# An Automatic Update Framework for As-designed Pipeline BIM Model Based on Laser Scanning Point Cloud

**Xinru Wang[1], Bin Yang[1] and Tianjia Lu[1]**

[1] College of Civil Engineering, Tongji University, China
2430797@tongji.edu.cn, yangbin@tongji.edu.cn, 2454442@tongji.edu.cn

**Abstract –**

**Building Information Modeling (BIM) is widely used in construction management, particularly for mechanical, electrical, and plumbing (MEP) systems, where accurate pipeline information is critical for efficient operation and maintenance. In practice, however, as-built pipelines often deviate from the as-designed BIM models, and manually updating these models is time-consuming. This study proposes an automatic pipeline for BIM updating, which leverages laser-scanned point clouds together with as-designed BIM priors. The framework integrates point cloud semantic segmentation, parameter extraction, and spatial topology analysis to establish component correspondences and generate an updated BIM model. A case study in a complex MEP environment demonstrates that this framework can robustly update pipeline BIMs, reducing modeling time from 3-5 days to approximately 3 hours, thereby improving the efficiency of BIM-based MEP management.**

**Keywords –**

**Point cloud; Building Information Modeling; As-designed model; Automatic modeling**

## 1 Introduction

BIM has become a key enabler for efficient lifecycle management in the architecture, engineering and construction industry, especially when integrated with Internet of Things (IoT) for real-time data acquisition and facility operation optimization. In smart construction, enriched BIM models are increasingly required to support performance monitoring and maintenance of MEP systems [1]. However, as-designed BIM models often deviate from as-built conditions due to construction tolerances and on-site modifications, which significantly degrades the reliability of BIM-based MEP management.

In practice, BIM models are often generated manually from component specifications and installation records, causing low efficiency in complex MEP environments. Laser scanning can capture dense point clouds that accurately represent as-built environment [2]. However, raw point clouds lack semantic and topological information and cannot be directly used for engineering applications [3]. Although automated scan-to-BIM approaches have been developed [4-5], many rely primarily on local geometric features, which limits their robustness in reconstructing complex MEP systems.

To address these limitations, this paper proposes an automatic framework for updating as-designed pipeline BIM models using laser-scanned point clouds. The main contributions are summarized below. First, a semantic segmentation strategy is developed by integrating DBSCAN with deep learning to improve recognition of multi-scale pipeline components in cluttered MEP environments. Second, a logic-based geometric refinement method is proposed by utilizing as-designed BIM priors to correct coordinate deviations caused by noise and occlusion. Finally, the research bridges the semantic gap between unstructured as-built data and structured as-designed components by applying graph isomorphism, thereby enabling a fully closed-loop parametric update. Overall, the integrated workflow minimizes manual intervention and maximizes MEP modeling fidelity at the same time.

## 2 Related Work

### 2.1 Point Cloud Segmentation

Point cloud segmentation is critical for automated scan-to-BIM workflows but remains challenging in complex MEP environments [6-7]. Deep learning approaches such as PointNet++ and transformer-based networks [8] have improved segmentation performance. Synthetic or annotated datasets have also been explored to address data scarcity [9-10]. However, robustness is still limited by scale variation and occlusion, motivating hybrid methods that combine spatial preprocessing with learning-based segmentation.

### 2.2 Geometry Parameter Identification

Following segmentation, parameter identification extracts geometric information from point clouds for parametric BIM reconstruction. Early methods mainly relied on manual feature extraction and geometric fitting techniques such as RANSAC [11]. Although effective for regular shapes, these approaches depend on prior knowledge and manual tuning, limiting their applicability to complex MEP components with irregular geometry. Recent studies have therefore shifted toward deep learning-based parameter extraction, which learns structural representations directly from data and can handle both regular and irregular components without explicit feature engineering [12]. This data-driven paradigm improves the automation and robustness of parametric BIM reconstruction in complex MEP systems.

### 2.3 Automatic Modeling

Automatic modeling aims to convert geometric and semantic data into parametric BIMs. However, accurately reconstructing as-built pipelines remains challenging. Early Scan-to-BIM frameworks focused on detecting spatial discrepancies [13]. Subsequent studies [14-15] leveraged 3D point clouds for automated geometric modeling of cylindrical MEP elements. Recently, combining semantic segmentation with geometric extraction has enabled transformation from point cloud to BIM via visual programming (e.g., Dynamo) [4]. Furthermore, hybrid methods fusing image semantics with LiDAR precision enhance robustness in complex environments [16]. These studies collectively provide a solid foundation for automated pipeline reconstruction.

Despite recent progress in Scan-to-BIM research, several limitations still restrict its practical application in complex MEP environments. First, many existing approaches rely heavily on local geometric features, which reduces robustness in scenes with severe occlusion and dense pipeline layouts. Second, prior information from as-designed BIM models is not sufficiently utilized. Third, current methods still face challenges in achieving automated synchronization between as-designed and as-built pipeline models.

## 3 Methodology

This study proposes a framework to automatically update as-designed pipeline BIM model with laser-scanned as-built point clouds. As shown in Figure 1, the framework takes data acquired from a laser scanner as inputs. After point cloud preprocessing, DBSCAN and PointNet++ perform semantic segmentation. On this basis, parameter extraction with logical reasoning derives the geometric parameters of the as-built pipelines, while spatial topology analysis is used to establish the mapping between point-cloud instances and BIM components. Finally, as-built parameters are used to automatically update the as-designed BIM model, thereby generating an updated as-built pipeline BIM. The details of each step are presented in the following sections.

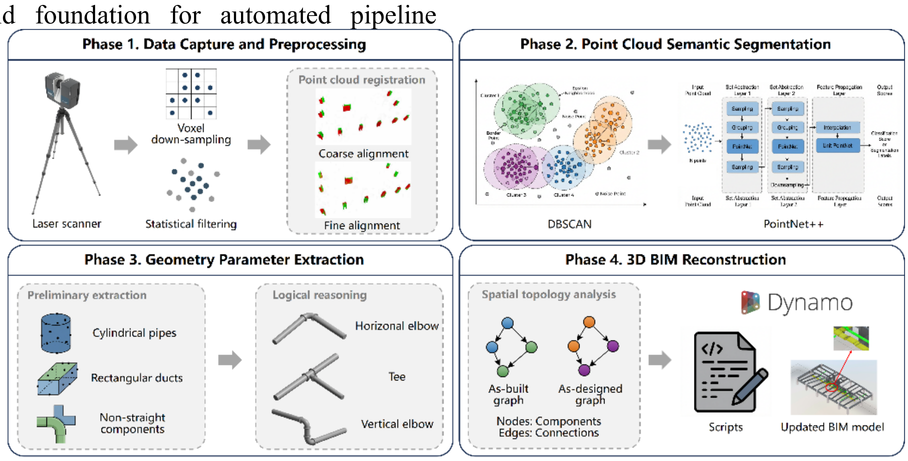


Figure 1. Proposed methodology framework

### 3.1 Data Capture and Preprocessing

High-quality point clouds are essential for accurate BIM updating, as they directly influence all subsequent steps. In MEP scenarios, dense and accurate laser scans enable reliable identification of pipeline geometry and connections, improving the robustness of BIM updating [17]. Point cloud quality primarily depends on the data acquisition strategy and preprocessing procedures, which together determine point density and noise level.

This study uses a terrestrial laser scanner (Trimble TX5) to obtain the as-built data of the MEP system, its detailed technical specifications are summarized in Table 1. Multi-station laser scanning is employed to ensure adequate coverage and to mitigate the impact of occlusions in indoor MEP environments. The raw point clouds are first preprocessed using voxel down-sampling with a grid size of 0.01 to reduce data redundancy while preserving key geometric features, and then refined through statistical outlier removal to eliminate noise and irrelevant points. This process yields sparse and low-noise point clouds suitable for subsequent segmentation and parameter extraction.

Table 1. Characteristics of the Trimble TX5 Scanner

| Laser Type | Parameters |
|---|---|
| Distance range | 0.6-120 m |
| Accuracy | ±2 mm@25 m |
| Angle range | Horizontal: 360°; vertical: 300° |
| Acquisition speed | Up to 976,000 pts/s |

The coordinate system of the as-designed model is used as the reference for the workflow. Walls and columns, assumed to be consistent with the design, are selected as registration primitives to align the as-built point clouds. A combined registration strategy using Fast Global Registration (FGR) for coarse alignment and Iterative Closest Point (ICP) for fine alignment transforms the as-built pipeline point clouds into the design coordinate system, providing a reliable basis for further analysis [18].

## 3.2 Point Cloud Semantic Segmentation

After obtaining the as-built pipeline point cloud in unified coordinate system, it is necessary to segment and identify different pipeline components for parametric modeling. PointNet++ [19] is a widely used semantic segmentation network that samples centroids uniformly and aggregates local features through set abstraction layers. However, in complex MEP environments with large scale variation, large-diameter pipes dominate the sampled points, while small components are under-sampled. This causes the loss of critical local geometry for small parts and reduces segmentation accuracy. To address this limitation, this study proposes a two-stage semantic segmentation method that integrates DBSCAN with deep learning to better capture pipeline components across scales.

### 3.2.1 Density-Based Pipeline Clustering

To reduce scene complexity and improve sampling balance, this study first applies the DBSCAN algorithm [20] to segment the global point cloud into independent pipeline instances. DBSCAN identifies clusters by locating points that satisfy a minimum density criterion, controlled by the neighborhood radius ($\epsilon$) and the minimum number of points. In MEP systems, pipelines are densely arranged but usually separated by small spatial gaps. By setting $\epsilon$ according to the minimum design clearance, DBSCAN clusters points from the same physical run into a single instance while discarding isolated noise. This separation allows each pipeline segment to be processed independently in the subsequent stage, avoiding feature interference from adjacent and unrelated structures.

### 3.2.2 PointNet++-Based Component Semantic Segmentation

Once individual pipeline instances are extracted via DBSCAN, the resulting segments are fed into PointNet++ model for fine-grained component recognition. This step classifies them into eight key categories: exhaust pipes, exhaust pipe elbows, exhaust pipe tees, air pipes, air pipe elbows, air pipe tees, water pipes and water pipe tees, covering the main elements in MEP layouts.

We construct a dedicated training set by collecting pipeline point clouds from multiple construction sites and manually composing them into diverse spatial layouts. Random translations, rotations and gaussian noise are applied to enhance robustness to sensor errors and scan-angle variations. The final dataset, summarized in Table 2, balances large pipes and small fittings.

Table 2. Dataset description

| Label | Category | Train Set | Test Set |
|---|---|---|---|
| 1 | Exhaust pipe | 158 | 40 |
| 2 | Exhaust pipe elbow | 120 | 32 |
| 3 | Exhaust pipe tee | 96 | 24 |
| 4 | Air pipe | 160 | 46 |
| 5 | Air pipe elbow | 138 | 40 |
| 6 | Air pipe tee | 96 | 24 |
| 7 | Water pipe | 180 | 48 |
| 8 | Water pipe tee | 122 | 32 |

In the segmentation stage, each isolated and normalized component is uniformly resampled to 8192 points, ensuring that even small parts retain sufficient point density. The network employs Multi-Scale Grouping (MSG) in its set abstraction layers to capture features across multiple radii, enabling it to learn both the geometry of straight pipes and the complex shapes of elbows and tees. Through feature propagation, local geometric cues are integrated with global context to assign a semantic label to every point. This instance-level strategy significantly improves overall performance and produces accurate component boundaries in dense MEP scenes.

## 3.3 Geometry Parameter Extraction

### 3.3.1 Preliminary extraction

For cylindrical pipe segments, RANSAC cylinder fitting algorithm is employed to robustly estimate the centerline and radius by minimizing the orthogonal distance between points and the cylinder surface. Once the axis direction is obtained, the pipe length is computed by projecting all points onto the axis and measuring the distance between the two extreme projections.

For rectangular pipes, RANSAC-based plane fitting algorithm is applied to extract the four main faces. The width and height are derived from the distances between opposing parallel planes, while the pipe orientation is inferred from the plane normal vectors.

For non-straight components such as elbows and tees, an indirect strategy is adopted. Insertion points and orientations are inferred from the intersections of centerlines of adjacent straight segments, ensuring topological consistency between fittings and pipes during reconstruction.

### 3.3.2 Correction Based on Logical Reasoning

However, geometric parameters extracted directly from point clouds are often affected by noise, occlusions and fitting errors, resulting in joint gaps. To address this, a logical reasoning scheme adjusts the as-built parameters based on engineering constraints and as-designed data. As shown in Figure 2, the reasoning rules are defined for three types of connections.

If two pipes are connected by a horizontal elbow, we first extend the central axis of the two pipes. Then we calculate the intersection of the two center axes, which can be thought of as the center coordinates of the elbow. Next, we correct the endpoint coordinates of the two pipes by analyzing the known radii in Figure 2(a). The final coordinate formulas are shown below, where $r$ represents the standard elbow radius from as-designed model, $x$ and $y$ represent the calculated horizontal elbow center coordinates, $a_5$, $a_6$, $b_1$ and $b_3$ represent the original endpoint coordinates.

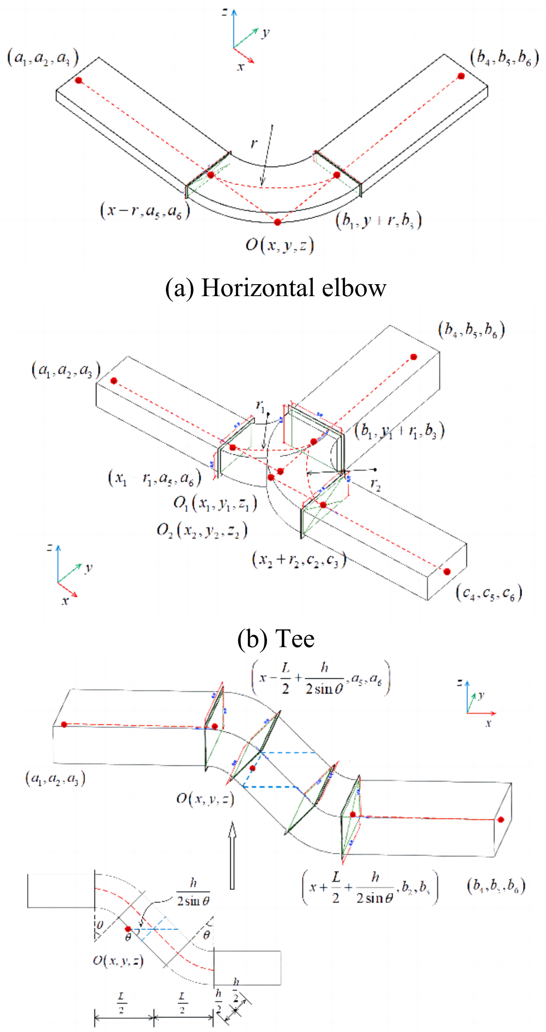
$$(x - r, a_5, a_6) \tag{1}$$
$$(b_1, y + r, b_3) \tag{2}$$

If two pipes are connected by a tee, we first find the three pipe endpoints around the tee, and distinguish the three pipes into two main pipes on a straight line, and a separate branch pipe. Then, we find the two intersection points of the central axis of the branch pipe and the central axis of each main pipe respectively. Next, we match the two radii with intersection and pipe, so the pipe endpoint coordinates are corrected by analyzing the known radii as shown in Figure 2(b). The inference formulas are as follows, $r_1$ and $r_2$ are two known radii of the tee in the as-designed model, $x_1$, $x_2$ and $y_1$ are the calculated center coordinates of the tee, $a_5$, $a_6$, $b_1$, $b_2$, $c_2$ and $c_3$ represent the endpoint coordinates.

$$(x_1 - r_1, a_5, a_6) \tag{3}$$
$$(b_1, y_1 + r_1, b_3) \tag{4}$$
$$(x_2 + r_2, c_2, c_3) \tag{5}$$

If two pipes are connected by upper and lower elbows, we first calculate the horizontal plane between the upper and lower pipes by deriving the central axis of the two pipes. Then we exploit the RANSAC algorithm to extract the cuboidal extent of the elbow horizontal section based on the horizontal plane. Next, we extract the line perpendicular to the tube axis at the edges of the horizontal plane and take their midpoints as standard points as Figure 2(c) shows. Afterwards, we correct the endpoint coordinates by the formulas. $L$, $h$ and $\theta$ represent the horizontal distance, height and curve angle of the upper and lower elbows respectively in the as-designed model, $x$ represents the calculated standard coordinate of elbow and $a_5$, $a_6$, $b_2$, $b_3$ represent the original coordinates.

$$\left(x - \frac{L}{2} + \frac{h}{2sin\theta}, a_5, a_6\right) \tag{6}$$
$$\left(x + \frac{L}{2} + \frac{h}{2sin\theta}, b_2, b_3\right) \tag{7}$$



(a) Horizontal elbow

(b) Tee

(c) Vertical elbow

Figure 2. Logical reasoning mathematical analysis

The logical reasoning module processes all detected connections in the pipeline network. For each connection node, the module first retrieves the corresponding connector type and its standard size from the as-designed BIM model, then applies the corresponding reasoning rules, and updates the endpoint coordinates of the pipeline accordingly. When the same pipeline involves multiple corrections, the corrections will be carried out sequentially, and the final coordinates are calculated according to the weighted average of the confidence of each connecting node. This iterative optimization process transmits geometric constraints to the entire network, effectively eliminating the cumulative error.

## 3.4 Graph Isomorphism and Modeling

### 3.4.1 Spatial Topology Analysis

To establish correspondence between as-built point cloud components and as-designed BIM elements, both datasets are represented as directed graphs. In each graph, vertices denote pipeline components, while directed edges encode physical connection from pipes to connectors based on spatial adjacency. The construction of the as-built graph is based on the segmented point cloud instances. The as-designed graph is generated from the BIM model by extracting component attributes and predefined connection relationships.

Graph matching is performed through a topology guided search strategy. First, candidate node pairs are initialized based on component type consistency and similarity of geometric attributes such as diameter, length and orientation. To reduce ambiguity in repetitive pipeline layouts, node degree and local connection patterns are further used as pruning constraints.

Starting from distinctive seed components, the matching process incrementally propagates along connected branches of the pipeline network. During traversal, each potential correspondence is verified by jointly evaluating geometric compatibility and topological consistency. Backtracking is applied when conflicts occur, ensuring a one-to-one mapping between the as-built graph and the design graph. The resulting correspondence enables automatic transfer of geometric parameters to the as-designed BIM model.

### 3.4.2 3D BIM Reconstruction

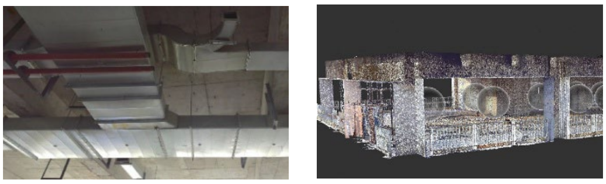
In the final stage, we use parametric programming in Dynamo to update the pipeline BIM model so it matches the as-built data. Dynamo scripts import the updated start and end coordinates, as well as the size and type of each pipe, elbow, and tee. Using the mapping between point cloud instances and as-designed BIM components, the system then automatically updates the corresponding parameters in the prefabricated family library.

# 4 Results and Discussion

To evaluate the feasibility of the proposed framework, we conducted a case study on standard floor A1# of a shopping center. The environment features a complex MEP layout with narrow spaces and densely distributed pipelines. The scene includes eight types of pipeline components, such as air pipes, exhaust pipes, water pipes and their corresponding elbows and tees. Its complexity is further increased by severe occlusion: 16 pipes are blocked by others, 5 pipes overlap, and 8 connectors are heavily covered. This challenging scenario provides a rigorous test platform for assessing the framework's capability to update the pipeline BIM model based on the as-built point cloud and the as-designed model.

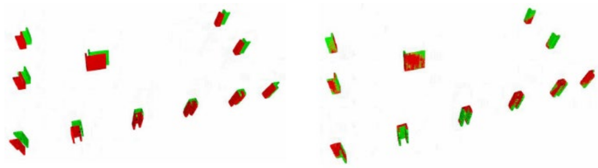
## 4.1 Point Cloud Collection and Processing

As shown in Figure 3, the acquired point cloud provides comprehensive coverage of the pipeline system despite severe occlusion. After registration, the point cloud achieved accurate alignment with the designed model. The RMSE decreased from 0.16m to $9.97\times10^{-3}$m, demonstrating effective spatial consistency. This precise alignment establishes a reliable basis for subsequent processes. The registration results are illustrated in Figure 4.



(a) Engineering practice (b) As-built point cloud

Figure 3. As-built pipelines



(a) FGR (b) ICP with FGR

Figure 4. Point cloud registration result

## 4.2 Point Cloud Segmentation

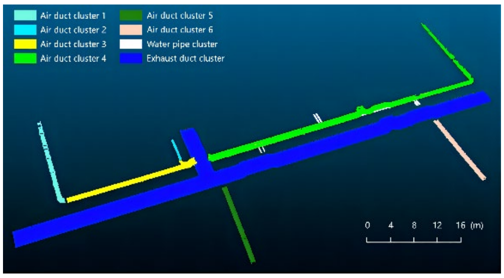
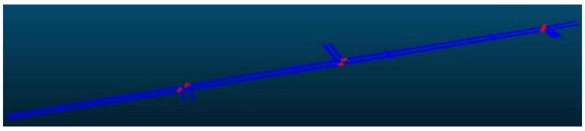
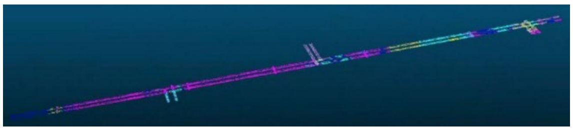



Figure 5. DBSCAN initial segmentation result

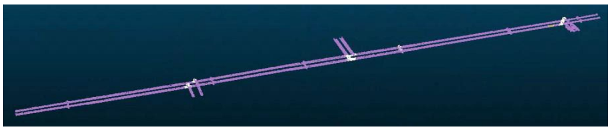
We integrated DBSCAN clustering with PointNet++ for semantic segmentation. With a radius of 0.025m and a minimum of 20 points, DBSCAN partitioned the point cloud into 8 clusters and removed 1,185 noise points (Figure 5). These clusters were subsequently processed by PointNet++, achieving an overall accuracy of 95.85%.



(a) Ground truth



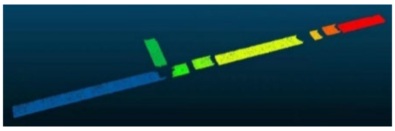
(b) PointNet++ without DBSCAN



(c) PointNet++ with DBSCAN

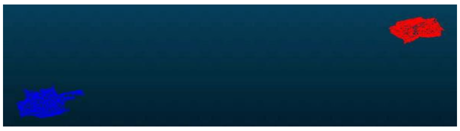
Figure 6. Water pipeline point cloud segmentation results

Figure 6 compares the water pipeline segmentation results in three cases: the actual scene (Figure 6(a)), PointNet++ alone (Figure 6(b)), and the combined DBSCAN-PointNet++ method (Figure 6c). Figure 7 further demonstrates successful instance segmentation of exhaust pipes, elbows and tees.



(a) Pipes

(b) Elbows



(c) Tees

Figure 7.Exhaust pipeline point cloud segmentation results

Table 3. PointNet++ segmentation results with DBSCAN and without DBSCAN

| Type | Without DBSCAN | | | With DBSCAN | | |
|---|---|---|---|---|---|---|
| | Precision (%) | Recall (%) | IoU (%) | Precision (%) | Recall (%) | IoU (%) |
| Exhaust pipe | 94.35 | 93.62 | 88.65 | 97.48 | 94.13 | 91.89 |
| Exhaust pipe elbow | 48.94 | 94.67 | 47.63 | 89.72 | 91.83 | 83.09 |
| Exhaust pipe tee | 71.58 | 82.89 | 62.36 | 70.80 | 73.93 | 56.66 |
| Air pipe | 94.69 | 95.20 | 90.37 | 98.90 | 98.92 | 97.84 |
| Air pipe elbow | 62.19 | 46.70 | 36.37 | 80.28 | 74.85 | 63.22 |
| Air pipe tee | 40.82 | 90.59 | 39.16 | 59.72 | 94.11 | 57.57 |
| Water pipe | 96.55 | 90.92 | 9.89 | 99.31 | 97.18 | 96.53 |
| Water tee | 52.31 | 82.90 | 11.54 | 53.85 | 83.85 | 48.79 |
| Average | 70.18 | 65.81 | 48.25 | 81.26 | 88.60 | 74.45 |

Using DBSCAN as a preprocessing step proves particularly effective in complex MEP scenes with significant scale variations, where PointNet++ alone often fails to capture data from small pipes. As shown in Table 3, the configuration with DBSCAN improves performance across all categories. For example, precision for air pipe elbows increases from 62.19% to 80.28% and recall for water tees rises from 82.90% to 83.85%. These results demonstrate the robustness of the proposed two-stage semantic segmentation strategy.

## 4.3 Geometry Parameter Extraction

Geometric parameters were first extracted via the RANSAC algorithm. However, initial extraction exhibited noticeable spatial errors. These were substantially reduced by applying the proposed logical reasoning module, which corrects pipe endpoints using known connector parameters from the as-designed model. Table 4 shows the resulting average coordinate errors.

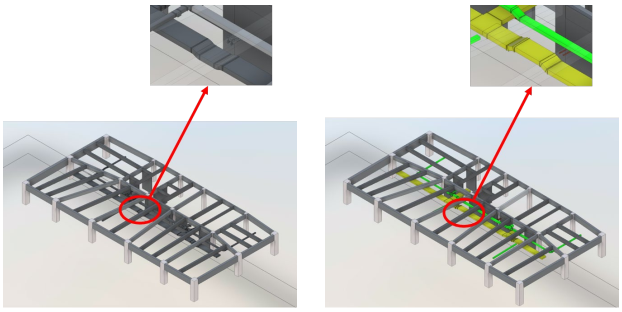
Table 4. Average Coordinate Error of Different Pipes

| Types | | Average spatial coordinate error | | |
|---|---|---|---|---|
| | | x (m) | y (m) | z (m) |
| Cuboidal exhaust pipes | Initial | 0.114 | 0.022 | 0.002 |
| | Corrected | 0.009 | 0.002 | 0.002 |
| Cuboidal air pipes | Initial | 0.027 | 0.026 | 0.003 |
| | Corrected | 0.010 | 0.011 | 0.003 |
| Cylindrical water pipes | Initial | 0.023 | 0.025 | 0.003 |
| | Corrected | 0.003 | 0.003 | 0.003 |

The results indicate that using RANSAC alone is often insufficient due to point cloud noise and occlusion, leading to misalignment between pipes and connectors. By mathematically inferring correct endpoints from connection relationships, the logical reasoning module effectively mitigates this issue. The result of errors generally below 0.01 m demonstrates the advantage of incorporating prior knowledge from the as-designed model over purely geometric methods, ensuring both complete and accurate connections.

### 4.4 Graph Isomorphism and Modeling

To automate the modeling process, spatial topology analysis was performed. Two directed graphs were constructed: one representing the logical structure of the as-built point cloud and the other corresponding to the as-designed BIM model. Graph isomorphism analysis established a one-to-one mapping between point cloud instances and BIM components. Based on this mapping, parametric programming was used to automatically update the geometric parameters. Figure 8 visually compares the as-designed BIM model (Figure 8(a)) with the updated BIM model (Figure 8(b)), demonstrating the success of reconstruction.



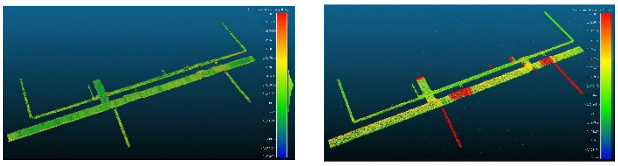
(a) As-designed (b) Updated

Figure 8. Pipeline BIM model

This step bridges the gap between geometric data and semantic BIM objects. By representing the pipeline network as a directed graph, the framework resolves the correspondence problem, enabling precise association between each BIM element and its corresponding extracted parameters. This automated mapping streamlines the transition from raw data to an information-rich BIM model, which is essential for the framework's practical application.

### 4.5 Precision and Efficiency Analysis

Our framework was evaluated in terms of both modeling accuracy and time efficiency. Deviation analysis between the updated BIM model and the laser-scan point cloud is presented in Figure 9(a). Errors followed a normal distribution, with over 90% of points exhibiting deviations under 0.03 m and an average deviation of only 0.0046 m. Figure 9(b) further compares the output with the original as-designed model, clearly indicating areas where on-site modifications were made. A significant gain in efficiency was achieved. Whereas traditional manual modeling for a scene of this complexity typically takes 3-5 days, our framework completed the update in only 3 hours, corresponding to a time reduction of nearly 90%. This demonstrates that the proposed framework offers an accurate and efficient solution for updating as-designed pipeline BIM models.



(a) Deviation distribution (b) Designed vs updated

Figure 9. Model comparison

## 5 Conclusion and Future Work

This study proposes and validates an automated framework for bridging as-designed BIM models with as-built point cloud data. By integrating two-stage semantic segmentation, parameter extraction with logical reasoning and graph isomorphism analysis, the framework addresses key challenges in complex MEP environments, including severe occlusion, multi-scale components and automated semantic association. Validation using a real-world shopping center case demonstrates that the proposed method can robustly update high-fidelity BIM models that accurately capture on-site conditions.

The results indicate that the hybrid segmentation strategy substantially enhances the detection of small-scale pipelines in cluttered scenes, effectively mitigating the loss of detail associated with conventional approaches. Moreover, by incorporating prior design knowledge through logical reasoning instead of relying solely on geometric fitting, the updated models preserve logical connectivity and topological correctness even when the raw data are imperfect. In addition, graph isomorphism analysis enables automated mapping between point-cloud instances and BIM elements, streamlining the workflow and reducing the time and labor required for manual modeling, while still meeting the accuracy requirements of facility management.

It is important to emphasize that in this study, the term automatic mainly refers to the automated updating of pipeline BIM components after data input and parameter initialization. Although several preprocessing parameters are defined according to engineering experience, the framework eliminates manual modeling operations and enables efficient generation of an updated BIM model.

Despite these contributions, several limitations warrant further research. First, the framework targets standard pipeline geometries, future work should expand the component library to cover more complex MEP elements, such as valves and pumps. Second, while terrestrial laser scanning is highly accurate, adding complementary data sources such as photogrammetry or handheld SLAM could improve completeness in narrow or occluded areas. Finally, extending the algorithms toward processing in near real time would support quality control during construction by enabling immediate detection and correction of deviations.

## References


[1] Y. Chen and X. Luo. Indoor scan-to-BIM workflows: Progress, challenges, and future directions (2014–2024). *Automation in Construction*, 181:106578, 2026.

[2] E. Valero, F. Bosché, and M. Bueno. Laser scanning for BIM. *Journal of Information Technology in Construction (ITcon)*, 27(23):486–495, 2022.

[3] S. Zbirovský and V. Nežerka. Open-source automatic pipeline for efficient conversion of large-scale point clouds to IFC format. *Automation in Construction*, 177:106303, 2025.

[4] B. Wang, C. Yin, H. Luo, J. C. P. Cheng, and Q. Wang. Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data. *Automation in Construction*, 125:103615, 2021.

[5] X. Li, V. J. L. Gan, K. Li, and M. Li. High-precision 3D BIM reconstruction for mechanical, electrical and plumbing components using terrestrial laser scanning and LiDAR point clouds. *Journal of Building Engineering*, 113661, 2025.

[6] J. Patil and M. Kalantari. Automatic Scan-to-BIM—The Impact of Semantic Segmentation Accuracy. *Buildings*, 15(7):1126, 2025.

[7] S. Jing, G. Cha, M. B. Maru, B. Yu, and S. Park. Improved building MEP systems semantic segmentation in point clouds using a novel multi-class dataset and local--global vector transformer network. *Journal of Building Engineering*, 96:110311, 2024.

[8] R. Zhang, Y. Wu, W. Jin, and X. Meng. Deep-learning-based point cloud semantic segmentation: A survey. *Electronics*, 12(17):3642, 2023.

[9] M. Mérizette, N. Audebert, P. Kervella, and J. Verdun. 3DSES: an indoor LiDAR point cloud segmentation dataset with real and pseudo-labels from a 3D model. *arXiv preprint arXiv*:2501.17534, 2025.

[10] H. Yue, Q. Wang, H. Huang, X. Xia, H. Fang, and J. C. P. Cheng. Enhancing semantic segmentation of MEP scenes with deep learning and BIM-generated synthetic point clouds. *Advanced Engineering Informatics*, 68:103723, 2025.

[11] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for point-cloud shape detection. *In Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[12] H. Yue, Q. Wang, Y. Yan, and G. Huang. Deep learning-based point cloud completion for MEP components. *Automation in Construction*, 175:106218, 2025.

[13] L. Centarti, N. E. Güidi, and B. Messina. Automated Scan-To-BIM Methodology for Accurate 3D Modeling of Embedded MEP Systems. In *Representation Across Boundaries: New Links with AI, AI-GEN, and XR Tools for Cultural Heritage and Innovative Design*, pages 905–924. Springer, Cham, 2026.

[14] V. S. Kalasapudi, Y. Turkan, and 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*, volume 2, pages 145–152. IEEE, 2014.

[15] F. Bosché, M. Ahmed, Y. Turkan, C. T. Haas, and 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*, 49:201–213, 2015.

[16] B. Wang, Q. Wang, J. C. P. Cheng, C. Song, and C. Yin. Vision-assisted BIM reconstruction from 3D LiDAR point clouds for MEP scenes. *Automation in Construction*, 133:103997, 2022.

[17] B. Wang, Z. Chen, M. Li, Q. Wang, C. Yin, and J. C. P. Cheng. Omni-Scan2BIM: A ready-to-use Scan2BIM approach based on vision foundation models for MEP scenes. *Automation in Construction*, 162:105384, 2024.

[18] Y. Zha, M. Chen, and S. Lu. Research on point cloud registration of industrial parts based on FGR-ICP algorithm. In *Journal of Physics: Conference Series*, vol. 1941, no. 1, p. 012014, 2021.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017.

[20] E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.