Matching Design-Intent Planar, Curved, and Linear Structural Instances in Point Clouds

Zhiqi Hu*, Ioannis Brilakis

*Corresponding Author

Department of Engineering, University of Cambridge, Cambridge, UK

Abstract

The lack of timely progress monitoring and quality control contributes to cost-escalation, lowering of productivity, and broadly poor project performance. This paper addressed the challenge of high-precision structural instance segmentation from point clouds by leveraging as-designed IFC models in Scan-vs-BIM contexts. We proposed an automatic method to segment the entire points corresponding to the as-designed instance. The workflow contains: 1) Instance descriptor generation; 2) PROSAC-based shape detection; 3) DBSCAN-based cluster optimization. The method matches design-intent planar, curved, and linear structural instances in complex scenarios including: 1) the as-built point cloud is noisy with high occlusions and clutter; 2) deviations between as-built instances and as-designed models in terms of position, orientation, and scale; 3) both Manhattan-World and non-Manhattan-World instances. The experimental results from five diverse real-world datasets showed excellent performance with mPrecision 0.962, mRecall 0.934, and mIoU 0.914. Benchmarking against state-of-the-art methods showed that the proposed method outperforms all existing ones.

Keywords

Digital Twins, Building Information Modelling (BIM), Object Detection, Instance Segmentation, Scan-vs-BIM, IFC model, Point Clouds, Structural Objects, Design Intent (DI)

1. Introduction

This research is about matching design-intent planar, curved, and linear structural objects in point clouds to maintain geometric building Digital Twins (DTs). By matching, we refer here to detect and segment object instances from point clouds into point clusters. By Design Intent (DI), we refer here to the client-approved, final as-designed model used as a benchmark at the construction stage [55]. In this paper, we focus on Industry Foundation Classes (IFC) models that serve as a standardised digital description of buildings [56]. By planar, curved, and linear structural objects, we refer here to the top frequent structural object classes ranked by [1], namely, planar and curved walls, slabs including floors and ceilings, beams, and columns. By point clouds, we refer here to data sets with millions of points made of XYZ coordinates [57]. By maintaining, we refer here to keeping DT's geometry dynamically updated with the assistance of DI to reflect the as-is statuses of a building at different timestamps during the construction stage [1]. By a geometric building DT, we refer here to a product information repository for storing and sharing physical and functional properties of a building over time with all Architectural, Engineering, and Construction (AEC) stakeholders throughout its

lifecycle [2]. A DT differs from a Building Information Model (BIM). A BIM only provides product information and can be updated at various timestamps throughout the life cycle of a DT [1]. By Scan-vs-BIM, we refer here to a process system that aligns scanned Point Cloud Data (PCD) with as-designed BIM models to compare and recognize object instances to support construction progress monitoring and quality control [3].

The lack of timely progress monitoring and prompt quality control are two of the problems plaguing current building construction projects, leading to poor construction project performance [58]. Over 50% of construction companies have suffered one or more underperforming projects in recent years [63]. Only a quarter of construction projects managed to stay within 10% of their initially planned deadlines [63]. Many construction projects exceed their budget. Specifically, 69% surpass their budget by over 10%, while only 31% stay within 10% of their initial estimate [62]. Executing large projects on time and within budget is typically a challenge for the construction business [4]. The construction industry continues to be one of the least digitised industries in comparison to media, finance, and other industries [59]. The AEC industry can benefit from digital technologies, including BIM and DT, with up to a 50% boost in field efficiency, a 10% acceleration in timelines, and an 80% decrease in modifications [61]. It is essential to digitise and automate the design, construction, operation, and refurbishment of buildings to enhance their efficiency and performance with the help of DTs [58, 60].

Using a DT to facilitate building progress monitoring and quality control still lacks automation in the current state of practice, leading to delayed feedback. For example, state-of-the-art commercial products, such as OpenSpace [64] and Buildots [65], enable image comparison methods to support progress monitoring. For this to be possible, a worker needs to wear a helmet equipped with a 360° camera and move around a construction site to record a video. Then, this two-dimensional (2D) data stream is compared with the DI to update the status of the project. However, these systems only enable visual inspections and cannot directly update three-dimensional (3D) geometric data. For example, they cannot retrieve the thickness of a wall or a window directly from images. Also, manual effort is still required for quality monitoring and control. In summary, the available commercial products fall short of the high demand for a higher degree of automation and level of resolution in the maintenance of a geometric building DT at different timestamps during the construction stage.

Three stages are required for maintaining a building geometric DT [4]: 1) As-designed BIM model to scanned PCD registration ensures that the DI model (e.g., IFC model) is aligned with the as-built PCD into the same coordinate system. 2) Matching DI object instances in the as-built PCD aims to detect and segment instances from the PCD with the help of the DI. 3) 3D representation from the extracted PCD converts the points into information-rich meshes and updates the meshes into the DT. The goal of the first stage is to determine the rigid transformation matrix to align BIM with PCD and it has been well studied. Random Sample Consensus (RANSAC)-based [82] methods are commonly used to (semi-)automate coarse registration [5-8]. Fine registration is always applied after coarse registration to obtain a more accurate result and most well-established methods were derived from the Iterative Closest Point (ICP) algorithm [9-12]. Software including Recap and CloudCompare are also capable of registration in practice. However, the second step is more complex and time-consuming than the first step. The current research including mapping points to the model's surface [21-26, 75] and RANSAC-based shape extraction [28, 29, 77-81] still have limitations in real and complex environments. The comprehensive literature review towards this stage will be conducted in Section 2 Background. Finally, the third stage has also been well-explored. The methods of mesh reconstruction developed by [66, 67] are effective in

generating detailed representations from PCD. Rashidi and Brilakis [68] also summarised the methods for filling gaps in PCD, which can be used to improve the performance of meshing.

In this paper, we focus on the second stage and propose an innovative, robust method to automatically detect and segment top frequent building structural objects from PCD with the assistance of DI in a real, complex context. The selected structural objects are built in planar, curved, and linear shapes. We need to detect and segment as-built instances before creating and assigning as-built meshes to a geometric DT to keep it updated. Keeping the geometry updated can help monitor the progress and control the quality at different timestamps during the construction stage. This paper highlights the following contributions in particular:

1. In the Scan-vs-BIM system, many existing approaches only focus on object detection to monitor the construction progress. However, the proposed method herein can not only detect but also segment instances from PCD to offer the entire points corresponding to the instances, where the whole extracted point cluster can help with quality assurance during the building construction stage.

2. Our proposed method is robust on the real, noisy PCD with significant occlusions and clutter, as opposed to most present methods which are only proven on synthetic or simple datasets that are clean and complete. For instance, temporarily stored building materials or workers moving in front of a wall may be scanned into the PCD as noisy points to occlude the wall; using current approaches may result in inadequate or irrelevant point extraction, which cannot accurately depict the as-built geometry.

3. Our proposed method can be used in more complex and real environments where there are distinct deviations in position, orientation, and scale between the DI geometry and the as-built instances, in contrast to most state-of-the-art Scan-vs-BIM methods that fully depend on the DI to detect or segment object instances from PCD. Our method supports progress monitoring and quality control in the real world by leveraging the DI model without being fully dependent on it.

4. While most of the current methods only focus on cuboid or cylindrical objects, our proposed method is designed for the most frequent structural objects in various shapes including planar, curved, and linear shapes.

5. While most of the current methods only focus on Manhattan-World buildings, our proposed method can also deal with non-Manhattan-World multi-storey buildings. For example, the proposed method is also robust when walls are not aligned with a horizontal axis, or when walls are curved.

The rest of this paper is organised as follows: the background, including the literature review, gaps in knowledge, and objectives, is reviewed in section 2; the proposed method, including the workflow, the details, and the pseudo-code, is introduced in section 3; the experiments and results are shown in section 4; discussion and conclusions are provided in sections 5 and 6.

2. Background

In this paper, we focus on the advancements in instance matching in the Scan-vs-BIM context. We aim to detect and segment the structural object instances from PCD by leveraging the DI models. This is a crucial step as the second stage in the maintenance of a

building's geometric DT. The state of research, including instance detection and segmentation, data fitting and clustering, IFC schema, and object descriptor, is discussed in detail in the following subsections before the knowledge gaps and research objectives are summarised.

2.1 Instance Detection and Segmentation

Object detection refers to identifying the location and class of each object instance while object segmentation refers to cutting the whole PCD down to the object instance level. They are crucial steps to generate and update a DT in a 3D environment for many applications, including construction project management and heritage building operation. Structural object detection and segmentation focus on walls, slabs, columns, and beams in buildings.

In Scan-vs-BIM, we investigated five types of methods for instance detection and segmentation: point-to-point, Hough transform, point-to-surface, feature-based, and RANSAC-based methods. Point-to-point matching was initially developed by [17] to detect points corresponding to a DI instance in the scanned PCD. The performance was evaluated by calculating the ratio of retrieved as-designed points to the total number of as-designed points. The threshold ratio was set as 50% to assess the retrieval result on small-scale datasets (4 columns and 1 slab, each within 18,000 points). This method was then adopted to monitor the progress of construction projects, by detecting primary and temporary structural objects [18, 19, 20] and mechanical objects [3]. In [70], the authors also applied the point-to-point method for automatic deviation detection for columns and beams. This method performs well in tracking the existing status of objects when there are few deviations between the as-designed and the as-built. However, since the retrieval result fully depends on the threshold value setting, the false-negative results will increase when the as-built objects have large spatial deviations against the DI. Also, the false positive will occur if part of another instance is at the same location with the covered points exceeding the threshold. Hough transform [71] maps edge points in image space to parameter space for shape detection. It performs well in line and circle detection with outliers. In [72, 73], the authors applied 2D Hough transform by projecting resampled 3D points into circle slices via the normal orientation to detect cylindrical pipes. However, its application demands a consistent as-built position and dimension with the DI geometry. In [74], the authors enhanced this approach by incorporating point-to-point comparison to detect out-of-pace instances and identify the instance completeness. Still, the Hough transform suffers from high storage and computational costs, struggles with high occlusions, and presumes predominant orthogonality in cylindrical object instances, making it less robust in complex environments. The third type of method, point-to-surface matching, computes the overlapping area between the PCD and the model directly, as demonstrated by [21, 22]. It is also used for deviation analysis between the DI and the as-built objects [75]. In [23], the authors improved the method by developing a surface coverage ratio calculation algorithm using alpha shape reconstruction. Other researchers used Euclidean distance to determine the nearest point to the model's surface for instance detection [24, 25, 26]. However, this method struggles to recognize all object points in cluttered PCDs or when the PCD and DI geometry deviate significantly. Their validity is compromised if as-built deviations surpass a manually set coverage ratio. A further approach takes advantage of an object's features, such as position, size, normal, and continuity, for instance detection. In [27], the authors used Lalonde, orientation, and continuity to identify instances, but the method presupposes all instances are DI-compliant. In [76], the authors used five features including length, size, colour, orientation, and the number of connections with adjacent object instances to test the prefabricated pipes in an environment without any occlusion and clutter. In [54], the authors developed a 3D eigenvector-based shape descriptor

using voxels for point cluster matching, but this method requires the PCD has few occlusions and clutter. In [15], the authors used the eigenvalues and shape histograms of the PCD for cluster matching, yet this method primarily localises different point clusters without yielding precising instance segmentation results. In [22], the authors computed the probability distribution of the PCD and the model's geometric attributes to match objects, but this requires the denoised PCD without any occlusion. All these methods may not work effectively when the PCD deviates from the DI geometry or when the PCD is cluttered significantly. On the other hand, RANSAC-based methods have been demonstrated to be more effective in instance segmentation in the Scan-vs-BIM context [77]. In [78], the author applied RANSAC to optimise the edge points for quality assurance of the full-scale precast concrete slabs. In [79], the authors used a normal-based region growing method with RANSAC to detect cylindrical pipes when the position and orientation of the as-built differ from the DI. In [28] and [29], the authors applied RANSAC and its variant (MLESAC) to segment cuboid-shaped instances and cast-in-place footing. In [80], the authors applied PCA to estimate the normal vector from PCD and RANSAC to estimate planes with different orientations. However, the experimental data is a separate bathroom and an office room with no connected spaces. It is difficult to identify instances after plane extraction for multi-space buildings. In [81], the authors proposed a slicing method with RANSAC for curved façade and window extraction from PCD. RANSAC-based methods primarily detect primitive shapes like cylinders and planes but struggle with complex structures like T-shaped joints or sprinklers. Additionally, the absence of verification or optimization steps can result in inaccuracies in cluttered and occluded environments. Overall, Table 1 summarises the five types of methods and highlights the common limitations of these state-of-the-art (SOTA) methods.

Table 1 Instance detection and segmentation methods and their limitations in Scan-vs-BIM.

Method	**Reference**	**Common Limitations**
Point-to-Point	[3, 17 – 20, 70]	sensitive to the threshold value setting
Hough transform	[72 – 74]	high storage and computational costs
Point-to-Surface	[21 – 26, 75]	few deviations between PCD and DI
Feature-based	[15, 22, 27, 54, 76]	PCD with few occlusions and clutter
RANSAC-based	[28, 29, 77 - 81]	object instances with primitive shapes

In the relevant area of Scan-to-BIM, deep learning has been widely used for semantic and instance segmentations. Semantic segmentation aims to assign a semantic label to each point in PCD, where points with the same semantic label belong to the same category. Instance segmentation, on the other hand, involves identifying and segmenting individual object instances with a unique label assigned. Scan-to-BIM aims to convert PCD into a BIM representation, which is pivotal when a pre-existing BIM of a building is absent. Scan-to-BIM methods are widely employed for retrofitting projects, historic preservation, and facility management. It can be a foundational step for Scan-vs-BIM by creating a digital representation of a building. This digital model can then be compared against an updated PCD to support progress monitoring and quality control during construction. Many networks have been developed to solve the semantic segmentation problem in the Scan-to-BIM context without support from DI models. PointNet [30] was the first network proposed for processing points in the point cloud. Its key principle is to learn a permutation invariant function that maps an unordered set of points to a fixed-size feature vector. In [31], the authors developed

an end-to-end trainable multi-view aggregation model by merging features from images into 3D points. The method is robust for large-scale indoor or outdoor semantic segmentation on the S3DIS benchmark (Stanford 3D Indoor Scene Dataset). In [32], the authors proposed a new window-normalization method by unifying the point densities in different parts to improve the segmentation performance. As for instance segmentation, PointCNN [33] was first proposed to segment points by computing a feature transformation matrix based on local geometric information of neighbouring points. Table 2 and Table 3 summarise the performance of the SOTA deep-learning models for semantic and instance segmentation. The SOTA model can only achieve around 77% mean Intersection over Union (mIoU) for semantic segmentation, and around 75% mean Precision (mPrec) and 72% mean Recall (mRec) separately for instance segmentation. These numbers show that the current SOTA algorithms cannot be directly applied to support construction management due to their relatively lower accuracy level. Higher precision and recall are necessary to support quality control in construction. Elevated precision in instance segmentation is imperative to minimize the inclusion of extraneous points, while enhanced recall ensures comprehensive retrieval of relevant points correlating with the model. Consequently, the segmented point cluster can exhibit superior efficacy in representing the current state of a given instance. In [34], the authors combined deep learning and void-growing approaches together to create digital twins of Manhattan-world buildings with a higher mIoU. Although Scan-to-BIM methods are versatile to support progress monitoring and quality control, some additional steps are still required. For example, a new BIM needs to be created regularly from the updated PCD to support comparison between two BIMs at different timestamps. Therefore, Scan-to-BIM methods cannot be directly adopted to support progress monitoring and quality control due to the lack of matching results between DI and PCD.

Table 2 SOTA Deep-learning semantic segmentation evaluated on S3DIS.

Model	**mIoU**	**Reference**
WindowNorm+StratifiedTransformer	0.776	[32]
PointMetaBase-XXL	0.770	[35]
PointNeXt-XL	0.749	[36]
PointNet	0.476	[30]

Table 3 SOTA Deep-learning Instance segmentation evaluated on S3DIS.

Model	**mPrec**	**mRec**	**Reference**
TD3D	0.868	0.766	[37]
MaskGroup	0.666	0.696	[38]
SPFormer	0.740	0.711	[39]
SoftGroup	0.753	0.698	[40]

In conclusion, structural object detection and segmentation in PCD is a rapidly growing field with significant potential for a wide range of applications. However, current SOTA methods for Scan-vs-BIM have common limitations (Table 1) that prevent them from being used on large-scale real-world applications. Despite the limitations and challenges that currently exist, ongoing research aimed at improving the performance and versatility of Scan-vs-BIM methods will likely result in continued advancements in this area.

2.2 Other Data Fitting and Clustering Methods

Besides RANSAC, its derivatives are widely used for dealing with situations where the underlying dataset contains a substantial number of outliers while attempting to estimate model parameters. These derivatives are applied to fit shapes and detect instances in PCD as model-driven algorithms. M-estimator Sample Consensus (MSAC) [83] minimizes a cost function that exhibits reduced sensitivity to outliers. In [41, 84, 85], the authors applied MSAC to extract planes for PCD segmentation. MSAC allows for a more seamless transition between inliers and outliers when compared to the rigid threshold in RANSAC, which leads to better performance when there is no clear-cut difference between inliers and outliers. Nevertheless, MSAC is computationally more demanding than RANSAC. Maximum Likelihood Estimation Sample Consensus (MLESAC) [83] estimates the model parameters that maximize the likelihood, assuming that the noise is Gaussian and that the outliers are uniformly distributed. In [43], the authors developed a Prior-MLESAC algorithm to extract both vertical and non-vertical planar and cylindrical structures. In [86], the authors applied MLESAC to fit surface primitives in PCD. MLESAC provides a more statistically rigorous estimation than RANSAC but requires an accurate estimation of the inlier ratio, which is not always available. Also, the assumption of uniformly distributed outliers may not hold in all cases. In [87], the authors proposed an outlier detection in PCD by getting maximum consistency with minimum distance (MCMD). The method is faster and more efficient in detecting planes by estimating consistent normal vectors. Another RANSAC variant named Progressive Sample Consensus (PROSAC) was proposed to exploit the linear ordering defined on the set of correspondences by a similarity function used in establishing tentative correspondences [42]. It capitalizes on the concept of assigning a likelihood to data points being inliers and arranges them in this order. This can improve the speed of the process and the robustness of the result by gradually decreasing the sample size, especially for datasets with a large number of noise or outliers. In [43], the experiments showed that PROSAC is more robust to the data with more outliers than RANSAC and MLESAC while processing time is less than Prior-MLESAC.

Clustering in PCD refers to the grouping of spatially or geometrically similar points into distinct subsets, aiding in data segmentation and feature extraction. K-means [88] is a partitioning method that clusters the points into K number of centroids. It is simple and efficient but requires the number of clusters to be pre-specified. Mean Shift [89] is a non-parametric, iterative clustering algorithm used primarily for mode seeking and data clustering. It can find clusters of any shape, but all points will gravitate toward cluster centres even for outliers. Agglomerative Clustering [90] is a hierarchical clustering method that starts with each data point as an individual cluster and successively merges the closest clusters until only one cluster remains or a specified stopping criterion is met. However, this method does not explicitly handle outliers and can be computationally expensive for large datasets. On the other hand, Density-based Spatial Clustering of Application with Noise (DBSCAN) is an unsupervised 3D data clustering algorithm that groups together the points that are close to each other based on a density criterion [44]. The algorithm can identify clusters of arbitrary shapes and noise points that do not belong to any cluster. The algorithm works by defining two parameters: a threshold for the number of neighbours, *minPts*, and a radius, Eps, to measure an arbitrary distance. Given a set of data points, DBSCAN begins by randomly selecting a point and examining all other points within a distance of Eps from that point. If there are at least *minPts* points within that radius, a new cluster is formed. In [45], the authors used DBSCAN for boundary detection for PCD. One of the advantages of DBSCAN is that it does not require specifying the number of clusters in advance, unlike many other clustering algorithms such as k-means. Another advantage of DBSCAN is that it can handle clusters of

different shapes and sizes, and it is also robust to noise and outliers in the dataset. However, it struggles with separating clusters with varying densities. HDBSCAN [91] is an extension of DBSCAN that can find clusters of varying densities. Instead of working with a single radius, HDBSCAN constructs a hierarchy of clusters and then extracts flat clusters from this hierarchy, but it is more computationally intensive.

2.3 IFC and Object Descriptor

Industrial Foundation Classes (IFC) is a schema and an open standard used in the AEC industry for representing and exchanging building and construction data among various software applications. It defines a standardized data structure for building information, including information about the building's geometry, spatial relationships, and properties of building elements, such as walls, floors, doors, and windows. Many applications including quantity take-off [46], model code compliance checking [47], and energy simulations [48] can be done by IFC models. An IFC model follows a top-down hierarchy to express the properties of a building's structural, mechanical, and electrical objects. Generally, it contains an object's ID label (*GUID*), dimension (*IfcBoundingBox*), location (*IfcObjectPlacement*), material properties (*IfcMaterial*), connection relationships (*IfcRelConnectsPathElements*), and space relationships (*IfcRelSpaceBoundary*), which can be employed for matching DI structural object instances in PCD in our proposed solution.

The concept of object descriptor is widely used in computer vision-based fields. It refers to a set of attributes or characteristics that can be used to identify or classify an object. In [93], the authors evaluated five SOTA 3D descriptors for object recognition, including local descriptors for instance recognition, and global descriptors for classification. Specifically, spin image [94], Signature of Histograms of Orientations (SHOT) [95], and Unique Shape Context (USC) [49] use histograms and point normal vectors to identify local features. Ensemble of Shape Functions (ESF) [13] uses distributions of distances, areas, and angles to identify objects. Principal Axes Descriptor [14] applies principal component analysis and occupancy ratios to identify object types. Similarly, in [54] and [15], the authors used eigenvectors and eigenvalues to generate descriptors for cluster matching. In the Scan-vs-BIM context, existing descriptors can be material-based including colour, texture, and reflectivity [29], or shape-based including height, width, radius, and curvature [50, 92], or some combination thereof. Some descriptors use heuristic models with human codification to identify or classify objects [16]. However, the specific attributes of an object descriptor restrict this approach to only being capable of DI-compliant cases for instance detection and segmentation in PCD. Relying solely on the descriptor is insufficient to address challenges in PCD with significant occlusions and clutter. Also, there is still a lack of leveraging IFC to develop a generic descriptor for instance matching in the Scan-vs-BIM context.

2.4 Gaps in Knowledge and Objectives

Structural object detection and segmentation in PCD in the Scan-vs-BIM context presents several gaps in knowledge:

1. At present, there is still a lack of generic methods to accurately segment the entire cluster of valuable points while simultaneously eliminating noisy points in highly cluttered and obstructed environments in the Scan-vs-BIM context. In certain situations, only a portion of an object's surface is visible and can be captured in the PCD, and obstructions such as building materials or workers can further complicate the scanning result. Because of these difficulties, PCD captured in such complex environments may not accurately reflect the status of instances and can make it

challenging to achieve high-precision instance segmentation and geometry reconstruction.

2. We do not yet know how to segment the complete point cluster with a high-precision result when there are significant deviations in the position, orientation, and scale of the DI geometry and the actual object instances. Existing methods are unable to ensure the extraction of all relevant points corresponding to the object instance in the Scan-vs-BIM context.

3. There is still a lack of generic methods to precisely segment the entire point cluster in non-Manhattan-world buildings in the Scan-vs-BIM context. Most methods focus on Manhattan-world buildings where objects are aligned with X-Y-Z axes. To the best of our knowledge, few methods can deal with matching as-designed diagonally positioned or curved walls with high-precision performance in PCD.

4. We do not yet know how to precisely detect and segment the complete point cluster for building object instances that have non-primitive shapes, such as cross piping joints, sprinklers, terminals, and light fixtures. The most existing methods are only effective for cuboid and cylindrical shapes in the Scan-vs-BIM context.

The objective of this research is to develop an automatic high-precision method to match (detect and segment) as-designed planar, curved, and linear structural object instances in PCD. The matching result of point clusters can be used for progress monitoring and quality control at the construction and operation stages. More specifically, this work addresses gaps 1, 2, 3, and part of gap 4 above for structural objects in buildings.

3. Proposed Solution

3.1 Scope and Overview

The scope of this research is limited to the most frequent planar, curved, and linear structural instances in a typical building. More specifically, we focus on planar walls, symmetrically curved walls, slabs including floors and ceilings, beams, and columns, since around 81.44% of the top frequent structural objects are from these classes [1].

The general thrust behind our proposed method is to follow a top-down idea to break the whole PCD as a high-level initial input into smaller, more manageable clusters in each step. We designed a recursively narrowing-down segmentation algorithm to process clusters from each step to finally reach a high-resolution result of instance-level segmentation. This method can reduce the information loss by clustering points with a necessary but not sufficient condition in each step.

The workflow of the proposed method is illustrated in Figure 1. We use acronyms to present inputs, intermediate outputs, and final outputs of each step. For example, {B|A} refers that B is a subset of A. The left part of the figure shows the inputs from the start and the outputs from the end along with the intermediate outputs of five process steps, while the right part of the figure elaborates on the process of each step within a grep dash box. We require an IFC model and the coarsely aligned PCD, P_0, as two kinds of data inputs from the beginning. More specifically, we determine the instance that we want to monitor and record this designated instance's GUID as an input for the further process in Step 1 (shown in magenta). On the other hand, the coarsely aligned PCD refers that the PCD scanned at the construction

stage is coarsely aligned with the IFC model. This is data pre-processing and will be discussed later in the section of research methodology. The flowchart in blue on the left side of the figure shows the processes of the coarsely aligned PCD with a recursively segmenting algorithm from Step 2 to Step 5, with the support of the outcome of Step 1. More precisely, an object instance descriptor of the designated instance generated from an IFC model is used to support cropping PCD within an enlarged bounding box in Step 2, segmenting the remaining PCD into a fitted shape in Step 3, and optimizing the final output cluster in Steps 4 and 5. The final output at the end is the point cluster corresponding to the designated wall, column, beam, or slab instances. Each step will be illustrated in detail in the following sections.

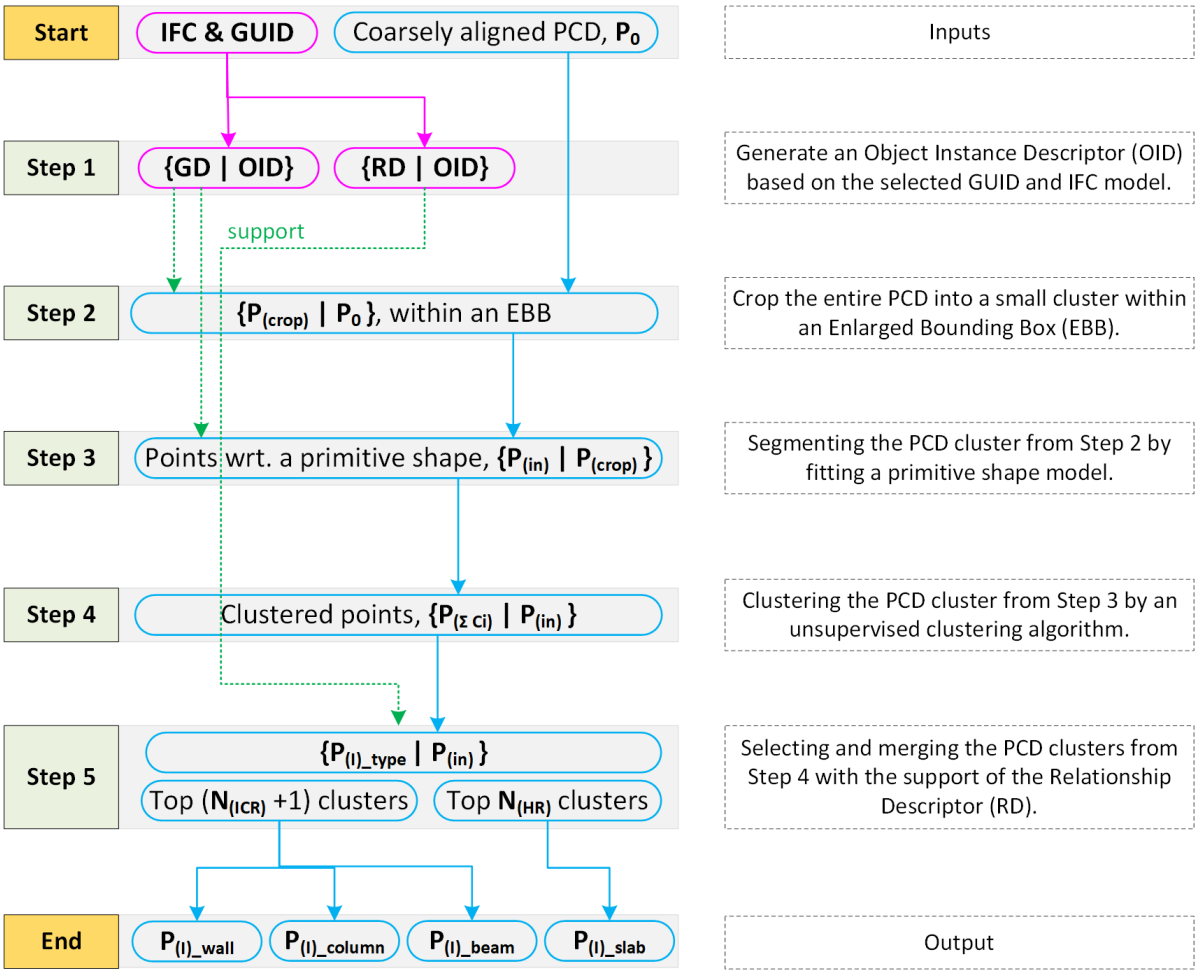


Figure 1 The workflow of the proposed method.

3.2 Generating Object Instance Descriptor (Step 1)

To the best of our knowledge, most of the existing object descriptors extracted features of object classes or instances, and they were only created for the specific subject matters of research. We therefore proposed a new descriptor named IFC-based Object Instance Descriptor (OID) that is more generic, standardised, and efficient to solve the matching problem in the Scan-vs-BIM context. We take advantage of IFC to calculate and encode a set of necessary variables which can support matching instances in the PCD since an input IFC model is a DI benchmark.

We define an OID here as a data structure that encapsulates the properties associated with a specific instance of a building. We proposed two sub-descriptors, namely Geometry Descriptor (GD) and Relationship Descriptor (RD) to compose a complete OID, as shown in equation (1). GD refers to the geometric attributes that reflect the instance itself while RD refers to the interaction attributes that reflect the surrounding information of the instance. We proposed these two sub-descriptors because we need the both of attributes' information to help us segment the PCD when there are deviations in terms of position, orientation, and scale between the DI and as-built instances.

$$OID = \begin{bmatrix} GD & 0 \\ 0 & RD \end{bmatrix} \quad (1)$$

Table 4 The structure of the proposed IFC-based Object Instance Descriptor (OID), including two sub-descriptors: Geometry Descriptor (GD) and Relationship Descriptor (RD).

IFC-based Object Instance Descriptor (OID)			
Sub-descriptor	**Acronym**	**Explanation**	**Description**
Geometry Descriptor (GD)	**O**	Orientation / Normal	(i, j, k)
	AABB	Axis Aligned Bounding Box	{Xmin, Xmax, Ymin, Ymax, Zmin, Zmax}
	S	Primitive Shape Type	cuboid, cylinder
Relationship Descriptor (RD)	**HR**	Hierarchy Relation	IfcBuilding → IfcSpace → IfcInstance
	ICR	Inner Connection Relation	∀AABBxy_connected ∉ (∀AABBxy+Δ)
	BCR	Border Connection Relation	∃AABBxy_connected ∈ (∀AABBxy+Δ)

The structure of the proposed OID for supporting the matching process of DI planar, curved, and linear structural instances in the PCD is illustrated in Table 4. In the sub-descriptor of GD, we encode 1) the maximum and minimum XYZ coordinates of the axis-aligned bounding box, $AABB$, as it can help to set the parameters of an Enlarged Bounding Box (EBB) to crop PCD in Step 2; 2) the attribute of orientation, O, as it serves as a constraint to help determine the points corresponding to the as-designed shape in Step 3; 3) the primitive shape type, S, as it also serves as a constraint to help shape model detection in Step 3. Specifically, we choose AABB rather than Oriented Bounding Box (OBB) because it is easier to compute the accurate value of $\{X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}\}$ for AABB without any other conditional inputs and is more generic to be applied on the instances with complex shapes. Figure 2 shows two examples of AABB of two diagonally positioned walls, separately. A diagonally positioned wall refers to the wall that is still perpendicular to the X-Y plane, but the principal axis is not aligned with neither the X nor the Y axis. The attribute of orientation, O, refers to two different variables in different cases, namely, it represents a normal vector, $\vec{n}$, when the object instance is a wall and slab, while it represents a principal axis, $\vec{a}$, when the instance is a beam or column. Finally, we only consider two types of primitive shapes, namely, cuboid and cylinder, as they can be used to represent the geometry of the most frequent structural object classes for this research.

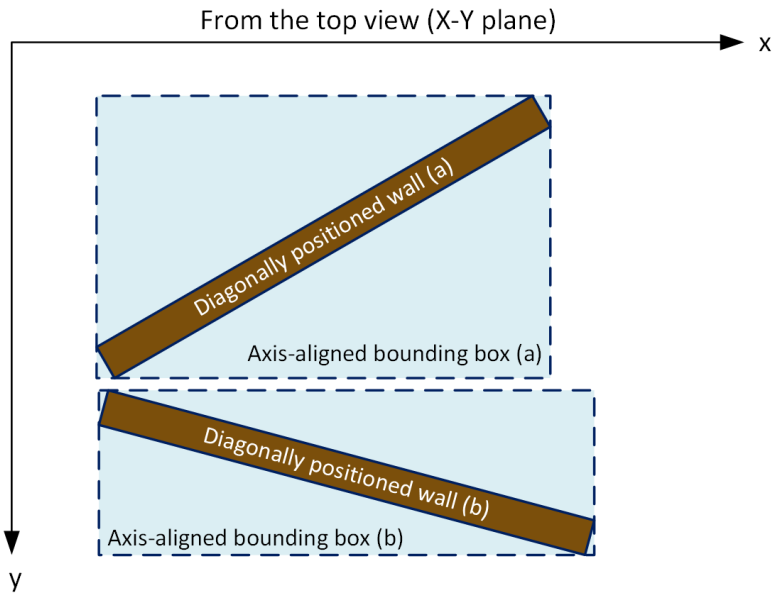


Figure 2 Two examples of axis-aligned bounding boxes of diagonally positioned walls.

In the sub-descriptor of RD, we define, compute, and encode three attributes, namely, Inner Connection Relation (ICR), Border Connection Relation (BCR), and Hierarchy Relation (HR). As discussed before, RD refers to the interaction attributes that reflect the surrounding information of the instance, which is useful for optimizing the final point clusters in real, complex Scan-vs-BIM contexts. Specifically, ICR and BCR work for the point cluster optimization of walls, beams, and columns, whereas HR works for slabs. Figure 3 indicates one example of ICR between the targeted wall 0 and the connected wall 2, as well as three examples of BCR between the targeted wall 0 and the connected wall 1 (in three cases). Precisely, we define ICR by first computing the maximum and minimum XY values of AABB of two related walls, and then discriminating ICR by the equation (2) shown below:

$$\forall AABB_{xy\_connected} \notin \left(\forall AABB_{xy+\Delta}\right) \quad (2)$$

where $AABB_{xy\_connected}$ refers to the XY value of AABB of the connected wall while $AABB_{xy}$ refers to the XY value of AABB of the targeted wall. We add a small tolerance, Δ, to improve the result's robustness. Figure 3 indicates that the connected wall 2 has an ICR with the targeted wall 0 since none of the vertices of wall 2's AABB belongs to the threshold of vertices of wall 0's AABB. Similarly, the discrimination of BCR follows the equation (3) below:

$$\exists AABB_{xy\_connected} \in \left(\forall AABB_{xy+\Delta}\right) \quad (3)$$

which means that the connected wall 1 has a BCR with wall 0 since at least one vertex of wall 1's AABB is within the threshold of vertices of wall 0's AABB in Figure 3. We only need to compute XY values from the top view to determine the connection relationship without considering Z values since the scope of this research is limited to the walls that are parallel to Z axis. We can also determine the connection relationship between beams and columns by equations 2 and 3. On the other hand, we define HR as the affiliation relations between spaces and instances and use this relationship to help optimize the point clusters and remove noisy points in PCD for slabs. We use a backward reasoning method to determine the

number of spaces from the related building elements including doors or walls. For example, in Figure 4, *IfcRelSpaceBoundary* can be determined by the inverse attribute of *IfcElement*. The *IfcSpace* can then be determined by the attribute "Relating Space" of *IfcRelSpaceBoundary*. We will explain in detail about how to use RD to optimize the result in Step 5.

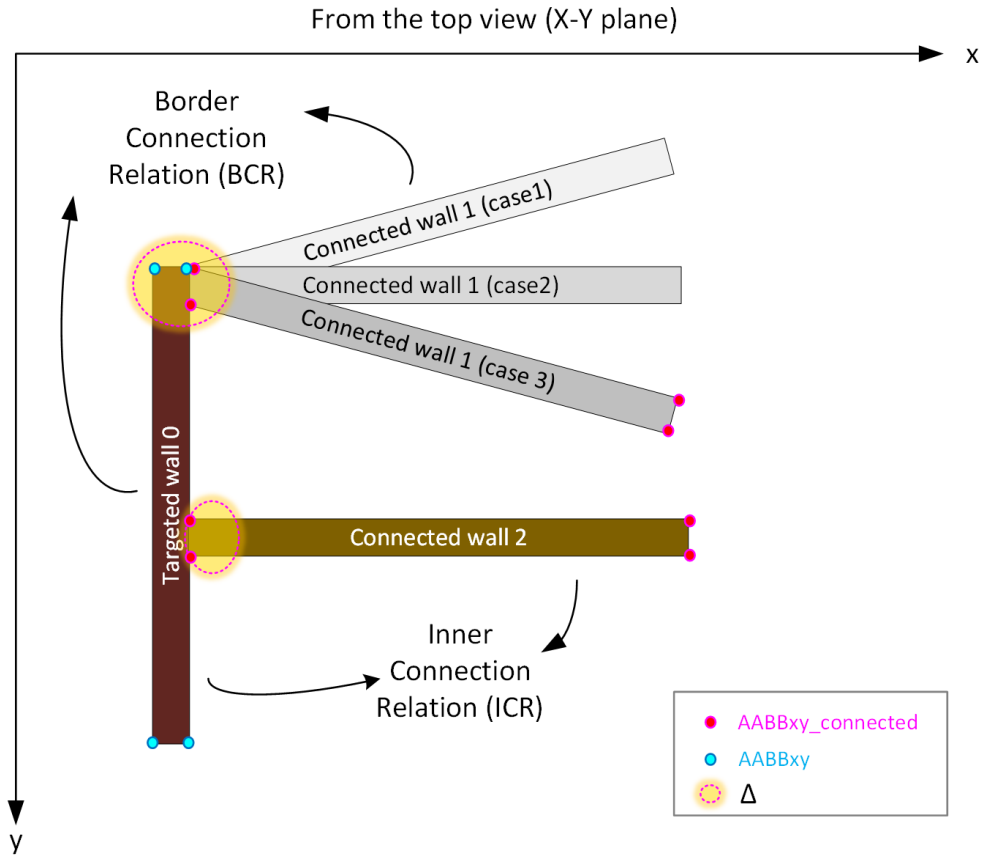


Figure 3 Indication of Inner Connection Relation (ICR) and Border Connection Relation (BCR) in a Manhattan-Word case (wall 0, wall 1-case 2, and wall 2), and non-Manhattan-World cases (wall 1-case 1 and wall 1-case 3).

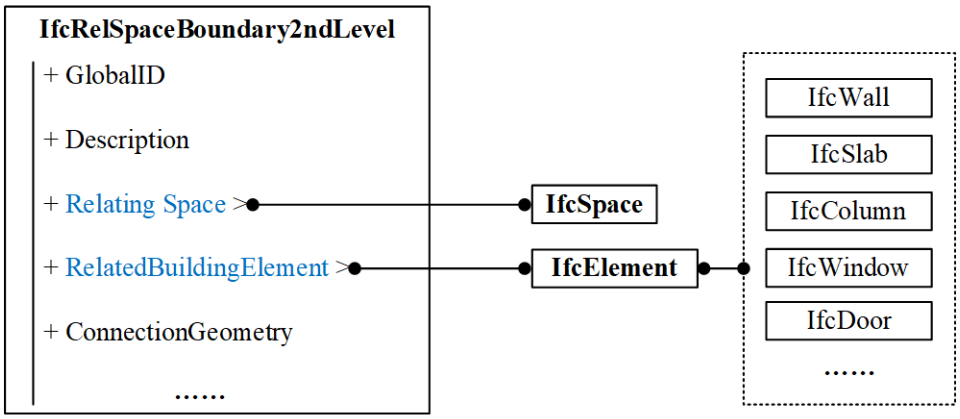


Figure 4 Indication of Hierarchy Relation (HR).

In summary, our proposed OID has three benefits to support matching structural instances in the Scan-vs-BIM context: 1) the attribute AABB in GD can support generating and modifying the EBB which is used for cropping the whole PCD into a small-scale, targeted cluster for subsequent object instance matching; 2) the attributes orientation and primitive shape type in GD can support detecting and extracting the points corresponding to the related shapes to narrow down the point clusters; 3) the RD can help to select the top clusters to optimize the final segmentation result; 4) the GD and RD can also support estimating artificial points to fill gaps in the point cluster (this benefit is out of this research scope).

3.3 Narrowing Down PCD with Shape Detection (Step 2&3)

As our proposed solution follows a recursively segmenting logic, we aim to narrow down the input PCD, P_0, into smaller clusters step by step until find the optimized result. We first use an EBB to crop the entire PCD into a smaller cluster in Step 2 to narrow down the size of the PCD. This idea was inspired by the real cases in the Scan-vs-BIM context, where there are deviations in terms of position, orientation, and scale between DI and as-built instances. Using an EBB instead of an AABB to crop the input PCD can allow these deviations to exist, and at the same time, reduce the processing time caused by the input data size. Figure 5 shows an example of an AABB and an EBB with a cuboid instance. We can take the value of the AABB from GD in Step 1 and extend the size of the AABB by around 20% to 50% to generate an EBB. Cropping an instance within an EBB allows an appropriate tolerance to let us determine the instance in the point cluster, but at the same time increase the existing probability of noisy points.

We would like to segment the points, P_{crop}, in the EBB with the support of the IFC model to further narrow down the size of PCD. Given the fact that the shape of an as-built structural instance is the same as the DI model, we can segment the cluster from Step 2 by fitting a primitive shape model. In this step, we aim to extract the representing points in a simple, fast, and easily implemented manner, therefore, we chose PROSAC here to fit the points with a given shape model because this algorithm can improve the speed of the process as well as keep the robustness of the result, especially for the dataset with a large number of noise or outliers. PROSAC is an enhanced variant of the RANSAC that incorporates prior knowledge in the form of point ranking for robust parameter estimation in the presence of many outliers. The fundamental principle behind PROSAC is that, given a set of data points where the quality ranking is known, it is statistically more probable for the better-ranked data points to be inliers than the ones ranked lower. Therefore, PROSAC begins by sampling only the top-ranked data points during its initial iterations, incrementally expanding the sampling base as the iterations progress. For shape fitting, the normal consistency between neighbouring points is considered a good metric. Points with a consistent normal to their neighbours are ranked higher to consider for initial iterations of the shape estimation, leading to a more efficient result.

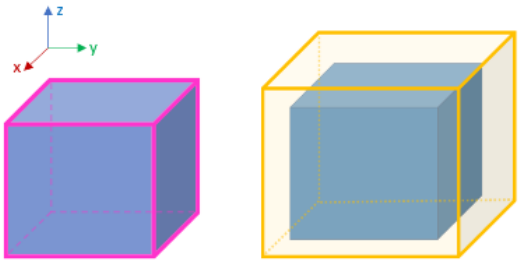


Figure 5 An example of an Axis-Aligned Bounding Box (AABB) in magenta on the left and an Enlarged Bounding Box (EBB) in yellow on the right, for an object instance of a cuboid.

We investigate two primitive shapes of the top common structural objects for segmenting PCD based on PROSAC: plane and cylinder, in Step 3, since cuboid and cylinder are two kinds of common geometry in the structural category. We use the model "plane" for cuboid detection (namely, planar walls, slabs, cuboid columns, and beams), and the model "cylinder" for curved edge or cylinder detection (namely, curved walls, cylindrical columns, and beams). Table 5 illustrates the type and the number of SAC models used in shape detection. The number of SAC models depends on the object types. Specifically, two planar surfaces are visible and captured in the scanned PCD for planar walls and slabs, so we need to detect two planes for these types of instances. Similarly, we need to detect four planes in the PCD cluster for cuboid columns and beams. Normal vectors of surfaces are estimated from GD in Step 1 to increase the robustness of the PROSAC algorithm. We only need to detect one cylinder for a cylindrical column or a beam with the radius value recorded in GD. Curved walls are more complex because we cannot use "plane" to detect the shape as planar walls. Since the curved wall is symmetrical, we can detect the curved two surfaces by detecting two cylinders. However, unlike cylindrical columns or beams that IFC has already recorded their radius attributes, few BIM models record the curvature of curved walls. For example, we usually do not know the radius and the centre of the curved wall in an IFC model. Therefore, we need to compute the radius of the curved surface for curved wall's shape detection.

Table 5 SAC Models used for Geometry Detection in PCD

Instance class	**SAC model**	**Quantity**	**Estimated parameters**
planar wall/slab	plane	2	normal vectors (i, j, k)
curved wall*	cylinder	2	radius margin (r_{min}, r_{max})
cylindrical column/beam	cylinder	1	radius margin (r_{min}, r_{max})
cuboid column/beam	plane	4	normal vectors (i, j, k)

**Curved walls are more complex. Details are explained in the body paragraph.*

The basic idea for estimating the curvature of a curved wall is to simulate a curved wall from the plan view as a circular arc of a circle and compute the radius and centre of this circle. Figure 6 (a) elaborates on how to simulate a curved wall in brown from the plan view into a circle and how to compute the centre and radius by three points on the circular arc with the help of the AABB. Specifically, we first compute the dimension of the curved wall's AABB from the IFC model, and then compute the tangent and crossing points between the curved wall and the AABB based on the coordinates of the maximum X, minimum X, maximum Y, or minimum Y. We only select three pairs of coordinates of tangent or crossing points from

552 the X-Y to compute the fitted circle's centre and radius. The intersection of the two
553 perpendicular bisectors of the lines with arbitrary two tangent or crossing points is the centre
554 of the circle. Assuming three tangent or crossing points are $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)$,
555 the centre $O(h, k)$ is:

556 $$h = \frac{(x_1^2 + y_1^2)(y_2 - y_3) + (x_2^2 + y_2^2)(y_3 - y_1) + (x_3^2 + y_3^2)(y_1 - y_2)}{2(x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2))} \quad (4)$$

557 $$k = \frac{(x_1^2 + y_1^2)(x_3 - x_2) + (x_2^2 + y_2^2)(x_1 - x_3) + (x_3^2 + y_3^2)(x_2 - x_1)}{2(x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2))} \quad (5)$$

558 The radius is:

559 $$r = \sqrt{(x_1 - h)^2 + (y_1 - k)^2} \quad (6)$$

560 In real-world cases, curved walls have various scales and orientations, which will lead to
561 different cases of tangent and cross points in the AABB. We investigated all five cases from
562 the plan view (X-Y plane) based on the orientation and scale of the curved wall in an IFC
563 model to infer the radius value of the simulated circle. Figure 6 (b) – (f) illustrate five cases
564 of an IFC curved wall from the plan view. Specifically, Figure 6 (b) shows a curved wall with
565 the shortest length of the arc, we need to add an arbitrary point on the arc besides two
566 crossing points to compute the radius. The rest four cases show different orientations and
567 lengths of the arc, along with the determined tangent and crossing points for computing the
568 radius value.

569 In summary, Steps 2 and 3 aim to narrow down the whole PCD input into a small size point
570 cluster and coarsely segment it by shape detection. It should be highlighted that all shapes
571 defined here are infinite. Therefore, the PROSAC-based shape detection can only find the
572 points corresponding to the required geometry in the EBB but cannot distinguish which point
573 belongs to the selected instance and which point just belongs to the shape (e.g., noisy points)
574 (Figure 7). To solve this problem, we developed a DBSCAN-based cluster optimisation
575 algorithm in Steps 4 and 5 with the help of the RD to remove the noisy points and optimise
576 the final segmentation result.

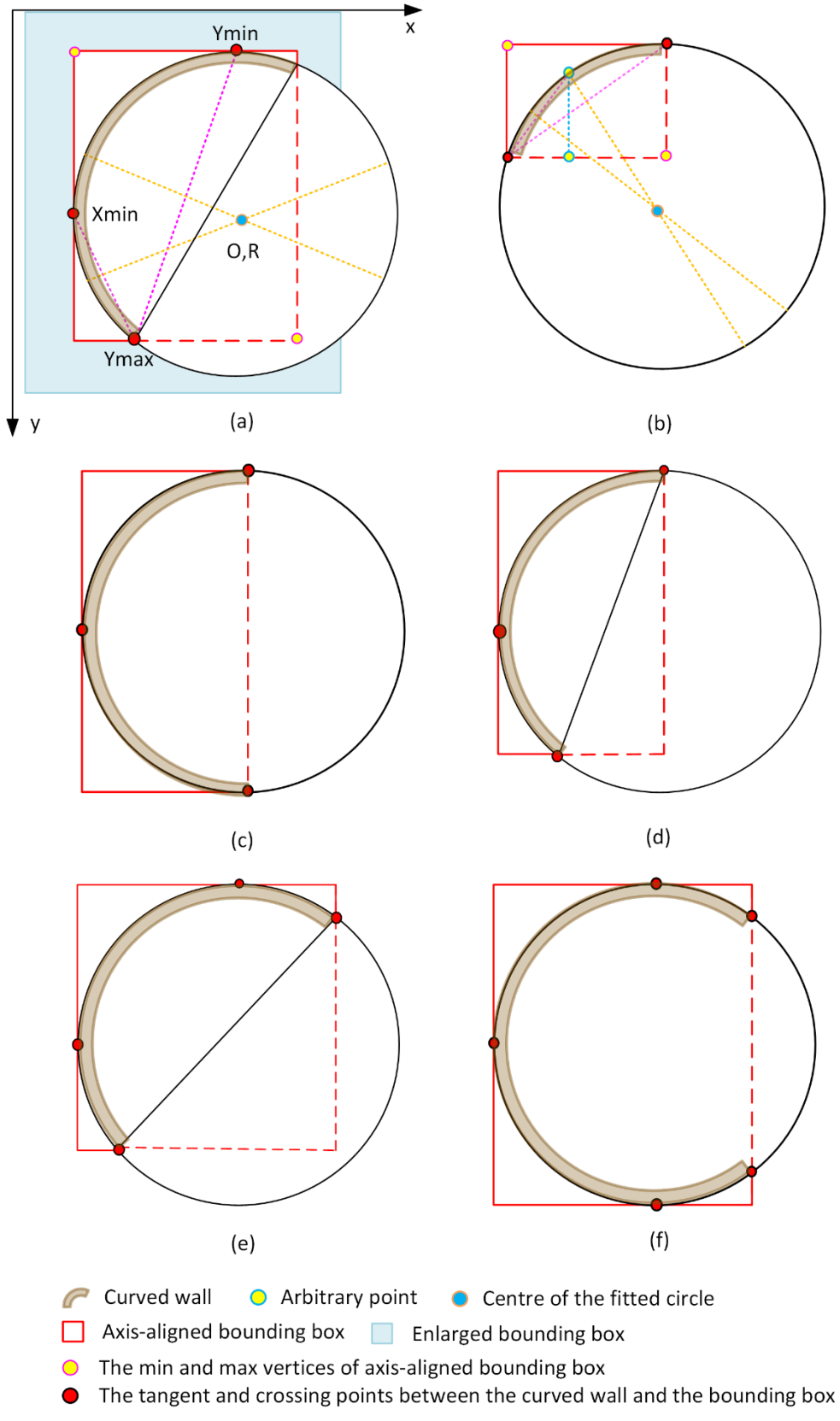
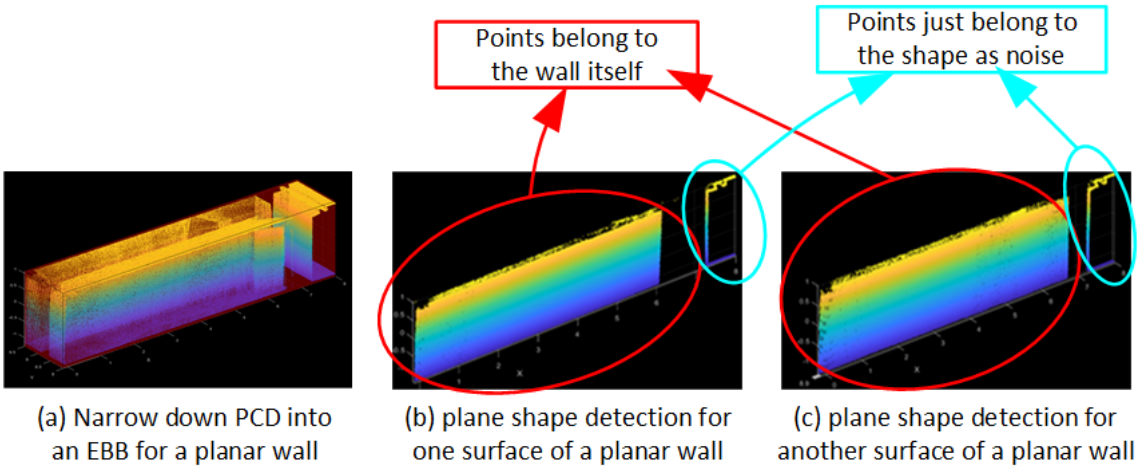


Figure 6 Curved walls in brown from the plan view (X-Y plane) in IFC models. (a): calculate the centre and the radius of the fitted circle. (b) - (f): five cases of a curved wall in an axis-aligned bounding box.



582

583 Figure 7 (a) narrowing PCD into an EBB for a planar wall; (b) & (c) the points of two surfaces
584 extracted by plane shape detection, the points in the red circle belong to the wall itself, while the
585 points in the blue circle just belong to the shape, considered as noisy points.

586

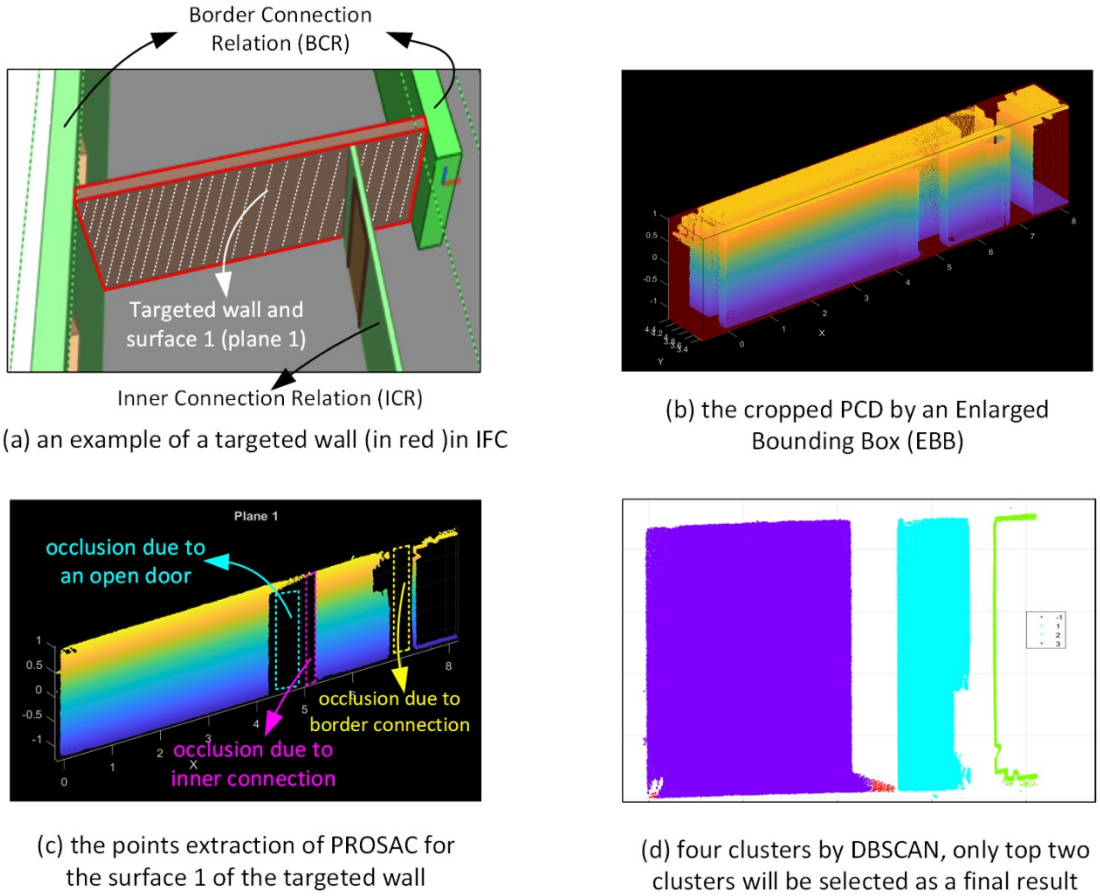
587 ## 3.4 Optimization with Unsupervised Clustering (Step 4&5)

588 PROSAC can only extract the points corresponding to the defined shape but not the points
589 corresponding to the designated instance, as illustrated in Figure 7. We need to segment the
590 point cluster extracted from Step 3 and remove the noisy points to optimise the result.
591 Furthermore, an instance can be represented by different numbers of point clusters due to
592 various occlusions and clutter, we therefore apply DBSCAN to cluster points into different
593 patches since DBSCAN is an unsupervised clustering algorithm which can identify clusters
594 of arbitrary shapes and noise points without setting the number of clusters. The key step to
595 obtain the desired clusters using DBSCAN is the setting of two parameters: the radius
596 threshold, *Eps*, and the minimum number of neighbours in a cluster, *minPts.* Specifically,
597 the points classified into the same cluster should reflect a part of the specific instance, so the
598 radius threshold should be larger than the density of PCD. Meanwhile, for each object, the
599 reason for the existence of different clusters is due to the inner-connected instances (e.g.,
600 walls) or spaces that segment the PCD. Therefore, determining the thickness of the connected
601 instances can help to set the radius threshold. In practice, due to scanning errors, density
602 settings from the scanner, and scale discrepancies between the DI and as-built instances, it is
603 optimal to adjust the setting of *Eps* based on the thickness values to achieve the best
604 performance. For the minimum number of neighbours in a cluster, *minPts*, compared to the
605 PCD size of each instance, it can be set with a small number (e.g., 60 – 100) as a default to
606 facilitate clustering and noise elimination. The point cluster can be split into different small
607 clusters after applying DBSCAN in Step 4, where we need to further determine which
608 clusters belong to the instance itself.

609 One instance may be represented by several different clusters in PCD because of occlusions
610 and clutter from the connected instances. Therefore, we take advantage of RD generated in
611 Step 1 to help rank, select, and merge the top number of clusters in Step 5 for result
612 optimization. Kd-tree is used here to accelerate the computation speed. More specifically, we
613 use ICR and BCR to rank the clusters for planar and curved walls; HR to rank the clusters for
614 planar slabs including floors and ceilings. Since columns and beams are always built
615 independently in a space such as a lobby or a room, we can directly segment the points by
616 fitting models.

Planar and Curved Walls. Figure 8 uses an example of a planar wall to elaborate on how ICR and BCR help to rank and select clusters corresponding to the instance as the final extraction result. In Figure 8 (a), the red-edged wall in the IFC model is the targeted planar wall, where the surfaces with white-dash lines are the targeted plane. The targeted wall has two BCR and one ICR with three adjacent walls. In Figure 8 (b), we crop the PCD by an EBB. Figure 8 (c) shows the result of applying PROSAC with the shape of "plane" for surface 1. It is obvious that the extracted plane contains some occlusions and clusters due to an open door, an inner connected wall, and a border-connected wall. Then we apply DBSCAN to cluster points into different clusters (Figure 8 (d)). We then rank the clusters by the number of points. We only select top $(N_{(ICR)} + 1)$ clusters as the final extracted point cluster, where N_{ICR} refers to the number of the ICR. We apply the same method for curved walls. The optimized result for a wall is shown below:

$$P_{(I)\_wall} = \sum_{1}^{(N_{ICR}+1)} (P_{cluster} \in P_{(in)}) \quad (7)$$



(a) an example of a targeted wall (in red)in IFC

(b) the cropped PCD by an Enlarged Bounding Box (EBB)

(c) the points extraction of PROSAC for the surface 1 of the targeted wall

(d) four clusters by DBSCAN, only top two clusters will be selected as a final result

Figure 8 An example of how the ICR supports DBSCAN to rank clusters for the final points extraction. (The IFC model and PCD are from an open source [51])

Slabs. Space enumeration over a slab is a key point in ranking and selecting clusters after DBSCAN. We take advantage of HR to inversely infer the number of spaces on a slab. Figure

9 illustrates two examples of using walls (left) and doors (right) to infer *IfcSpace* in an IFC model. We only select top ($N_{(HR)}$) clusters as the final extracted point cluster, where $N_{(HR)}$ refers to the number of spaces in the HR. The optimized result for a slab is shown below:

$$P_{(I)\_slab} = \sum_{1}^{(N_{HR})}(P_{cluster} \in P_{(in)}) \quad (8)$$

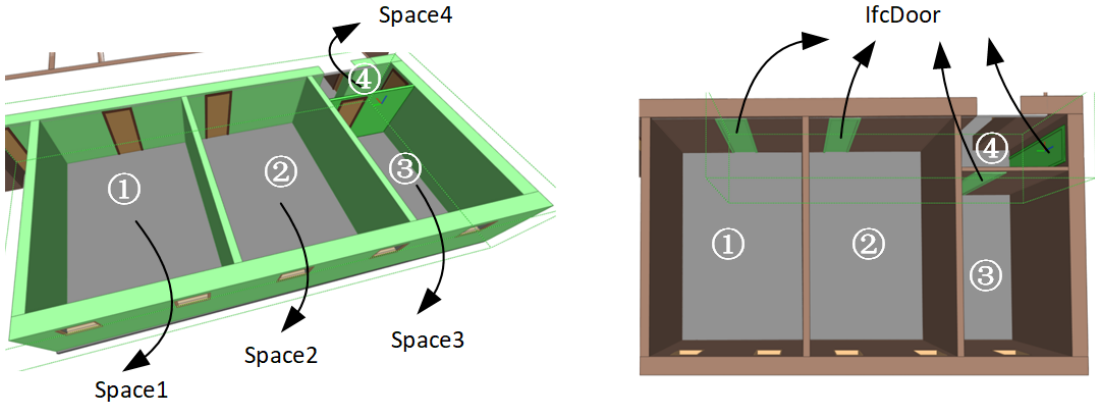


Figure 9 An example of the HR between spaces and walls (left) / doors (right) in an IFC model. (The model is from an open source [51]).

Columns and Beams. Usually, we do not need to select clusters for columns and beams as they are always built without any ICR or HR. In a few cases, we can apply ICR to select top clusters for result optimization if columns or beams have intermediately connected with other instances. Despite all that, there are always some signs and marks on the columns and beams in real-world cases, which can cause false positive results after applying PROSAC in Step 3. For example, Figure 10 (a) shows a sign of fire exit on a column, of which the points can also be extracted from PROSAC. We then proposed a size restraint algorithm to remove these noisy points and optimize the result. Figure 10 (b) illustrates the proposed size restraint algorithm. We first compute two distances (m and n) from two parallel planes from the plan view, separately. Then we compute d_{max} as the longest distance from the central point, which serves as the threshold to select candidates and remove noisy points. The proposed size restraint algorithm is robust for the real-world dataset since it does not rely on the instance scale of the as-designed model. It should be noted that this algorithm is effective when none of the four surfaces of the column is completely occluded by clutter. In other words, all four surfaces of the column need to be at least partially exposed to help determine the distances between two pairs of parallel faces from a top-down perspective. On the other hand, we do not need to apply this size restraint on cylindrical columns and beams but only need to use the ICR to optimize the result if applicable. The optimized result for a column or beam is shown below:

$$P_{(I)\_column/beam} = \sum_{1}^{(N_{ICR}+1)}((P_{cluster}|size\ restraint) \in P_{(in)}) \quad (9)$$

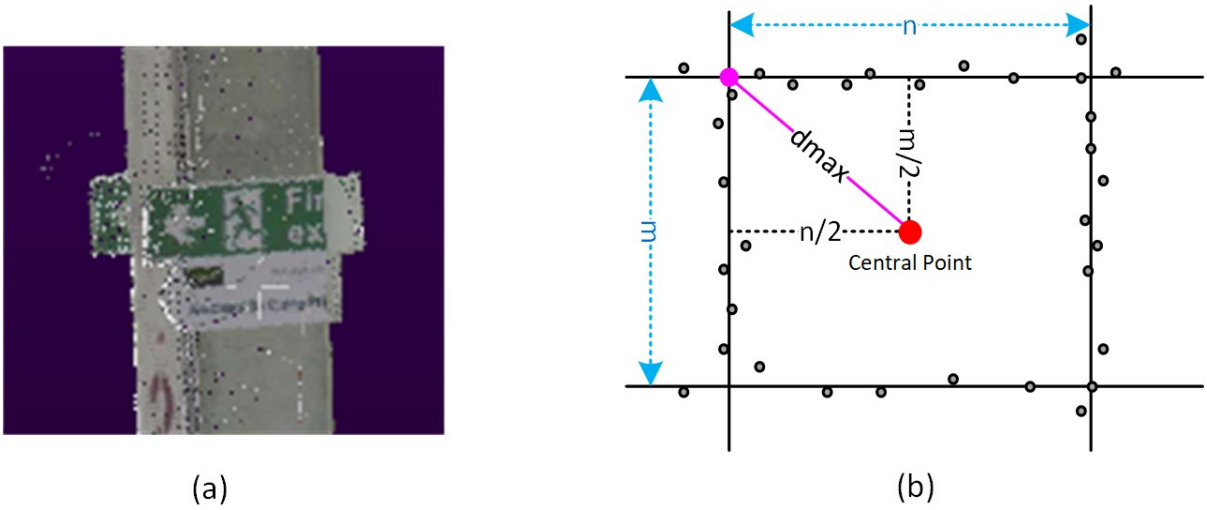


Figure 10 (a) An example of signs and marks on the column in a real point cloud cluster; (b) the plan view of four surfaces (solid black lines) fitting the point cluster (grey dots) of the as-built column, d_{max} is computed by surface distances as size restraint.

3.5 Pseudo Code and Summary

Overall, the pseudo-code of matching design-intent planar, curved, and linear structural objects in PCD is proposed in Algorithm 1. At the beginning, for each selected instance in the IFC model, M, we use its GUID, I, to compute the object instance descriptor, OID. Specifically, if the GUID belongs to the object classes of walls, columns, and beams, we then compute the values of attributes for GD, ICR, and BCR; otherwise, if the instance belongs to the slabs, we compute the attributes' values for GD and HR. Afterwards, an EBB is computed from the attribute's value of the AABB to crop and reduce the size of the entire PCD. Then, the PROSAC shape fitting is conducted by determining the inlier points, $P_{(in)}$, with the parameter hypothesis, $P_{(h)}$, of the designated shape, S, and the deviation value, ε. After that, DBSCAN is applied to divide the point cloud, $P_{(in)}$, into different clusters, C_m, by finding the neighbours of core points. Each cluster, C_i, is ranked by the number of points in descending order. Finally, the top $(N_{(ICR)}+1)$ of clusters are selected and merged as the final segmentation result, $P_{(I)}$, for walls, columns, and beams; the top $(N_{(HR)})$ of clusters are selected and merged as the final segmentation result for slabs. Our proposed method has some advantages compared with the SOTA methods: 1) An EBB is generated to crop the whole PCD, which can reduce the size of the input data into a small-scale cluster to decrease the computational complexity. 2) The bounding box is enlarged to make the method robust when there are distinct deviations in terms of position, orientation, and scale between the DI model and the as-built PCD. 3) The proposed method is robust for analysing raw PCD directly without any pre-processing or denoising. 4) The proposed method is robust for highly occluded PCD with clutter. 5) Compared with deep learning-based methods, the proposed method avoids the problem of a lack of training datasets and can directly identify the instance ID to facilitate progress monitoring and quality control.

Algorithm 1 Pseudo code for the proposed method.

Input:
$P_0(p_1,p_2,...p_n)$: coarsely-aligned PCD;
M: an IFC model;
I: GUID of an object instance;
S: a shape to initially explain observed data points;
ε: the tolerance value from scanner;
Eps: the radius threshold between points;
$minPts$: the minimum number of neighbourhood points, default: 60;
Output:
$P(I)$: the point cluster corresponding to the instance I;

for $I \in M$ **do**
  compute object instance descriptor OID :
  if $I \in walls, columns, beams$ **then**
    compute GD, ICR, BCR
  else if $I \in slabs$ **then**
    compute GD, HR
  end if
  compute the enlarged bounding box, EBB, from $AABB$
end for
for P_0 **do**
  crop P_0 into the EBB, as $P(crop)$
  for $P(crop)$ **do**
    compute the hypothesis $P(h)$, model parameters of S
    determine the inlier points, $P(in)$, by ε
  end for
end for
for each $p_i \in P(in)$ **do**
  compute neighbours within Eps: $N = get\ Neighbours\ (p_i,\ Eps)$
  if size of $N < minPts$ **then**
    mark p_i as noise
  else
    compute $ExpandCluster\ (pi,\ N,\ Cj,\ Eps,\ minPts)$
    compute $N' = getNeighbours\ (P',\ Eps)$
    $N = N \cup N'$, if size of $N' \geq minPts$, add all points to the new cluster C_j
  end if
end for
store all clusters, $C_m = \sum C_i$
for C_m **do**
  rank $C_i \in C_m$, based on the size of the points
  select top $(N_{(ICR)} + 1)$ clusters for walls, columns, beams
  otherwise, select top $(N_{(HR)})$ clusters for slabs
  save $P(I)$: = top clusters as final output
end for
End

4. Research Methodology

4.1 Data Acquisition and Pre-processing

We acquired five datasets shown in Table 6 to validate our proposed method. Each dataset contains an as-designed IFC model and an as-built PCD of buildings from the real world. The IFC models reflect both Manhattan-World and non-Manhattan-World buildings. The characteristics of PCD from the selected five datasets are summarised in Table 7. It is noted that both mobile and terrestrial scanners were used to collect points. Five datasets contain different numbers of points and various complexities of clutter and occlusions, which is suitable for evaluating our proposed method.

ISPRS (International Society for Photogrammetry and Remote Sensing) WG IV/5 Benchmark on Indoor Modelling contains six pairs of public datasets with different complexities [51, 52]. We selected three datasets with different building styles, point scales, and clutter complexities from the ISPRS benchmark. First, the PCD of the TUB2 dataset was captured by Zeb-Revo in a two-floor building at Technische Universität Braunschweig in Germany. The indoor scene comprises a total of 24 rooms on two floors which are enclosed by walls and ceilings with different thicknesses and heights. It also contains 51 open and closed doors and 21 windows. The building is not furnished. Second, the PCD of the UVigo dataset represents one room and an entrance hall captured at the University of Vigo in Spain. The scene contains one curtain wall, 20 windows and 7 open and closed doors. The scene also contains several columns with a circular cross-section and multiple rectangular surfaces. Third, the PCD of the GM dataset was captured in the Grainger Museum. This is a non-Manhattan-World building with many curved and diagonally positioned walls. Two further complex datasets were also produced from the real world. The PCD of Cambridge CEB was captured in the Civil Engineering Building (CEB) at the University of Cambridge with a FARO Focus 3D X330 Terrestrial Laser Scanner. It contains three floors with many furnished spaces. The PCD of ConSLAM was captured on a construction site at Whiteley's building in London. It contains columns with a rectangular cross-section. Overall, the geometry deviations of object instances in terms of position, orientation, and scale exist between each IFC model and its corresponding PCD in five datasets. As can be seen in Table 7, the levels of clutter and occlusions in the PCD of the GM, CEB, and ConSLAM datasets are relatively high.

Table 6 Real World Datasets Used to Validate the Proposed Method.

Dataset	Reference	Manhattan Type	Sensor
ISPRS WG IV/5-TUB2	[51]	Yes	Zeb Revo
ISPRS WG IV/5-UVigo	[51]	Yes	UVigo Backpack
ISPRS WG IV/5-GM	[51,52]	No	Zeb Revo RT
Cambridge CEB	self-collection	No*	FARO Focus 3D X330
ConSLAM	[53]	Yes	Velodyne VLP-16

**Although CEB itself is a Manhattan-World building, the coordinate system is not aligned with CEB, making it processed as a non-Manhattan-World building.*

733 Table 7 PCD Characteristics for the Five Datasets Used in This Study.

Dataset(abbr.) *	Scanner Type	No. of Points	Clutter & Occlusions
TUB2	Mobile	21.6×10^6	Low
UVigo	Mobile	14.9×10^6	Moderate
GM	Mobile	28.9×10^6	High
CEB	Terrestrial	37.9×10^7	High
ConSLAM	Mobile	15.8×10^7	High

734 **Using abbreviations for the selected datasets.*

735

736 We need to register the as-built PCD with its corresponding as-designed IFC model before
737 matching DI instances in PCD. The problem can be described as follows:

738 $$p_i = \boldsymbol{R}m_i + \boldsymbol{t} + \boldsymbol{V}_i \qquad (10)$$

739 where p_i refers to the points in the PCD and m_i refers to the points in the model, i = 1,2, 3...,
740 n. **R** is a rotation matrix and **t** is the 3D translation vector. $\mathbf{V_i}$ is the noise vector representing
741 the discrepancy after the coarse alignment by **R** and **t**. $\mathbf{V_i}$ can be considered as a slight
742 movement for registration refinement to minimise the discrepancy between the transformed
743 DI and the PCD. In this research, coarse registration is applicable as deviations are allowed
744 between the as-designed BIM and the as-built PCD. We proposed two efficient methods for
745 IFC to PCD coarse registration, summarized in Table 8. The first utilizes Recap for
746 coordinate system adjustment while the second calculates the registration matrix in
747 CloudCompare. We applied Method 1 to datasets TUB2, UVigo, GM, and CEB, and Method
748 2 to ConSLAM. Figure 11 showcases the registration results. Both methods aim to adjust the
749 coordinate system of PCD, using the IFC as a reference. Method 1 is more convenient when
750 the IFC's coordinate origin and the axis direction are easy to find, while Method 2 is more
751 reliable when the IFC is too complex to locate the coordinate origin and the axis direction.
752 The results of the two methods show little difference towards the coarse registration in this
753 study.

754 Table 8 Proposed Solutions for IFC to PCD Coarse Registration

Method	Step	Action	Software	File
	1	determine coordinate origin & direction	Solibri	IFC
1	2	adjust coordinate origin & direction	Recap	PCD
2	1	convert IFC file to OBJ file	IfcConvert	IFC
	2	select 4 pairs of corresponding points	CloudCompare	both

755

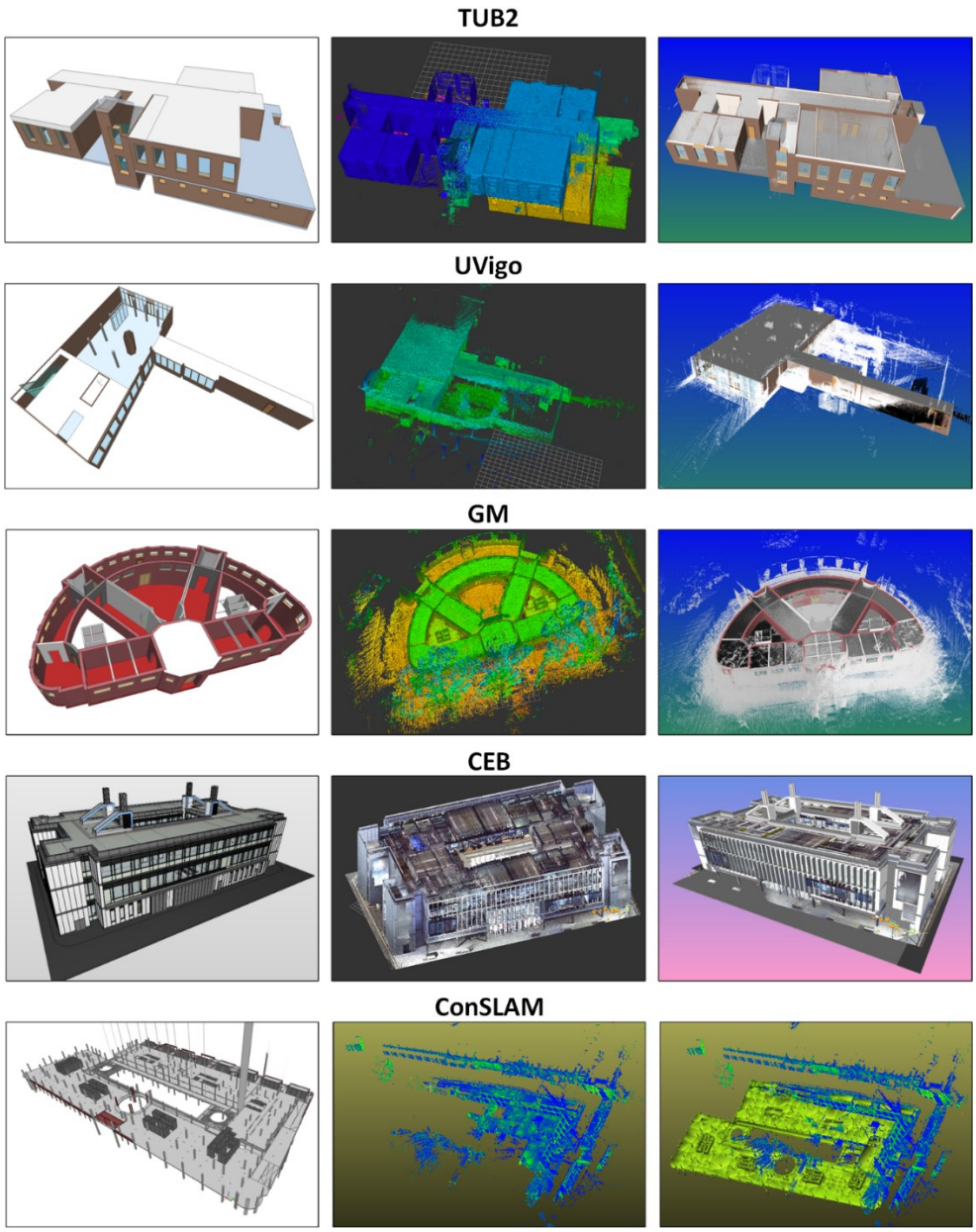


Figure 11 The as-designed IFC model (left), the corresponding as-built PCD (middle), and the coarse registration result (right) for the five datasets.

4.2 Experimental Results

We did experiments on the five datasets to evaluate the proposed method. For each class of the structural category, we selected several representative instances from five datasets. Specifically, we selected eight planar walls from TUB2, Uvigo, GM, and CEB datasets, where four of them were built in Manhattan-World type and four of them were in non-Manhattan-World type. Each planar wall has different complexities in terms of occlusions and clutter caused by doors, windows, furniture, and sundries. We also selected four curved walls and four slabs from GM, TUB2, UVigo, CEB, and ConSLAM datasets, including both Manhattan and non-Manhattan types. Besides the occlusions and clutter mentioned before, different curved walls have different constant curvatures, and the slabs have various shapes

771 from the plan view. Finally, we selected five columns including both cylindrical and cuboid
772 geometry from UVigo and ConSLAM datasets. In addition to the numerous occlusions and
773 clutter, the as-designed models are quite different from the as-built columns in terms of
774 dimensions and positions. In summary, we chose a wide range of planar, curved, and linear
775 instances with different complexities to validate the proposed method's efficiency and
776 reliability in automating the process of matching.

777 Having discussed the variety and complexity of the selected instances, we implemented the
778 proposed method in algorithms written by C++ and Python with Point Cloud Library (PCL)
779 [69] and IfcOpenShell. As Section 3.4 explained the process of setting DBSCAN's
780 parameters for clustering, we computed the minimum thickness of the connected walls in IFC
781 models for the selected instances and set a value slightly below this as the radius threshold.
782 Specifically, for planar walls and slabs, *Eps* was set around 0.1m for all datasets; for curved
783 walls, Eps was set around 0.6m in the GM dataset. The density of PCD in the GM dataset is
784 sparser and the connected walls are thicker than other datasets, so the *Eps* is larger than
785 others. All settings are larger than the PCD's density. It was not applied to columns since
786 there are no inner connections. The value of *minPts* was set as 60 to cluster the noisy points.
787 Figure 12 demonstrates the most representative experimental results. Specifically, the Figure
788 shows, from left to right, the as-designed model (IFC instance), the EBB, the result of the
789 extracted point cluster, and the ground truth. The left column of the figure illustrates the
790 selected IFC instance in red from DI, which is used to compute OID. The second column of
791 the figure illustrates the cropped PCD within the EBB. The following column shows the point
792 cluster of instance matching results after applying the proposed method. The right column
793 shows the ground truth of the instance point cluster which was generated from the PCD
794 manually. Specifically, TUB2-PW2 shows the segmented result when there are two inner
795 connections on one surface of the wall. GM-PW5 shows the result of a non-Manhattan planar
796 wall with significant clutter and occlusions. For GM-CW1, there are 4 thin planar walls in
797 grey connected to one surface of this curved wall, which are considered as the border
798 connection since they are very close to the boundary of CW1. For GM-CW4, it contains
799 significant clutter and one inner connection for one surface of the wall. TUB2-S1 shows the
800 segmented result of the slab connecting two floors. ConSLAM-RC2 shows the segmented
801 result when there are signs and clutter on the column. Finally, ConSLAM-RC4 shows a
802 successful segmentation result when there is a distinct scale deviation.

803 Throughout our experiments, we applied our proposed method to a total of 8 planar walls, 4
804 curved walls, 4 slabs, and 5 columns with different complexities in PCD. The overall
805 experimental results and visual representations are presented in Appendix Figures 13 - 15. In
806 conclusion, the experimental visualization presents a good result in real-world datasets for
807 matching structural objects in the scan-vs-BIM environment with significant occlusions,
808 clutter, deviations, and scales.

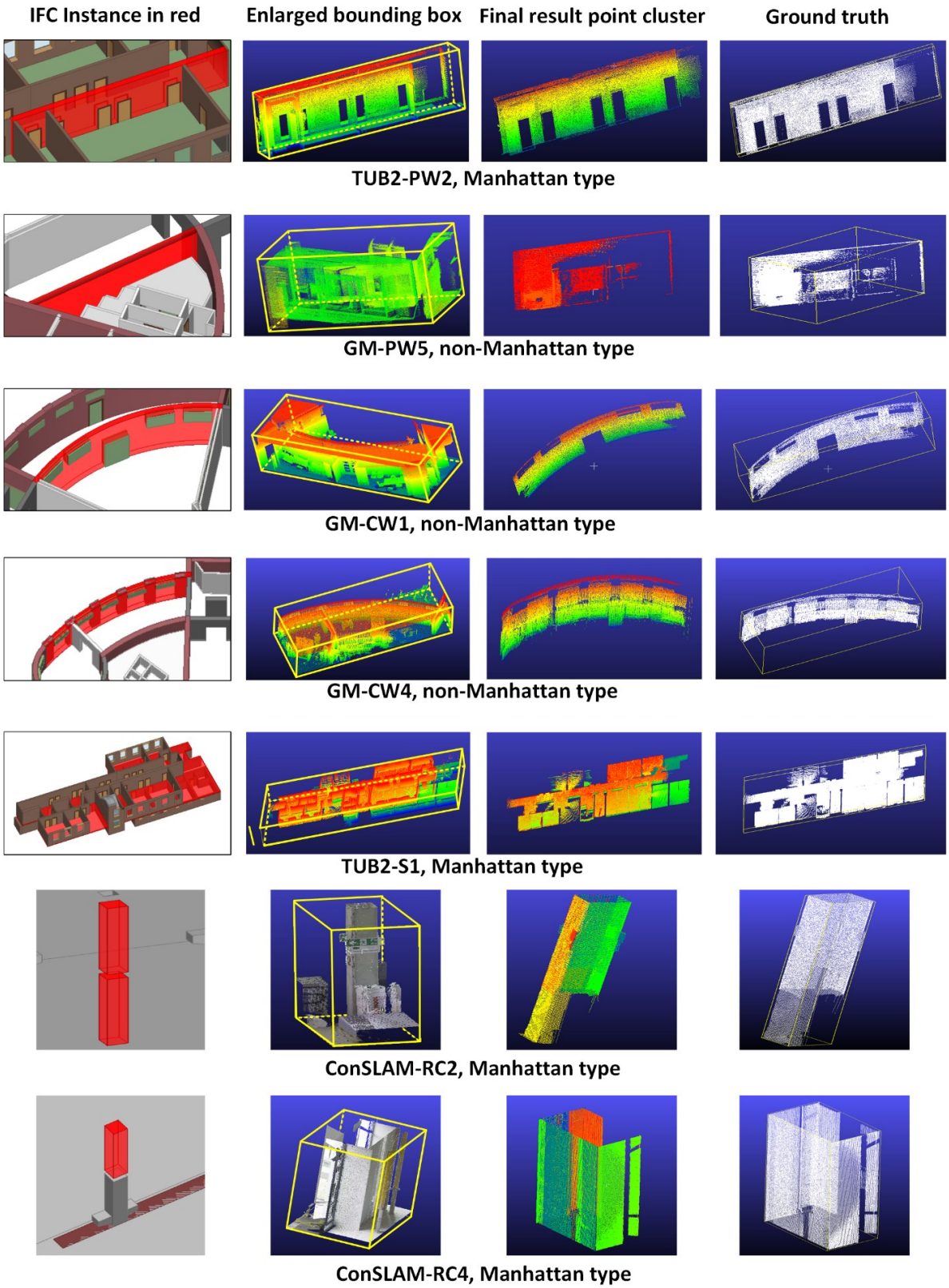


Figure 12 The representative selection of PCD matching results for planar walls (PW), curved walls (CW), Slabs (S), and cuboid columns (RC). From left to right: the selected IFC instance in red; the cropped PCD within the enlarged bounding box; the result of extracted point cluster using the proposed method; the manually generated ground truth.

5. Discussion

5.1 Evaluation Metrics and the Ground Truth

We applied a point-to-point comparison method to evaluate the experimental results against the ground truth (Please see the overall experimental results and visualization representations in Appendix Figures 13 – 15). We first assigned a unique label to each point besides the X, Y, Z coordinates before computing the result cluster and generating the ground truth cluster. Then, we matched each pair of corresponding points between these two sets of point clusters to compute Precision, Recall, and IoU. This evaluation method is more accurate than the surface-to-surface comparison since it computes the correspondence directly, rather than after transforming the points to surfaces. The metrics formulas are shown below:

$$Precision = TP/(TP + FP) \quad (11)$$

where TP refers to true positive and FP refers to false positive.

$$Recall = TP/(TP + FN) \quad (12)$$

where FN refers to false negative.

$$IoU = TP/(TP + FP + FN) \quad (13)$$

The evaluation between the result and the ground truth is shown in Table 9 and Table 10. The overall mean precision is over 0.962, the mean recall is over 0.934, and the mean IoU is over 0.914. Overall, our proposed solution is robust with high precision and recall on planar, curved, and linear structural instance segmentation in the Scan-vs-BIM context with various complexities. To the best of our knowledge, the proposed method is a novel workflow, and no similar method has been proposed before.

Table 9 Matching Result Evaluation by Instance Code: Proposed Method vs the Ground Truth (PW: planar wall, CW: curved wall, S: slab, CC: cylindrical column, RC: rectangular column)

Instance Code	Precision	Recall	IoU
PW1	0.999	0.964	0.964
PW2	0.991	0.928	0.921
PW3	0.989	0.932	0.923
PW4	0.978	0.967	0.947
PW5	0.948	0.931	0.890
PW6	0.964	0.979	0.944
PW7	0.993	0.953	0.947
PW8	0.971	0.958	0.931
CW1	0.945	0.907	0.862
CW2	0.920	0.890	0.823
CW3	0.952	0.880	0.843
CW4	0.947	0.894	0.851
S1	0.997	0.916	0.913
S2	0.999	0.979	0.978

S3	0.997	0.886	0.884
S4	0.986	0.955	0.942
CC1	0.987	0.905	0.895
RC1	1.000	0.985	0.984
RC2	0.989	0.998	0.988
RC3	0.995	0.943	0.939
RC4	0.992	0.942	0.935

Table 10 Matching Result Evaluation by Instance Type: Proposed Solution vs Ground Truth

Instance type	**mPrecision**	**mRecall**	**mIoU**
Planar wall	0.979	0.952	0.933
Curved wall	0.941	0.893	0.845
Slab	0.995	0.934	0.929
Column & Beam	0.993	0.955	0.948
All	0.962	0.934	0.914

5.2 SOTA Method Comparison

In this section, we aim to compare the performance of our proposed method with the SOTA methods to demonstrate its superior performance in real-world, complex environments. Previously, we introduced and summarised five SOTA method types in the Scan-vs-BIM context in Section 2.1 Table 1. To improve the SOTA methods' capability and robustness in different cases, we here merged Point-to-Point and Point-to-Surface methods as the improved SOTA method 1, and the last two methods (Feature-based and RANSAC-based methods) as the improved SOTA method 2 We do not consider Hough transform in evaluation since this method is only applied for cylinder detection, and sensitive to the noisy PCD with high occlusions and clutter.

Specifically, for SOTA method 1, we directly selected the points within a threshold from the surface after calculating the nearest distance between the as-built PCD and the centre of the as-designed model. For SOTA method 2, we used normal vector, shape, and length information together with RANSAC to select points from the as-built PCD. The improved SOTA methods can have a more stable performance in different scenarios from real-world datasets. Hence, the comparison results can be more convincing to prove the better performance, feasibility, and robustness of our new proposed method.

We conducted the comparison experiments from two perspectives: instance types and deviation scenarios, because we want to understand the SOTA method's performance towards 21 instances as we used for validating our proposed method, and the SOTA method's performance with different deviation scenarios in terms of position, orientation, and scale. Table 11 shows the matching evaluation results based on different instance types. Two SOTA methods perform fairly when detecting slabs. SOTA method 1 is better for column and beam segmentation in precision while SOTA method 2 performs better for planar wall and slab segmentation. SOTA method 2 has better overall performance in mIoU. Both two SOTA methods cannot deal with curved walls directly since the geometry information is not

868 complete and cannot be extracted directly from the original IFC model. On the other hand,
869 the SOTA method's performances in different deviation scenarios in terms of scale,
870 orientation, and position also need to be evaluated. These scenarios often happen in the real-
871 world context and our proposed solution can perform well towards these situations. Table 12
872 shows the evaluation results for both SOTA methods against the ground truth in four
873 scenarios: 1) the IFC instance model is directly aligned with PCD; 2) the IFC instance model
874 is reduced to around 25% smaller scale compared with the as-built instance; 3) the IFC
875 instance model is rotated approximately 20° from the as-built instance; 4) the IFC instance
876 model is moved around 20% from the original position. Finally, Table 13 compares the mean
877 precision, mean recall, and mean IoU between the proposed method and improved SOTA
878 methods. It is evident that our proposed solution performs much better than SOTA methods
879 in all cases.

880

881 Table 11 Matching Result Evaluation by Instance Type: Improved SOTA Method 1, Improved SOTA
882 Method 2 vs Ground Truth

Instance type	SOTA Method 1			SOTA Method 2		
	mPrecision	**mRecall**	**mIoU**	**mPrecision**	**mRecall**	**mIoU**
Planar wall	0.641	0.579	0.498	0.647	0.452	0.410
Curved wall	/	/	/	/	/	/
Slab	0.873	0.606	0.537	0.983	0.664	0.658
Column & Beam	0.834	0.369	0.363	0.791	0.525	0.502
All	0.783	0.518	0.466	0.807	0.547	0.532

883

884 Table 12 Matching Result Evaluation by Scenarios: Improved SOTA Method 1, Improved SOTA
885 Method 2 vs Ground Truth

Scenarios	SOTA Method 1			SOTA Method 2		
	mPrecision	**mRecall**	**mIoU**	**mPrecision**	**mRecall**	**mIoU**
IFC aligned with PCD	0.852	0.777	0.689	0.806	0.696	0.662
IFC scaled ~25%	0.899	0.576	0.530	0.827	0.481	0.460
IFC oriented ~20°	0.551	0.296	0.269	0.722	0.488	0.473
IFC moved ~25%	0.829	0.421	0.376	0.873	0.523	0.498
All	0.783	0.518	0.466	0.807	0.547	0.523

886

887

888 Table 13 Matching Result Evaluation: Improved SOTA Methods vs Our Proposed Method

Methods	**mPrecision**	**mRecall**	**mIoU**
Improved SOTA method 1	0.783	0.518	0.466
Improved SOTA method 2	0.807	0.547	0.523
Our proposed method	0.962	0.934	0.914

5.3 Strengths and Limitations

Our proposed method offers several advantages over existing SOTA approaches. First, we apply an EBB to crop the whole PCD, which can reduce the data size and computational complexity, as well as enhancing the method's robustness against position, orientation, and scale deviations between the DI model and as-built PCD. Second, the method does not require any pre-processing, such as denoising, for the raw PCD; it is adept at handling PCDs with significant occlusions and clutter for both Manhattan and non-Manhattan-Word buildings. Lastly, unlike deep learning techniques, our method does not require extensive training datasets; it can directly segment PCDs on the instance level and identify instance IDs by leveraging IFC models in the Scan-vs-BIM context. With the high-precision result of instance matching, the proposed method can be employed for construction progress monitoring and quality control for project management.

Although our method performs high-precision results, there are several factors that can influence instance segmentation in the Scan-vs-BIM environment. First, too large intervals among scanned points can cause false negative results when segmenting point clusters. For example, in our cases, points for planar walls will be considered as noise and removed if the distance is larger than around 0.1m based on experience, the right part of TUB2-PW2's result cluster in Figure 12 is an example. This situation typically arises when the distance between the scanner and the object is considerable. Due to the limited precision of the scanner and the increased scanning distance, the sampled points become sparse and can be possibly detected as noise by our method. Second, two instances with seamless and smooth surfaces from IFC models are not easily distinguishable in PCDs and thus will cause false-positive segmentation results. Additionally, for cuboid columns, the size restraint algorithm may not be effective in removing points of signs if one surface of a column is completely occluded and this surface cannot be detected in PCD. This can lead to false positive results. Finally, the floor junction may also be included during segmenting if the edge of an interior wall does not connect to another wall, leading to false positive results. A potential solution is to first determine whether the wall's edge connects to another wall; if not, further segmentation methods need to be developed to address this problem.

6. Conclusion

This paper first examines the current SOTA methods with their strengths and limitations in detecting and segmenting structural object instances from PCD in the Scan-vs-BIM context. Such insights are helpful for future researchers aiming to further improve on these methods. Subsequently, we proposed a novel method that can rapidly, efficiently, and precisely segment common planar, curved, and linear structural instances from PCD in complex and real-world environments. For the academic contributions, the proposed method exhibits robustness in scenarios where: 1) the input PCD contains numerous occlusions and clutter; 2) the as-built object instances have significant deviations from the as-designed model in terms of position, orientation, and scale; 3) the as-designed or as-built are Manhattan-world buildings or non-Manhattan-world buildings. The proposed method following a top-down idea enhances the current SOTA in computer vision-based instance matching in the Scan-vs-BIM context. It has wide-ranging academic implications for studies in similar or allied domains, and potentially can act as a benchmark for upcoming research. For practical contributions, the automation of DI instance matching in PCD can significantly reduce manual checking time, leading to faster project progress monitoring and quality control at the construction stage. As this matching algorithm can be implemented at different timestamps

during the building's construction, the discrepancies between DI and as-built status can be detected and reported timely. It can help avoid costly rectifications in the later stages of construction. Finally, the matching result can aid in the updating of DT for infrastructure, which is crucial for modern facility management and predictive maintenance.

In future research directions, standardising the matching solution for complex mechanical and electrical instances will broaden the method's applicability. The automation of matching DI top frequent object classes, including pipe segments, duct segments, pipe joints, terminals, and lighting fixtures, will make the method more generic to support the geometric DT's maintenance. Also, optimizing or automating the registration process of complex PCD with its respective DI file can enhance the congruence between the physical and digital realms. Lastly, addressing the inconsistencies such as gaps and truncation in extracted point clusters will lead to more refined models for streamlining the process of updating geometric DTs. For future potential applications, the proposed method integrated with the updated DT can aid in monitoring construction progress, identifying discrepancies early, and ensuring adherence to design specifications. Building managers can utilize the method to maintain up-to-date DT of facilities, helping in predictive maintenance and space optimization. Furthermore, updating a geometric DT automatically and dynamically can foster better collaboration between architects, engineers, and construction professionals, ensuring everyone works from the most accurate and up-to-date data during a building's lifecycle.

7. Acknowledgement

This work is funded by the European Commission's Horizon 2020 for the CBIM (Cloud-based Building Information Modelling) European Training Network under agreement No. 860555.

Reference:

[1] V. Drobnyi, Z. Hu, Y. Fathy, I. Brilakis, Construction and maintenance of building geometric digital twins: State of the art review, Sensors 23 (9) (2023) 4382. https://doi.org/10.3390/s23094382.

[2] R. Sacks, I. Brilakis, E. Pikas, H. S. Xie, M. Girolami, Construction with digital twin information systems, Data-Centric Engineering 1 (2020). https://doi:10.1017/dce.2020.16.

[3] F. Bosche, A. Guillemet, Y. Turkan, C. T. Haas, R. Haas, Tracking the built status of mep works: Assessing the value of a scan-vs-bim system, Journal of Computing in Civil Engineering 28 (4) (2014) 05014004. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000343.

[4] Z. Hu, Y. Fathy, I. Brilakis, Geometry updating for digital twins of buildings: A review to derive a new geometry-based object class hierarchy, in: Proceedings of the 2022 European Conference on Computing in Construction, Vol. 3 of Computing in Construction, 2022, pp. 270–277. http://doi:10.35490/EC3.2022.155.

[5] F. Bosche, Plane-based registration of construction laser scans with 3d/4d building models, Advanced Engineering Informatics 26 (1) (2012) pp. 90–102. https://doi.org/10.1016/j.aei.2011.08.009.

[6] N. A. Sheik, G. Deruyter, P. Veelaert, Plane-based robust registration of a building scan with its bim, Remote Sensing 14 (9) (2022) 1979. https://doi.org/10.3390/rs14091979.

[7] T. Kaiser, C. Clemen, H.-G. Maas, Automatic co-registration of photogrammetric point clouds with digital building models, Automation in Construction 134 (2022) 104098. https://doi.org/10.1016/j.autcon.2021.104098.

977 [8] M. Bueno, F. Bosche, H. Gonzalez-Jorge, J. Martınez-Sanchez, P. Arias, 4-plane congruent sets for
978 automatic registration of as-is 3d point clouds with 3d bim models, Automation in Construction 89 (2018) pp.
979 120–134. https://doi.org/10.1016/j.autcon.2018.01.014.

980 [9] J. Yang, H. Li, Y. Jia, Go-icp: Solving 3d registration efficiently and globally optimally, in: Proceedings of
981 the IEEE International Conference on Computer Vision, 2013, pp. 1457–1464.
982 https://doi.org/10.1109/ICCV.2013.184.

983 [10] C. H. P. Nguyen, Y. Choi, Comparison of point cloud data and 3d cad data for on-site dimensional
984 inspection of industrial plant piping systems, Automation in Construction 91 (2018) pp. 44–52.
985 https://doi.org/10.1016/j.autcon.2018.03.008.

986 [11] C. Liu, M. Guang, S. Yu, Point cloud and bim model registration based on genetic algorithm and icp
987 algorithm, in: Journal of Physics: Conference Series, Vol. 2132, IOP Publishing, 2021, p. 012007.
988 https://doi.org/10.1088/1742-6596/2132/1/012007.

989 [12] J. Chen, S. Li, W. Lu, Align to locate: Registering photogrammetric point clouds to bim for robust indoor
990 localization, Building and Environment 209 (2022) 108675. https://doi.org/10.1016/j.buildenv.2021.108675.

991 [13] W. Wohlkinger, M. Vincze, Ensemble of shape functions for 3d object classification, in: 2011 IEEE
992 International Conference on Robotics and Biomimetics, IEEE, 2011, pp. 2987–2992.
993 https://doi.org/10.1109/ROBIO.2011.6181760.

994 [14] J. Chen, Y. Fang, Y. K. Cho, C. Kim, Principal axes descriptor for automated construction-equipment
995 classification from point clouds, Journal of Computing in Civil Engineering 31 (2) (2017) 04016058.
996 https://doi.org/10.1061/(ASCE)CP.1943-5487.0000628.

997 [15] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, C. Cadena, Segmatch: Segment based place
998 recognition in 3d point clouds, in: 2017 IEEE International Conference on Robotics and Automation (ICRA),
999 IEEE, 2017, pp. 5266–5272. https://doi.org/10.1109/ICRA.2017.7989618.

1000 [16] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3d point cloud based object maps for
1001 household environments, Robotics and Autonomous Systems 56 (11) (2008) pp. 927–941.
1002 https://doi.org/10.1016/j.robot.2008.08.005.

1003 [17] F. Bosche, C. T. Haas, Automated retrieval of 3d cad model objects in construction range images,
1004 Automation in Construction 17 (4) (2008) pp. 499–512. https://doi.org/10.1016/j.autcon.2007.09.001.

1005 [18] Y. Turkan, F. Bosche, C. T. Haas, R. Haas, Automated progress tracking using 4d schedule and 3d sensing
1006 technologies, Automation in Construction 22 (2012) pp. 414–421. https://doi.org/10.1016/j.autcon.2011.10.003.

1007 [19] Y. Turkan, F. Bosche, C. T. Haas, R. Haas, Toward automated earned value tracking using 3d imaging
1008 tools, Journal of Construction Engineering and Management 139 (4) (2013) pp. 423–433.
1009 https://doi.org/10.1061/(ASCE)CO.1943-7862.0000629.

1010 [20] Y. Turkan, F. Bosche, C. T. Haas, R. Haas, Tracking of secondary and temporary objects in structural
1011 concrete work, Construction Innovation 14 (2) (2014) pp. 145–167. https://doi.org/10.1108/CI-12-2012-0063.

1012 [21] C. Zhang, D. Arditi, Automated progress control using laser scanning technology, Automation in
1013 Construction 36 (2013) pp. 108–116. https://doi.org/10.1016/j.autcon.2013.08.012.

1014 [22] T. Gao, S. Ergan, B. Akinci, J. Garrett, Evaluation of different features for matching point clouds to
1015 building information models, Journal of Computing in Civil Engineering 30 (1) (2016) 04014107.
1016 https://doi.org/10.1061/(ASCE)CP.1943-5487.0000425.

1017 [23] H. Tran, K. Khoshelham, Building change detection through comparison of a lidar scan with a building
1018 information model., International Archives of the Photogrammetry, Remote Sensing & Spatial Information
1019 Sciences (2019). https://doi.org/10.5194/isprs-archives-XLII-2-W13-889-2019.

1020 [24] G. Bariczova, J. Erdelyi, R. Honti, L. Tomek, Wall structure geometry verification using tls data and bim
1021 model, Applied Sciences 11 (24) (2021) 11804. https://doi.org/10.3390/app112411804.

[25] M. Bassier, J. Vermandere, H. De Winter, Linked building data for construction site monitoring: A test case, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2 (2022) pp. 159–165. https://doi.org/10.5194/isprs-annals-V-2-2022-159-2022.

[26] S. Park, S. Ju, S. Yoon, M. H. Nguyen, J. Heo, An efficient data structure approach for bim-to-point-cloud change detection using modifiable nested octree, Automation in Construction 132 (2021) 103922. https://doi.org/10.1016/j.autcon.2021.103922.

[27] C. Kim, H. Son, C. Kim, Automated construction progress measurement using a 4d building information model and 3d data, Automation in Construction 31 (2013) pp. 75–82. https://doi.org/10.1016/j.autcon.2012.11.041.

[28] J. Guo, Q. Wang, J.-H. Park, Geometric quality inspection of prefabricated mep modules with 3d laser scanning, Automation in Construction 111 (2020) 103053. https://doi.org/10.1016/j.autcon.2019.103053.

[29] C. Rausch, C. Haas, Automated shape and pose updating of building information model elements from 3d point clouds, Automation in Construction 124 (2021) 103561. https://doi.org/10.1016/j.autcon.2021.103561.

[30] C. R. Qi, H. Su, K. Mo, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660. https://doi.org/10.1109/CVPR.2017.16.

[31] D. Robert, B. Vallet, L. Landrieu, Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5575–5584. https://doi.org/10.1109/CVPR52688.2022.00549.

[32] Q. Wang, S. Shi, J. Li, W. Jiang, X. Zhang, Window normalization: Enhancing point cloud understanding by unifying inconsistent point densities, arXiv preprint arXiv:2212.02287 (2022). https://doi.org/10.48550/arXiv.2212.02287.

[33] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, Pointcnn: Convolution on x-transformed points, Advances in Neural Information Processing Systems 31 (2018). https://doi.org/10.48550/arXiv.1801.07791.

[34] Y. Pan, A. Braun, A. Borrmann, I. Brilakis, 3d deep-learning-enhanced void-growing approach in creating geometric digital twins of buildings, Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction (2023) pp. 24–40. https://doi.org/10.1680/jsmic.21.00035.

[35] H. Lin, X. Zheng, L. Li, F. Chao, S. Wang, Y. Wang, Y. Tian, R. Ji, Meta architecture for point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 17682–17691. https://doi.org/10.48550/arXiv.2211.14462.

[36] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, B. Ghanem, Pointnext: Revisiting pointnet++ with improved training and scaling strategies, Advances in Neural Information Processing Systems 35 (2022) pp. 23192–23204. https://doi.org/10.48550/arXiv.2206.04670.

[37] M. Kolodiazhnyi, D. Rukhovich, A. Vorontsova, A. Konushin, Top-down beats bottom-up in 3d instance segmentation, arXiv preprint arXiv:2302.02871 (2023). https://doi.org/10.48550/arXiv.2302.02871.

[38] M. Zhong, X. Chen, X. Chen, G. Zeng, Y. Wang, Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation, in: 2022 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2022, pp. 1–6. https://doi.org/10.1109/ICME52920.2022.9859996.

[39] J. Sun, C. Qing, J. Tan, X. Xu, Superpoint transformer for 3d scene instance segmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 2393–2401. https://doi.org/10.1609/aaai.v37i2.25335.

[40] T. Vu, K. Kim, T. M. Luu, T. Nguyen, C. D. Yoo, Softgroup for 3d instance segmentation on point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2708–2717. https://doi.org/10.1109/CVPR52688.2022.00273.

[41] K.Pleansamai, & K. Chaiyasarn. M-estimator sample consensus planar extraction from image-based 3D point cloud for building information modelling. GEOMATE Journal, (2019) 17(63), pp. 69-76. https://doi.org/10.21660/2019.63.09667.

[42] O. Chum, J. Matas, Matching with prosac-progressive sample consensus, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, IEEE, 2005, pp. 220–226. https://doi.org/10.1109/CVPR.2005.221.

[43] B. Zhao, X. Hua, K. Yu, W. Xuan, X. Chen, & W. Tao, Indoor point cloud segmentation using iterative gaussian mapping and improved model fitting. IEEE Transactions on Geoscience and Remote Sensing, (2020) 58(11), pp. 7890-7907. https://doi.org/10.1109/TGRS.2020.2984943.

[44] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, Dbscan revisited, revisited: why and how you should (still) use dbscan, ACM Transactions on Database Systems (TODS) 42 (3) (2017) pp. 1–21. https://doi.org/10.1145/3068335.

[45] H. Chen, M. Liang, W. Liu, W. Wang, & P.X. Liu. An approach to boundary detection for 3D point clouds based on DBSCAN clustering. Pattern Recognition, (2022) 124, 108431. https://doi.org/10.1016/j.patcog.2021.108431.

[46] T. Akanbi, J. Zhang, Y.-C. Lee, Data-driven reverse engineering algorithm development method for developing interoperable quantity takeoff algorithms using ifc-based bim, Journal of Computing in Civil Engineering 34 (5) (2020) 04020036. https://doi.org/10.1061/(ASCE)CP.1943-5487.000090.

[47] X. Tan, A. Hammad, P. Fazio, Automated code compliance checking for building envelope design, Journal of Computing in Civil Engineering 24 (2) (2010) pp. 203–211. https://doi.org/10.1061/(ASCE)0887-3801(2010)24:2(203).

[48] T. Gerrish, K. Ruikar, M. Cook, M. Johnson, M. Phillip, C. Lowry, Bim application to building energy performance visualisation and management: challenges and potential, Energy and Buildings 144 (2017) 218–228. https://doi.org/10.1016/j.enbuild.2017.03.032.

[49] F. Tombari, S. Salti, L. Di Stefano, Unique shape context for 3d data description, in: Proceedings of the ACM Workshop on 3D Object Retrieval, 2010, pp. 57–62. https://doi.org/10.1145/1877808.1877821.

[50] H. Mansor, S. Shukor, R. Wong, An overview of object detection from building point cloud data, in: Journal of Physics: Conference Series, Vol. 1878, IOP Publishing, 2021, p. 012058. https://doi.org/10.1088/1742-6596/1878/1/012058.

[51] K. Khoshelham, L. D. Vilarino, M. Peter, Z. Kang, D. Acharya, The isprs benchmark on indoor modelling, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 42 (2) (2017) W7. https://doi.org/10.5194/isprs-archives-XLII-2-W7-367-2017.

[52] K. Khoshelham, H. Tran, D. Acharya, L. Dıaz Vilarino, Z. Kang, S. Dalyot, The isprs benchmark on indoor modelling–preliminary results, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 43 (2020) pp. 207–211. https://doi.org/10.5194/isprs-archives-XLIII-B5-2020-207-2020.

[53] M. Trzeciak, K. Pluta, Y. Fathy, L. Alcalde, S. Chee, A. Bromley, I. Brilakis, P. Alliez, Conslam: Periodically collected real-world construction dataset for slam and progress monitoring, in: Computer Vision–European Conference on Computer Vision 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII, Springer, 2023, pp. 317–331. https://doi.org/10.1007/978-3-031-25082-8_21.

[54] J. Wang, R. Lindenbergh, M. Menenti, Sigvox – a 3d feature matching algorithm for automatic street object recognition in mobile laser scanning point clouds, ISPRS Journal of Photogrammetry and Remote Sensing 128 (2017) pp. 111–129. https://doi.org/10.1016/j.isprsjprs.2017.03.012.

[55] D. Shepherd, The BIM Management Handbook, Routledge, 2019. ISBN 9781859466056.

[56] A. Borrmann, J. Beetz, C. Koch, T. Liebich, S. Muhic, Industry foundation classes: A standardized data model for the vendor-neutral exchange of digital building models, Building Information Modeling: Technology Foundations and Industry Practice (2018) pp. 81–126. https://doi.org/10.1007/978-3-319-92862-3_5.

[57] Y. Ge, H. Tang, D. Xia, L. Wang, B. Zhao, J. W. Teaway, H. Chen, T. Zhou, Automated measurements of discontinuity geometric properties from a 3d-point cloud based on a modified region growing algorithm, Engineering Geology 242 (2018) pp. 44–54. https://doi.org/10.1016/j.enggeo.2018.05.007.

[58] M. Parsamehr, U. S. Perera, T. C. Dodanwala, P. Perera, R. Ruparathna, A review of construction management challenges and bim-based solutions: perspectives from the schedule, cost, quality, and safety

management, Asian Journal of Civil Engineering 24 (1) (2023) pp. 353–389. https://doi.org/10.1007/s42107-022-00501-4.

[59] Z. Turk, Structured analysis of ict adoption in the European construction industry, International Journal of Construction Management 23 (5) (2023) pp. 756–762. https://doi.org/10.1080/15623599.2021.1925396.

[60] Pan, Z. Hu, I. Brilakis, Digital twins and their roles in building deep renovation life cycle, in: Disrupting Buildings: Digitalisation and the Transformation of Deep Renovation, Springer International Publishing Cham, 2023, pp. 83–96. https://doi.org/10.1007/978-3-031-32309-6_6.

[61] J. Fingland, Preparing for Digital Transformation in the AEC Industry. Retrieved from https://constructible.trimble.com/construction-industry/preparing-for-digital-transformation-in-the-aec-industry. (18 March 2019) Accessed on 17 October 2023.

[62] Multiproject, Why do construction projects frequently go over budget? Retrieved from https://multiproject.org/why-do-construction-projects-frequently-go-over-budget/. (28 June, 2021) Accessed on 17 October 2023.

[63] KPMG, Global Construction Survey 2015 Climbing the curve. Retrieved from https://assets.kpmg.com/content/dam/kpmg/pdf/2015/05/construction-survey-201502.pdf. (March 2015) Accessed on 17 October 2023.

[64] OpenSpace. Retrieved from https://www.openspace.ai/. Accessed on 17 October 2023.

[65] Buildots. Retrieved from https://buildots.com/. Accessed on 17 October 2023.

[66] A. Abdelkader, C. L. Bajaj, M. S. Ebeida, A. H. Mahmoud, S. A. Mitchell, J. D. Owens, A. A. Rushdi, Vorocrust: Voronoi meshing without clipping, ACM Transactions on Graphics (TOG) 39 (3) (2020) pp. 1–16. https://doi.org/10.1145/3337680.

[67] Y.-M. Hong, J.-H. Zhang, C.-Y. Chen, B. C. W¨unsche, H.-J. Chien, T.-K. Ying, Surface reconstruction of 3d objects using local moving least squares and kd trees, in: 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ), IEEE, 2017, pp. 1–6. https://doi.org/10.1109/IVCNZ.2017.8402500.

[68] A. Rashidi, I. Brilakis, Point cloud data cleaning and refining for 3d as-built modeling of built infrastructure, in: Construction Research Congress 2016, 2016, pp. 919–929. https://doi.org/10.1061/9780784479827.093.

[69] R. B. Rusu, S. Cousins, 3d is here: Point cloud library (pcl), in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 1–4. https://doi.org/10.1109/ICRA.2011.5980567.

[70] J. Chen, Y. K. Cho, Point-to-point comparison method for automated scan-vs-bim deviation detection, in: Proceedings of the 2018 17th International Conference on Computing in Civil and Building Engineering, Tampere, Finland, 2018, pp. 5–7. https://programme.exordo.com/icccbe2018/delegates/presentation/303/.

[71] P. V. Hough, Machine analysis of bubble chamber pictures, in: International Conference on High Energy Accelerators and Instrumentation, CERN, 1959, 1959, pp. 554–556. https://cir.nii.ac.jp/crid/1572824500736446720.

[72] M. Ahmed, C. T. Haas, R. Haas, Autonomous modeling of pipes within point clouds, in: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, Vol. 30, IAARC Publications, 2013, pp. 1093-1100. https://doi.org/10.22260/ISARC2013/0120.

[73] M. F. Ahmed, C. T. Haas, R. Haas, Automatic detection of cylindrical objects in built facilities, Journal of Computing in Civil Engineering 28 (3) (2014) 04014009. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000329.

[74] F. Bosch´e, M. Ahmed, Y. Turkan, C. T. Haas, R. Haas, The value of integrating scan-to-bim and scan-vs-bim techniques for construction monitoring using laser scanning and bim: The case of cylindrical mep components, Automation in Construction 49 (2015) pp. 201–213. https://doi.org/10.1016/j.autcon.2014.05.014.

[75] V. Stojanovic, R. Richter, J. D¨ollner, M. Trapp, Comparative visualization of bim geometry and corresponding point clouds, Building Information Systems in the Construction Industry 13 (2018) pp. 12-23. https://doi.org/10.2495/SDP-V13-N1-12-23.

[76] V. S. Kalasapudi, Y. Turkan, P. Tang, Toward automated spatial change analysis of mep components using 3d point clouds and as-designed bim models, in: 2014 2nd International Conference on 3D Vision, Vol. 2, IEEE, 2014, pp. 145–152. https://doi.org/10.1109/3DV.2014.105.

[77] R. Schnabel, R. Wahl, R. Klein, Efficient ransac for point-cloud shape detection, in: Computer Graphics Forum, Vol. 26, Wiley Online Library, 2007, pp. 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x.

[78] M.-K. Kim, Q. Wang, J.-W. Park, J. C. Cheng, H. Sohn, C.-C. Chang, Automated dimensional quality assurance of full-scale precast concrete elements using laser scanning and bim, Automation in Construction 72 (2016) pp. 102–114. https://doi.org/10.1016/j.autcon.2016.08.035.

[79] C. H. P. Nguyen, Y. Choi, Comparison of point cloud data and 3d cad data for on-site dimensional inspection of industrial plant piping systems, Automation in Construction 91 (2018) pp. 44–52. https://doi.org/10.1016/j.autcon.2018.03.008.

[80] L. Yang, Y. Li, X. Li, Z. Meng, H. Luo, Efficient plane extraction using normal estimation and ransac from 3d point cloud, Computer Standards & Interfaces 82 (2022) 103608. https://doi.org/10.1016/j.csi.2021.103608.

[81] S. I. Zolanvari, D. F. Laefer, Slicing method for curved facade and window extraction from point clouds, ISPRS Journal of Photogrammetry and Remote Sensing 119 (2016) pp. 334–346. https://doi.org/10.1016/j.isprsjprs.2016.06.011.

[82] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) pp. 381–395. https://doi.org/10.1145/358669.358692.

[83] P. H. Torr, A. Zisserman, MLESAC: A new robust estimator with application to estimating image geometry, Computer Vision and Image Understanding 78 (1) (2000) pp. 138–156. https://doi.org/10.1006/cviu.1999.0832.

[84] K. Babacan, J. Jung, A. Wichmann, B. Jahromi, M. Shahbazi, G. Sohn, M. Kada, Towards object driven floor plan extraction from laser point cloud, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 41 (2016) pp. 3–10. https://doi.org/10.5194/isprs-archives-XLI-B3-3-2016.

[85] S.-Y. An, L.-K. Lee, S.-Y. Oh, Fast incremental 3d plane extraction from a collection of 2d line segments for 3d mapping, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 4530–4537. https://doi.org/10.1109/IROS.2012.6386004.

[86] G. Zhang, P. Vela, I. Brilakis, Detecting, fitting, and classifying surface primitives for infrastructure point cloud data, in: Journal of Computing in Civil Engineering (2013), 2013, pp. 589–596. https://doi.org/10.1061/9780784413029.074.

[87] A. Nurunnabi, G. West, D. Belton, Robust outlier detection and saliency features estimation in point cloud data, in: 2013 International Conference on Computer and Robot Vision, 2013, pp. 98–105. https://doi.org/10.1109/CRV.2013.28.

[88] K. Krishna, M. N. Murty, Genetic k-means algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29 (3) (1999) pp. 433–439. https://doi.org/10.1109/3477.764879.

[89] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8) (1995) pp. 790–799. https://doi.org/10.1109/34.400568.

[90] E. K. Tokuda, C. H. Comin, L. d. F. Costa, Revisiting agglomerative clustering, Physica A: Statistical Mechanics and its Applications 585 (2022) 126433. https://doi.org/10.1016/j.physa.2021.126433.

[91] L. McInnes, J. Healy, S. Astels, hdbscan: Hierarchical density based clustering., Journal of Open Source Software 2 (11) (2017) 205. https://doi.org/10.21105/joss.00205.

1207 [92] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3d
1208 shape descriptors, in: Symposium on Geometry Processing, Vol. 6, 2003, pp. 156–164.
1209 https://doi.org/10.2312/SGP/SGP03/156-165.

1210 [93] J. Chen, Y. Fang, Y. K. Cho, Performance evaluation of 3d descriptors for object recognition in
1211 construction applications, Automation in Construction 86 (2018) pp. 44–52.
1212 https://doi.org/10.1016/j.autcon.2017.10.033.

1213 [94] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, IEEE
1214 Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) pp. 433–449.
1215 https://doi.org/10.1109/34.765655.

1216 [95] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in:
1217 Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece,
1218 September 5-11, 2010, Proceedings, Part III 11, Springer, 2010, pp.356–369. https://doi.org/10.1007/978-3-642-
1219 15558-1_26.

Appendix: Visualization of all matching results (Figures 13 - 15).

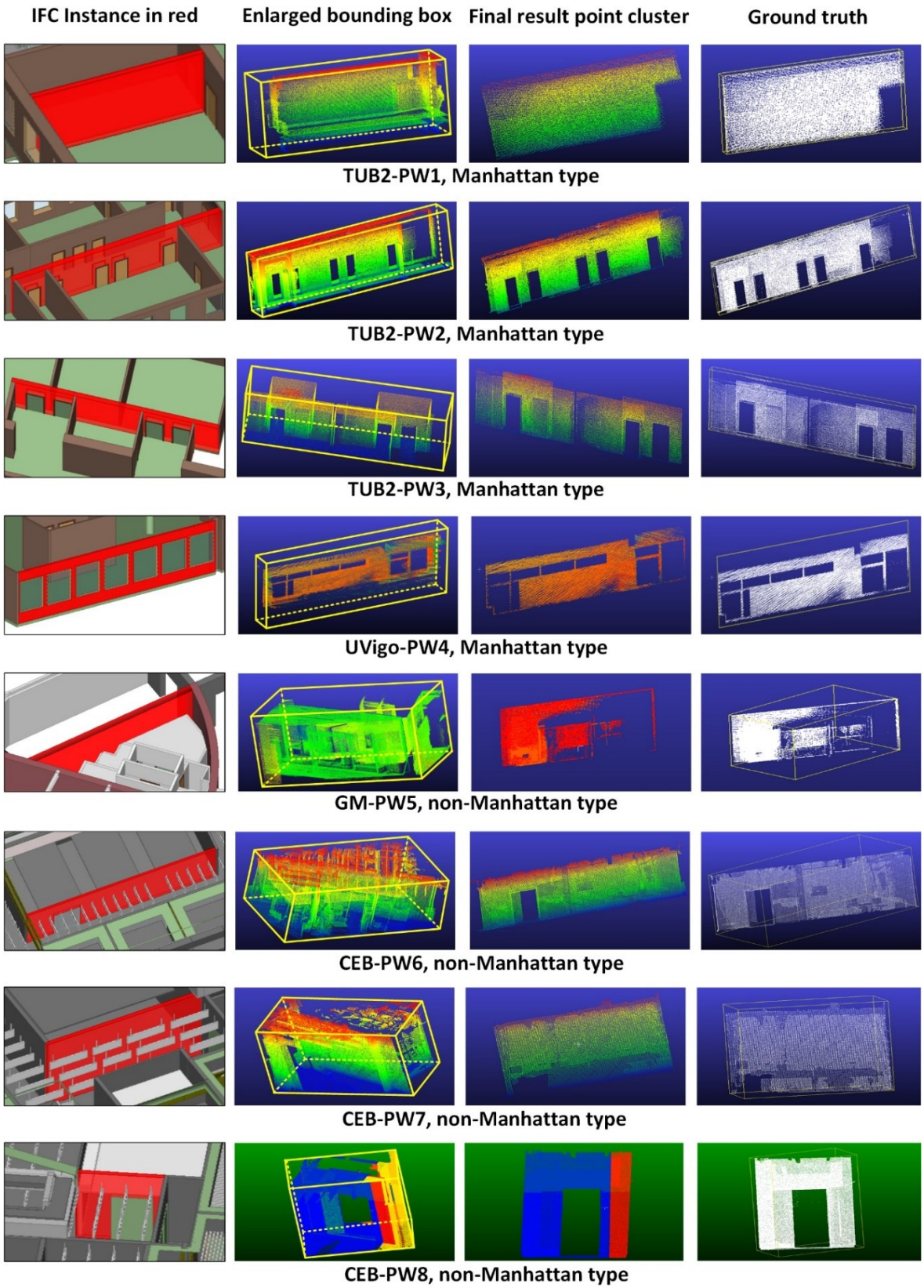


Figure 13 PCD matching results for planar walls (PW).

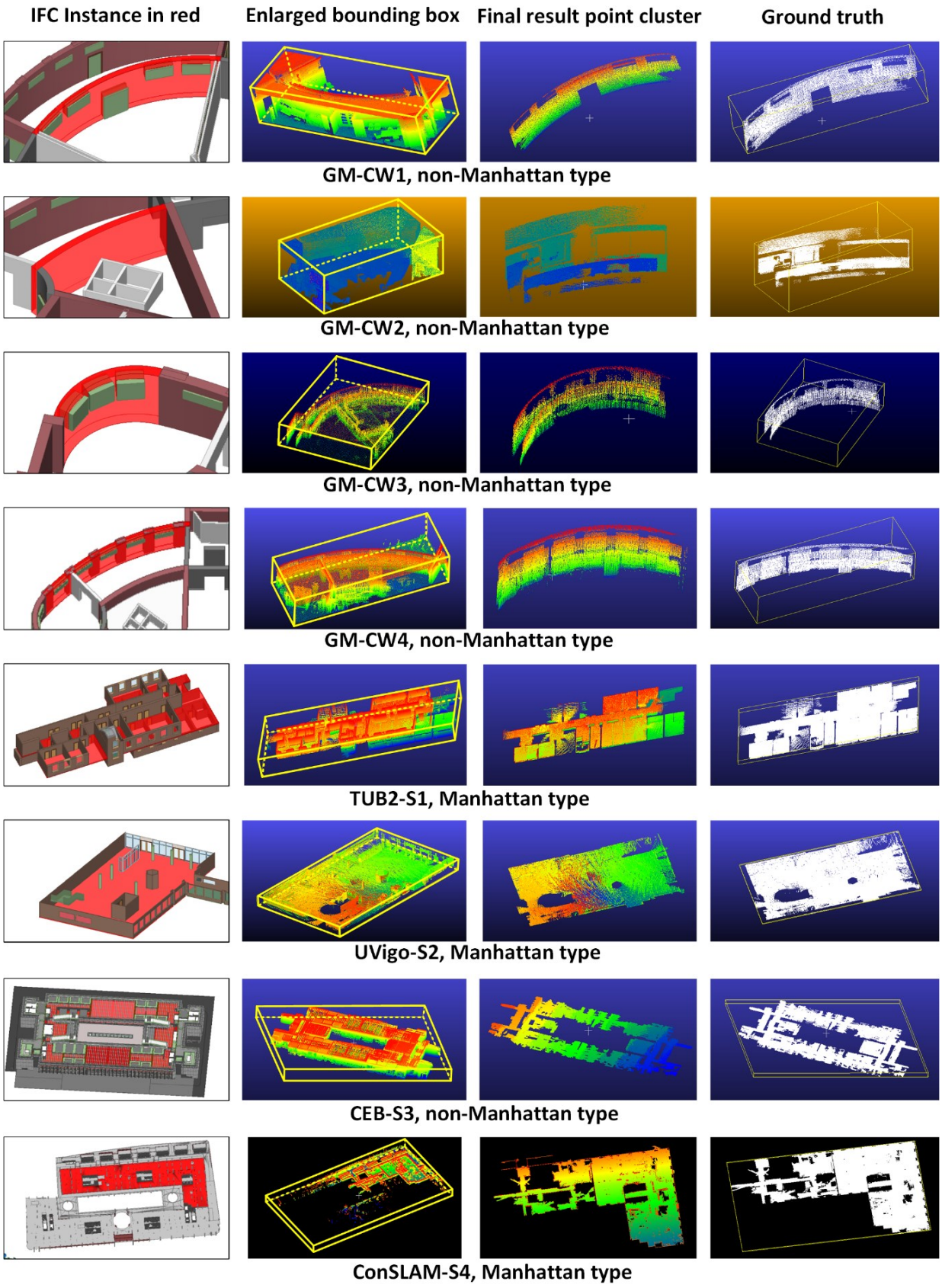


Figure 14 PCD matching results for curved walls (CW) and Slabs (S).

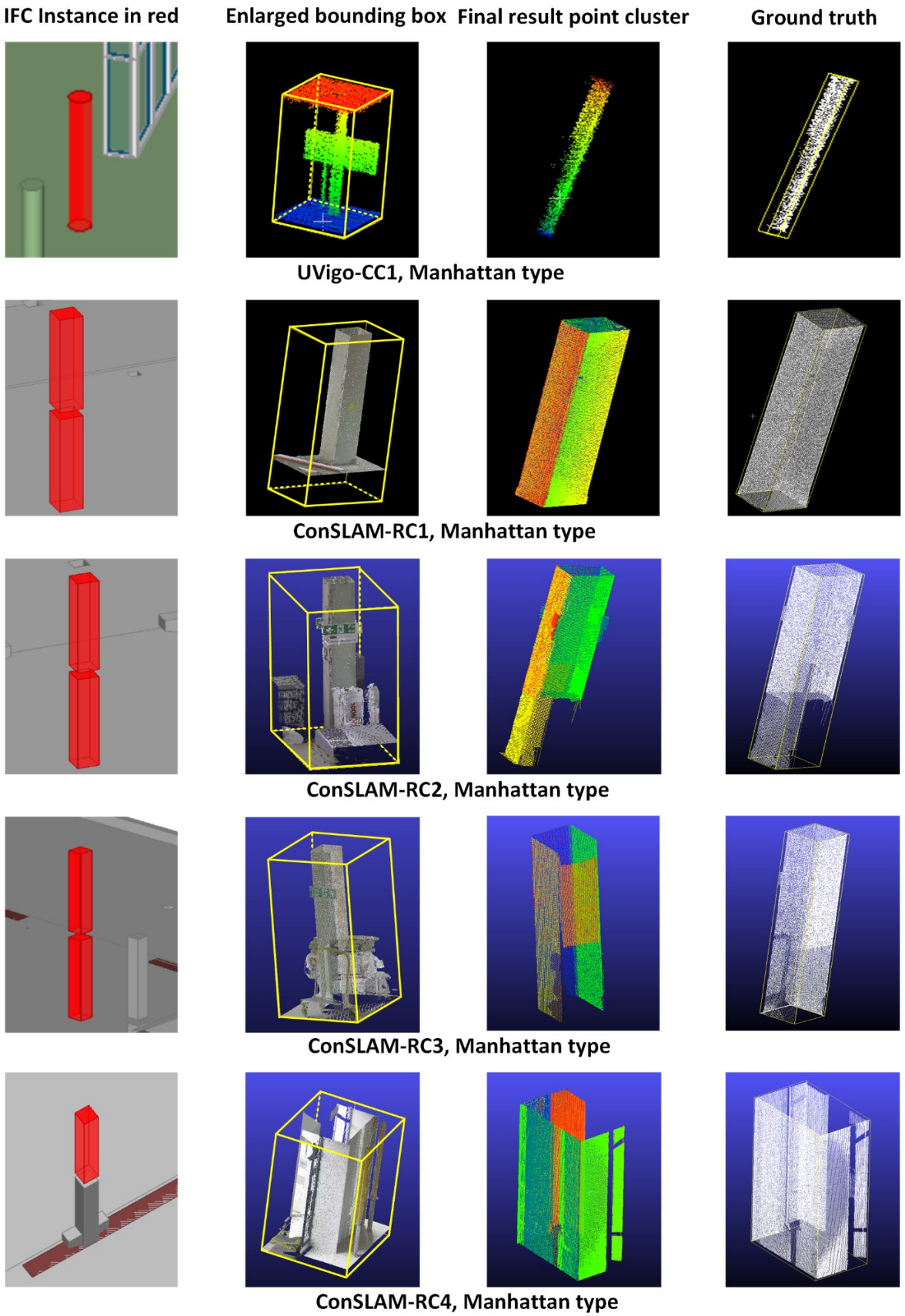


Figure 15 PCD matching results for cylindrical columns (CC) and cuboid columns (RC).