. Ramos, J. P. Martins, R. M. Almeida, E. Barreira, M. L. Simões, and V. Cardoso. 2020. "A framework for in-situ geometric data acquisition using laser scanning for BIM modeling." *J. Build. Eng.* 28 (Mar): 101073. https://doi.org/10.1016/j.jobe.2019.101073.

Shen, X., and I. Stamos. 2021. "3D object detection and instance segmentation from 3D range and 2D color images." *Sensors* 21 (4): 1213. https://doi.org/10.3390/s21041213.

Son, H., C. Kim, and C. Kim. 2015. "3D reconstruction of as-built industrial instrumentation models from laser-scan data and a 3D CAD database based on prior knowledge." *Autom. Constr.* 49 (Part B): 193–200. https://doi.org/10.1016/j.autcon.2014.08.007.

Subbarao, R., and P. Meer. 2006. "Beyond RANSAC: User independent robust regression." In *Proc., 2006 Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, 101–101. New York: IEEE.

Sun, J., P. Olsson, H. Eriksson, and L. Harrie. 2020. "Evaluating the geometric aspects of integrating BIM data into city models." *J. Spatial Sci.* 65 (2): 235–255. https://doi.org/10.1080/14498596.2019.1636722.

Tang, S., X. Li, X. Zheng, B. Wu, W. Wang, and Y. Zhang. 2022. "BIM generation from 3D point clouds by combining 3D deep learning and improved morphological approach." *Autom. Constr.* 141 (Sep): 104422. https://doi.org/10.1016/j.autcon.2022.104422.

Tang, S., D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao. 2020. "BIM assisted building automation system information exchange using BACnet and IFC." *Autom. Constr.* 110 (Feb): 103049. https://doi.org/10.1016/j.autcon.2019.103049.

Teizer, J., and T. Kahlmann. 2007. "Range imaging as emerging optical three-dimension measurement technology." *Transp. Res. Rec.* 2040 (1): 19–29. https://doi.org/10.3141/2040-03.

TEJJY. 2021. "Point cloud to BIM conversion: A reality check." Accessed May 16, 2022. https://www.tejjy.com/point-cloud-laser-scan-to-bim-modeling-a-reality-check/.

Torr, P. H., and A. Zisserman. 1997. "Robust parameterization and computation of the trifocal tensor." *Image Vis. Comput.* 15 (8): 591–605. https://doi.org/10.1016/S0262-8856(97)00010-3.

Trimble. 2022. "Trimble X12 3D laser scanning system." Accessed May 16, 2022. https://geospatial.trimble.com/products-and-solutions/trimble-x12.

United-BIM. 2022. "A walk-through of point cloud to BIM with a real life project." Accessed May 16, 2022. https://www.united-bim.com/walk-through-of-point-cloud-to-bim-process/.

Wang, B., Q. Wang, J. C. Cheng, C. Song, and C. Yin. 2022a. "Vision-assisted BIM reconstruction from 3D LiDAR point clouds for MEP scenes." *Autom. Constr.* 133 (Jan): 103997. https://doi.org/10.1016/j.autcon.2021.103997.

Wang, B., Q. Wang, J. C. Cheng, and C. Yin. 2022b. "Object verification based on deep learning point feature comparison for scan-to-BIM." *Autom. Constr.* 142 (Oct): 104515. https://doi.org/10.1016/j.autcon.2022.104515.

Wang, C., Y. K. Cho, and C. Kim. 2015. "Automatic BIM component extraction from point clouds of existing buildings for sustainability applications." *Autom. Constr.* 56 (Aug): 1–13. https://doi.org/10.1016/j.autcon.2015.04.001.

Wang, Q. 2019. "Automatic checks from 3D point cloud data for safety regulation compliance for scaffold work platforms." *Autom. Constr.* 104 (Aug): 38–51. https://doi.org/10.1016/j.autcon.2019.04.008.

Wang, Q., and M. K. Kim. 2019. "Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018." *Adv. Eng. Inf.* 39 (Jan): 306–319. https://doi.org/10.1016/j.aei.2019.02.007.

Wang, Q., Y. Tan, and Z. Mei. 2020. "Computational methods of acquisition and processing of 3D point cloud data for construction applications." *Arch. Comput. Methods Eng.* 27 (2): 479–499. https://doi.org/10.1007/s11831-019-09320-4.

Xiang, Z., G. Ou, and A. Rashidi. 2020. "Integrated approach to simultaneously determine 3D location and size of rebar in GPR data." *J. Perform. Constr. Facil.* 34 (5): 04020097. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001502.

Xiang, Z., G. Ou, and A. Rashidi. 2021. "Robust cascaded frequency filters to recognize rebar in GPR data with complex signal interference." *Autom. Constr.* 124 (Apr): 103593. https://doi.org/10.1016/j.autcon.2021.103593.

Xiao, Y., Y. Taguchi, and V. R. Kamat. 2018. "Coupling point cloud completion and surface connectivity relation inference for 3D modeling of indoor building environments." *J. Comput. Civ. Eng.* 32 (5): 04018033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000776.

Xiong, Z., and T. Wang. 2021. "Research on BIM reconstruction method using semantic segmentation point cloud data based on PointNet." In Vol. 719 of *Proc., IOP Conf. Series: Earth and Environmental Science*, 022042. Bristol, UK: IOP Publishing.

Xu, Y., X. Tong, and U. Stilla. 2021. "Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry." *Autom. Constr.* 126 (Jun): 103675. https://doi.org/10.1016/j.autcon.2021.103675.

Xue, F., W. Lu, K. Chen, and C. J. Webster. 2019. "BIM reconstruction from 3D point clouds: A semantic registration approach based on multimodal optimization and architectural design knowledge." *Adv. Eng. Inf.* 42 (Oct): 100965. https://doi.org/10.1016/j.aei.2019.100965.

Xue, F., W. Lu, Z. Chen, and C. J. Webster. 2020. "From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles." *ISPRS J. Photogramm. Remote Sens.* 167 (Sep): 418–431. https://doi.org/10.1016/j.isprsjprs.2020.07.020.

Yang, L., J. C. Cheng, and Q. Wang. 2020. "Semi-automated generation of parametric BIM for steel structures based on terrestrial laser scanning data." *Autom. Constr.* 112 (Apr): 103037. https://doi.org/10.1016/j.autcon.2019.103037.

Yin, C., B. Wang, V. J. Gan, M. Wang, and J. C. Cheng. 2021. "Automated semantic segmentation of industrial point clouds using ResPointNet++." *Autom. Constr.* 130 (Oct): 103874. https://doi.org/10.1016/j.autcon.2021.103874.

Zhan, Q., Y. Liang, and Y. Xiao. 2009. "Color-based segmentation of point clouds." *Laser Scanning* 38 (3): 155–161.

Zhang, J., X. Lin, and X. Ning. 2013. "SVM-based classification of segmented airborne LiDAR point clouds in urban areas." *Remote Sens.* 5 (8): 3749–3775. https://doi.org/10.3390/rs5083749.

Zhang, J., X. Zhao, Z. Chen, and Z. Lu. 2019. "A review of deep learning-based semantic segmentation for point cloud." *IEEE Access* 7 (Dec): 179118–179133. https://doi.org/10.1109/ACCESS.2019.2958671.

Zhang, L., J. Sun, and Q. Zheng. 2018. "3D point cloud recognition based on a multi-view convolutional neural network." *Sensors* 18 (11): 3681. https://doi.org/10.3390/s18113681.