www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

A PARAMETRIC APPROACH TOWARDS SEMI-AUTOMATED 3D AS-BUILT MODELING

SUBMITTED: March 2023
REVISED: August 2023
PUBLISHED: December 2023
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2023.041

Yu-Chen Lee, Ph.D Student, corresponding author
Dept. of Civil, Architectural and Environmental Engineering, The University of Texas at Austin
ORCID: 0000-0001-7479-9968
email: yuchenlee@utexas.edu

Jong Won Ma, Ph.D., Assistant Professor
Dept. of Building, Civil and Environmental Engineering, Concordia University
ORCID: 0000-0001-5289-9183
email: jongwon.ma@concordia.ca

Fernanda Leite, Ph.D., P.E., F.ASCE, Professor and John A. Focht Centennial Teaching Fellow in Civil Engineering, Construction Engineering and Project Management Program
Dept. of Civil, Architectural and Environmental Engineering, The University of Texas at Austin
ORCID: 0000-0002-7789-4474
email: fernanda.leite@utexas.edu

SUMMARY:*Building Information Modeling (BIM) has been developed in response to the growing complexity of construction projects. BIM implementation is beneficial throughout the entire building life cycle, and thus has been widely adopted in new projects. However, BIM implementation in existing buildings is impeded by the lack of as-built models. Conventionally, three-dimensional (3D) as-built BIMs are generated by experienced modelers, which is time-consuming and error-prone. To cater to the need, Scan-to-BIM is a solution for automation in as-built BIM generation. In the context of automated Scan-to-BIM, the parametric modeling process is worth investigating as it has the ability to not only reconstruct 3D objects from a variety of categories from point clouds but also offer flexibility to update objects simply by changing the values of the correlative parameters. Hence, this study proposes a semi-automated framework for assisting 3D as-built modeling through a parametric modeling approach. The presented methodology starts with wall boundary parameter extraction through 3D to 2D projection of the wall segments and line detection techniques, followed by retrieving geometric parameters of all other non-wall elements via CloudCompare. The extracted parameters are structured into a Microsoft® Excel® file and fed into Autodesk® Dynamo for 3D BIM creation using a series of designed logic. To substantiate the viability, the proposed framework is employed in two datasets containing structural, architectural, furniture, mechanical, and plumbing objects (further categorized into structural, hosted and non-hosted elements). The Intersection over Union (IoU) of structural elements was 96.35%, while the root-mean-square error (RMSE) of hosted and non-hosted elements was 3.634 and 2.607 mm, respectively. This study established a universal methodology for semi-automated 3D as-built modeling that can guide future research.*

KEYWORDS: *Scan-to-BIM, Parametric modeling, As-built BIM, Autodesk® Dynamo.*

REFERENCE: *Yu-Chen Lee, Jong Won Ma, Fernanda Leite (2023). A parametric approach towards semi-automated 3D as-built modeling. Journal of Information Technology in Construction (ITcon), Vol. 28, pg. 806-825, DOI: 10.36680/j.itcon.2023.041*

1. INTRODUCTION

The construction industry accounts for a significant portion of the U.S. economy (U.S. Bureau of Labor Statistics, 2021). According to the U.S. Bureau of Labor Statistics, the construction sector contributed 5.9 percent of all nonfarm employment and 4.3 percent of Gross domestic product (GDP) in the U.S. in 2020. In spite of its crucial economic role, the construction industry has a poor reputation for low productivity (Davila Delgado et al., 2019). Given the complexity and uncertainty of projects, the industry has been plagued by construction waste, schedule delays, quality assurance, increasing costs, and information transfer between multiple disciplines (Ullah et al., 2019). For the sake of tackling obstacles in project management, Building Information Modeling (BIM) is one of the most important advancements and has been widely adopted in the architecture, engineering, and construction (AEC) industry in recent decades (Heaton et al., 2019). Adopting BIM in construction projects can assist decision-makers in simulating and evaluating potential influences in a virtual environment (Gruber et al., 2018), facilitating problem-solving with constrained labor and project costs. For instance, BIM can support clash detection, cost estimation, design visualization, and effective cooperation to improve project performance (Leite, 2019; Volk et al., 2014). BIM can be further integrated with other technologies, such as geographic information systems, extended reality, and laser scanning, for a broader range of applications (Alizadehsalehi et al., 2020; Bansal, 2021; Bosché et al., 2015).

In addition to implementing BIM in new construction projects, BIM adoption in existing buildings can better facilitate operation and maintenance by reflecting changes made during the construction phase. As-built BIMs can be used in non-invasive analysis to evaluate structural deformations in historic buildings preservation (Moyano et al., 2022). Although a significant development in BIM applications has been made, the issue of automation in as-built modeling is far from being settled. 2D paper-based drawings might not be well-preserved for existing structures or heritage buildings, not to mention 3D BIMs. Therefore, creating as-built BIMs is the very first step of BIM implementation in existing building structures. The traditional method of using 2D drawings for as-built BIM creation has the possibility of causing errors by not capturing and updating variations during construction. Scan-to-BIM can transform scanned point clouds with precise information into 3D BIMs by integrating 3D laser-scanning and BIM. Some of the issues faced with Scan-to-BIM include dealing with occluded or hidden objects in the scanned data, which can pose difficulties in accurately modeling these areas. However, despite these challenges, the Scan-to-BIM technique offers advantages such as high efficiency and less human intervention outweigh its limitations. It is noteworthy that currently developed software and tools for transforming point clouds into as-built BIMs are still missing the capability to accurately model a variety of objects in real-world environments without the necessity of manual intervention (Jung et al., 2018). Accordingly, the importance of investigating an automatic Scan-to-BIM solution for efficient and rigorous 3D as-built BIM generation is self-evident.

In light of automating Scan-to-BIM, object-based parametric modeling can create as-built BIMs with fully defined parametric objects. Parametric modeling refers to automatically generating digital models for representing elements through predefined parameters. Parametric modeling employs an objected-oriented approach in which every object in the built environment is associated with its inherent attributes that should be identified and defined. Parametric modeling is agile and flexible, where objects can be updated interactively by modifying the corresponding value of predefined parameters (Barazzetti, 2016; Ma et al., 2022). Nevertheless, detailed logics or algorithms for parametric modeling are not thoroughly investigated for modeling real-world objects. Accordingly, this research seeks to establish a general framework leveraging Autodesk® Dynamo, a commercially available BIM authoring software, Autodesk® Revit®. The aim is to facilitate the creation of as-built BIMs for indoor objects utilizing a parametric approach that can be applied across a variety of contexts. The authors enrich the Scan-to-BIM knowledge base both intellectually and practically by: 1) pinpointing essential parameters for modeling prevalent indoor objects in the built environment, 2) presenting a well-structured workflow in Autodesk® Dynamo for diverse object types, which could serve as a guideline for future research, 3) examining the feasibility of the suggested methodology by implementing the proposed framework to model two distinct types of real-world buildings, and 4) reducing the need for human intervention in as-built BIM generation.

2. BACKGROUND RESEARCH

2.1 Scan-to-BIM

Scan-to-BIM is a process that generates as-built BIMs from scanned point clouds, typically comprised of three stages: scanning, registration, and modeling (Esfahani et al., 2021). This approach is widely applied to reconstruct

as-built information for managing or monitoring existing buildings and heritage structures, particularly when 2D drawings or 3D models are absent (Allegra et al., 2020; Brumana et al., 2020; Pepe et al., 2020). With the advancement of laser scanning technologies, the scanned point clouds with enriched geometric information enable accurate and rapid data collection (Chen et al., 2018). However, traditional Scan-to-BIM practices often involve significant manual efforts, which are labor-intensive, subjective, and prone to error, requiring experiential professionals to perform the tasks (Tang et al., 2010).

In recent years, several research efforts have been placed on the automation of Scan-to-BIM. For instance, Macher et al. (2021) presented a semi-automated approach for detecting windows from point clouds of building façades. The proposed method combined geometric and radiometric information, utilizing intensity histogram analysis to extract peaks representing different objects. In another study, Chen et al. (2018) developed a template-based regional proposal mechanism leveraging the voxel grid method to identify major building components, including beams, columns, roofs, stairs, and window frames. While the method conveniently omits the segmentation process, enabling it to work on unstructured point clouds, it does necessitate user intervention and has a relatively low precision rate. Son et al. (2015) provided an in-depth evaluation of three existing Scan-to-BIM software in terms of automation level and performance in detecting pipes, including Trimble RealWorks®, Leica Cyclone, and ClearEdge3D EdgeWise^{3D}. The authors concluded that all software examined in the study leverages semi-automated approaches, given their high reliance on user inputs for tasks such as selecting point clouds to be modeled or defining necessary parameters.

Further contributing to reducing human input in Scan-to-BIM, Bosché et al. (2015) proposed a method that specifically targets pipes, conduits, and ducts. The circular cross-section detection approach was incorporated with the Scan-vs-BIM technique to identify and match objects from the as-built point clouds with the designed 3D BIM of the project. Then, the as-built model was generated by the best-fitting centerline and radius from cross-sections to create straight cylinders objects. While the proposed method is promising for automating the modeling process, it carries a limitation of requiring readily available as-planned BIMs.

Though previous research shows significant progress in Scan-to-BIM automation, a certain extent of manual intervention may not be avoided, especially in creating as-built models based on extracted geometric information. Consequently, exploring automation in the modeling stage is a necessary step toward achieving a fully automated Scan-to-BIM process.

2.2 Parametric modeling approach for Scan-to-BIM

Data collection and object recognition in Scan-to-BIM have attracted considerable attention recently. However, studies focusing on the modeling stage remain limited, as summarized in Table 1. The workflow presented by Andriasyan et al. (2020) utilizing a combination of Rhinoceros and Grasshopper-ArchiCAD illustrated the potential of parametric modeling in reconstructing complex heritage buildings from point clouds. Similarly, a recent study by Barazzetti (2016) applied an innovative semi-automated method. The proposed approach employed Non-Uniform Rational B-Splines (NURBS) surfaces and curves to generate complex and irregular components, further emphasizing the potential of parametric modeling in this sphere.

Yang et al. (2020) put forth a semi-automated methodology for generating steel structural components from point cloud data through CloudCompare, MATLAB, and Autodesk® Dynamo. In this study, CloudCompare was used for point cloud segmentation and several algorithms were deployed in MATLAB for extracting geometric information. The proposed workflow incorporated Autodesk® Dynamo for as-built model generation, demonstrating its capability to efficiently create parametric models. Nevertheless, the research was limited to four common structural components, indicating the need for further development to enhance its applicability. In a separate study, Wang et al. (2021) introduced a method to reconstruct mechanical, electrical, and plumbing (MEP) components, which involves a 2D to 3D analysis for object detection and geometric information extraction. This method underscored the potential of using Dynamo for parametric modeling. However, the framework had a limitation as it required all MEP elements to be included in the model library, limiting its effectiveness with unseen items.

Numerous studies have highlighted the potential of Dynamo for creating parametric BIMs. However, several common challenges persist in the literature. For example, the lack of detail in explaining the procedure of designing the Dynamo workflow hinders future research, strengthening the need for detailed guidelines to facilitate further development of parametric modeling leveraging Dynamo. In addition, most previous research endeavors have focused on individual areas of interest, such as structural or MEP systems, illustrating the need for a more

universally applicable method. Inspired by preceding studies, this research aims to address the aforementioned issues to further enhance the modeling stage in Scan-to-BIM. Thus, the authors outline a comprehensive framework for 3D as-built modeling through a semi-automated parametric approach leveraging Autodesk® Dynamo and investigate its viability using two real-world datasets consisting of structural, architectural, furniture, and MEP objects.

Table 1: Overview of Previous Studies on Parametric Modeling Approaches.

References	Focused Objects	Method/Tool	Automation Level	Accuracy	Limitations
Andriasyan et al., (2020)	Heritage building components	Rhinoceros and Grasshopper-ArchiCAD	Semi-automated	Standard deviation: 68.28 pixels	• Lower 3D meshing accuracy in volumetric components. • The algorithm can only represent the overall shape, volume, and deformations of the building components.
Barazzetti (2016)	Complex and irregular components	NURBS surfaces and curves	Semi-automated	Error: ±2 to ±4 mm	• Requires creating custom Revit family objects for modeling.
Yang et al. (2020)	Steel structural components	CloudCompare, MATLAB, and Autodesk® Dynamo	Semi-automated	RMSE: 3.8 to 11.8 mm	• Limited to four common structural components.
Wang et al. (2021)	MEP components	2D to 3D analysis and Autodesk® Dynamo	Automated	Retrieval rate: 91.3% RMSE: 2.45 mm	• Requires all MEP elements to be included in the model library. • Limited to MEP components that are only aligned with the orientation of the building.

3. RESEARCH METHODOLOGY

3.1 Overall Methodology

Building upon on our previous study (Ma et al., 2022), this research leverages parametric modeling to generate 3D as-built models from scanned point clouds for a variety of indoor objects in the built environment. This study assumes that all objects have been properly separated from raw point clouds, as point cloud segmentation is beyond the scope of this research. Moreover, the proposed method is assuming the pre-segmented point clouds are nearly flawless and accurate, as errors or inconsistencies in the segmentation could influence the accuracy of the extracted geometric information. The authors acknowledge that this assumption is ideal and is not always feasible in practice. Therefore, the proposed framework can still function to handle less-than-perfect conditions, with the understanding that modeling accuracy might be compromised.

As shown in Figure 1, the proposed research methodology consists of two stages: pre-processing and parametric modeling.

1. Pre-processing stage
   The authors projected wall segments to 2D images and conducted line detection to obtain boundary information. Python scripts and measurement tools in CloudCompare are utilized to extract parametric information of all other non-wall objects. The parameter sets for each object category are then archived in a Microsoft® Excel® file.

2. Parametric modeling stage
   This stage first starts with importing the Microsoft® Excel® file into Autodesk® Dynamo, which is subsequently followed by the creation of wall, floor, and roof elements to outline the skeleton of the structure. Then, hosted elements such as doors and windows can be generated by specifying their host walls. Hosted elements are elements that require attachment to a host wall. Simultaneously, non-hosted

elements including tables, chairs, and other types of furniture are created. These non-hosted elements are characterized as movable and rotatable objects that can be placed anywhere without needing attachment to host walls.

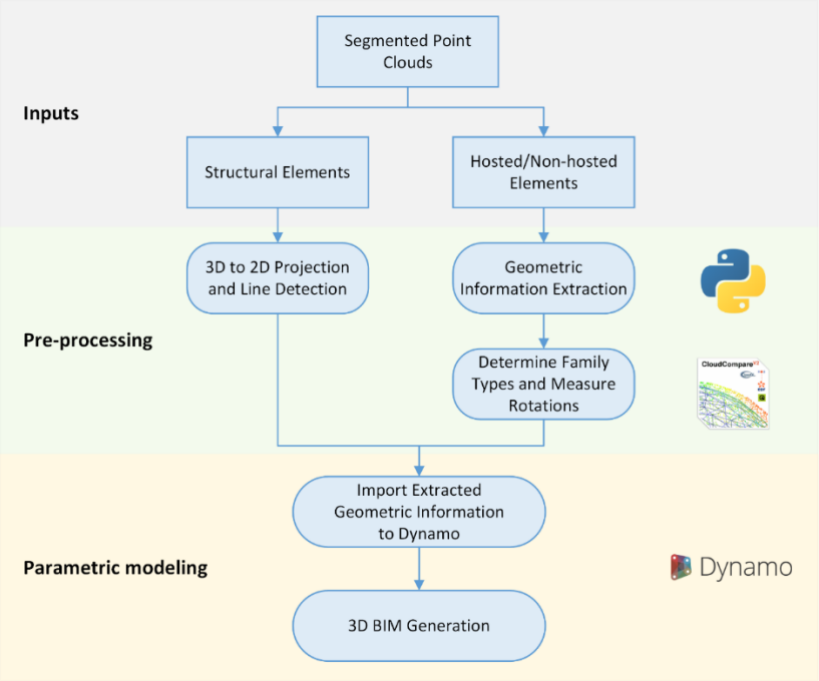

Figure 1: Workflow of the proposed framework.

3.2 Dataset

This research utilizes two benchmark datasets for case studies. The first case is the Stanford 3D Indoor Scene Dataset (S3DIS), which comprises point clouds of 271 rooms in 6 indoor areas at Stanford University (Armeni et al., 2016). Each point in the S3DIS is annotated and objects are categorized into 13 groups, such as column, wall, floor, bookcase, table, and so on. The S3DIS stores segmented point clouds as .txt files, which are named after the object type they represent (e.g. bookcase_1.txt or window_1.txt). For the first case, area five from the S3DIS was selected for applying the proposed framework as it encompasses the largest number of rooms (68 in total) and diverse environments.

The Medical Center Mechanical Room dataset is chosen for the second implementation case. The dataset consists of 3D BIMs featuring MEP system elements like pipes and elbows. However, this dataset does not include pre-segmented point clouds. As this research focuses on enhancing the modeling stage of Scan-to-BIM, the segmentation process falls outside the scope of this study. Instead, the authors employ the dataset's as-designed BIMs to extract necessary information using Autodesk® Dynamo under the assumption that the geometric information accurately represents the actual objects, mimicking the conditions of well-segmented point clouds. This methodological choice allows us to focus on evaluating the potential and constraints of the proposed parametric modeling approach in a controlled setting. Except for these noted differences, the second case followed the same workflow to generate as-built models as the first case.

3.3 Geometric Information Requirement

When it comes to as-built modeling, the modeling parameters differ according to the type of object. For example, when modeling wall elements, necessary parameters include the coordinates of the start and end points that are representing wall centerlines (Figure 2), along with the wall's starting level and height. The generation of floors and roofs requires coordinates of multiple points to form enclosed curves as well as the start and end levels for each curve. In the case of wall-dependent objects, such as doors, windows, bookcases, and boards, modeling requires the start and end coordinates of their host walls. The calculation of distances from the centers of these

wall-based objects to the start point coordinates of their respective host walls is also necessary. In addition, the specific family types and dimensions of those objects are needed for accurate sizing. For other objects that are not attached to walls, the parameters include the center's location, family types, and object dimensions. Table 2 summarizes the information necessary for each object type, providing a detailed overview.

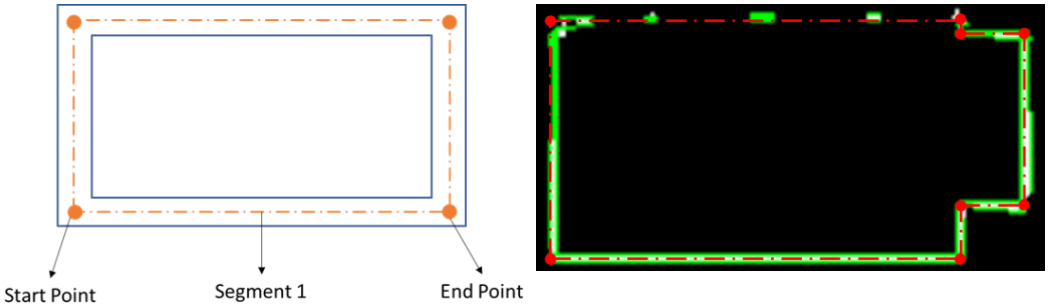

Figure 2: Illustration of wall centerline.

Table 2: Required information for different object types.

Object Type	Family Type	Required information
Structural elements	Walls	• Location coordinates of the start point and end points representing wall centerlines • Starting level and height • Wall type
	Floors and ceilings	• Location coordinates of points forming closed curves • Floor or ceiling type
Hosted elements	Doors, windows, bookcases, and whiteboards	• Location coordinates of host walls • Length from the center of hosted objects to the start point of its host wall • Family type and dimensions
Non-hosted elements	Tables and chairs	• Location coordinates of the object's center • Family type and dimensions
Mechanical elements	Mechanical equipment	• Location coordinates of the object's center • Family type and dimensions
Plumbing elements	Pipes and flex ducts	• Location coordinates of the start and end points representing pipe centerlines • Level • Family type and diameter
	Pipe fittings	• Location coordinates of the object's center • Family type and diameter

3.4 Parametric Information Extraction – Python scripts, CloudCompare, and Microsoft® Excel®

In the pre-processing phase, the authors utilized scripts to project 3D point clouds onto 2D images, then used the Hough transform (Hough, 1962) to extract lines representing walls, as illustrated in Figure 3. The Hough transform is a method for extracting shape characteristics from an image via a voting procedure. Fig. 3b shows the outcome of the line detection algorithm, where green lines represent detected lines. The wall centerline information is derived by calculating the median line between two lines and their intersections. The coordinates of start and end points for each centerline segment are then recorded to a Microsoft® Excel® spreadsheet on a sheet named "Wall."

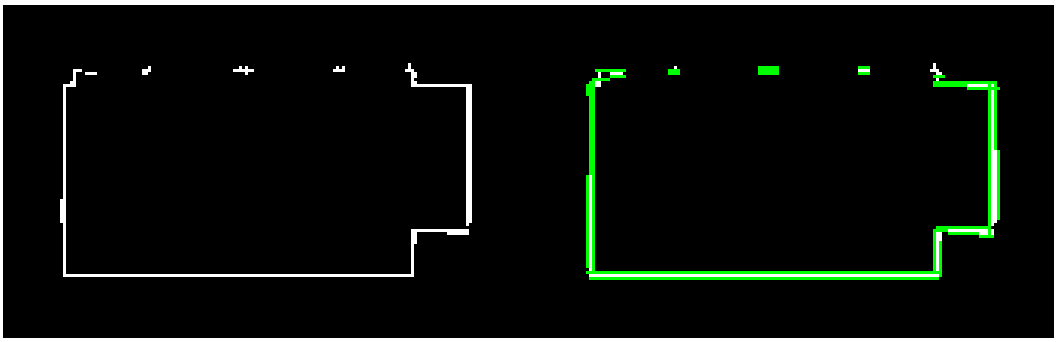

Figure 3: Examples of (a) projected point cloud image and (b) line detection image.

Upon acquiring the wall boundary information, the authors developed Python to automatically extract geometric information for all other non-wall objects within the indoor space. These scripts search through all folders in the parent folder, looking for an "Annotations" folder where point cloud files are stored. The proposed methodology presumes that object physical sizes can be approximated by bounding boxes, which enables an efficient representation and subsequent analysis of data (as illustrated in Figure 4). For hosted elements, the center location, family type, and dimensions (length, width, and height) of the bounding box were determined. The scripts identified the nearest wall for these objects by computing distances to previously outlined walls. For non-hosted objects, the algorithm determined the center location, as wall information was not required. The information was recorded in the spreadsheet containing the wall centerline data, with separate sheets created for different object types. Sheets in the output file use object type as sheet name, such as “Door,” “Bookcase,” and “Window.” Each sheet contains object-specific information, as mentioned in Table 2. Furthermore, the processing time for each room’s point clouds was also documented, as shown in Figure 5. Table 3 shows the number of objects processed and the time taken for different room types using the script. The total time spent processing all rooms in area five of the S3DIS is 1426.8 seconds for a total of 2338 objects, equating to approximately 0.6 seconds per object. All computations were performed on a system equipped with an Intel® Core™ i7-8700 CPU, 64 GB RAM, operating on a 64-bit system.

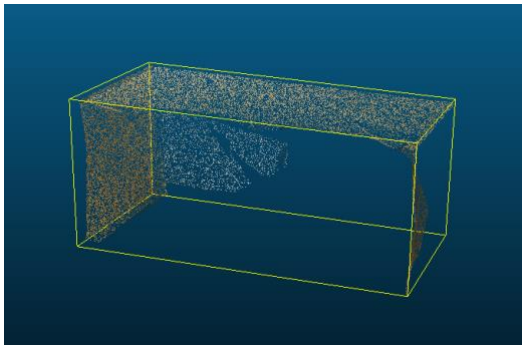

Figure 4: Object represented by a bounding box with dimensions approximating its size.

Table 3: Summary of total processing time and number of objects by room type.

Room Type	Total Time (seconds)	Number of Objects
Conference Room	65.68	131
Hallway	789.39	292
Lobby	2.11	45
Office	526.34	1708
Pantry	0.05	19
Storage	38.84	95
WC	4.39	48
Total	1426.8	2338

```python
# Process each folder
for folder in folder_list:
    annotations_folder = os.path.join(folder, 'Annotations')

    # Print processing start message
    print(f"Processing {os.path.basename(folder)} folder ...")

    # Start the timer
    start_time = time.time()

    process_folder(annotations_folder)

    # Calculate and print processing time
    processing_time = time.time() - start_time
    print(f"Time spent processing {os.path.basename(folder)}: {processing_time:.2f} seconds")
    print()
```

```
Processing conferenceRoom_1 folder ...
File saved for folder: conferenceRoom_1
Time spent processing conferenceRoom_1: 10.75 seconds

Processing conferenceRoom_2 folder ...
File saved for folder: conferenceRoom_2
Time spent processing conferenceRoom_2: 38.60 seconds

Processing conferenceRoom_3 folder ...
File saved for folder: conferenceRoom_3
Time spent processing conferenceRoom_3: 15.93 seconds

Processing hallway_1 folder ...
File saved for folder: hallway_1
Time spent processing hallway_1: 29.43 seconds

Processing hallway_10 folder ...
File saved for folder: hallway_10
Time spent processing hallway_10: 40.19 seconds
```

Figure 5: Example of the script in action, documenting the time spent for processing each room.

CloudCompare serves as the tool to gather information relating to Revit family types. The authors utilized standard Revit family objects, which are universally accessible to users. This strategy ensures that the proposed approach can be reproduced by users who may not have access to custom objects. Using built-in family items could also streamline the process, as custom objects may require additional information or specific import procedures. However, it should be noted that built-in family types are not exhaustive and may not be a perfect fit for every real-world object. Therefore, the authors have to perform their judgment to select the most appropriate match. In this study, point cloud files are imported into CloudCompare, and researchers determine the default family type most similar in shape to the point clouds to represent each object. Also, the authors utilize measurement tools in CloudCompare to gather rotation angle information. The manually gathered information using CloudCompare consists of identifying the family type that best matches the shape of the objects and the rotation angle information. The chosen family type and orientation details are then appended to the corresponding spreadsheet, equipping it with crucial information for the 3D modeling process.

	G	H	I	J	K	L
1	StartPoint.X	StartPoint.Y	StartPoint.Z	EndPoint.X	EndPoint.Y	EndPoint.Z
2	-9.491	16.873	0	-4.883	16.873	0
3	-4.883	16.873	0	-4.883	19.943	0
4	-4.883	19.943	0	-9.491	19.943	0
5	-9.491	19.943	0	-9.491	16.873	0
6						
7						
8						
9						
10						

Wall | Chair | Board | Door | Table | Window | Bookcase

Figure 6: An example of the output of geometric information extraction.

The methodology of the study is designed specifically for efficient implementation in Revit using the Dynamo plugin. While other BIM tools, such as Grasshopper for Rhino, can perform similar tasks, their compatibility and

functionality may differ based on project requirements. Thus, the optimization is currently focused on using Revit and Dynamo. In sum, this research extracts essential geometric information using a combination of Python scripts and CloudCompare, compiling it into a Microsoft® Excel® spreadsheet containing several worksheets for different element types, as demonstrated in Figure 6.

3.5 Parametric Modeling – Autodesk® Dynamo

The framework proposed in this research selects Autodesk® Dynamo for the parametric modeling process. Dynamo is an add-on for Autodesk® Revit® and a visual programming language that provides high flexibility for users to design and execute sophisticated workflows. Utilizing customized nodes along with built-in nodes, users can design algorithms or automate tasks to assist in 3D BIM generation simply by connecting and sequencing nodes. Dynamo is especially appropriate for modeling objects with complex design features, as it can help reduce human error in geometric modeling (Kontoudaki and Georgopoulos, 2022). Furthermore, once a generic workflow is established within Dynamo, it can be efficiently applied to similar structures. Given the aforementioned advantages, Dynamo is selected as a facilitator for efficient parametric modeling.

4. PARAMETRIC MODELING PROCESS – CASE 1

As mentioned previously, area five of S3DIS consists of indoor rooms with standard office objects (including walls, floors, ceilings, doors, windows, bookcases, and boards). Figure 7 illustrates the configuration of the rooms in area five of S3DIS. The parametric modeling process of this dataset begins with importing data from the Microsoft® Excel® file, followed by creating structural elements that define the room outlines. Subsequently, hosted and non-hosted objects can be created simultaneously. For each object type, the design logic and the required information are different. Therefore, the following section dives deeper into the parametric modeling process for the three primary object categories: (1) structural elements, (2) hosted elements, and (3) non-hosted elements.

The designed logic contains built-in nodes and nodes from 'Clockwork for Dynamo' (Dieckmann, 2013) and 'Springs for Dynamo' (Venkov, 2015) packages, which are open access resources published by Dynamo users. To facilitate the parametric modeling process and serve the needs, the authors designed a customized node' GetPointsFromExcel(xyz)' that allows users to obtain x, y, and z coordinates of points from a Microsoft® Excel® file by specifying the file path, name of the worksheet, and the index of columns.

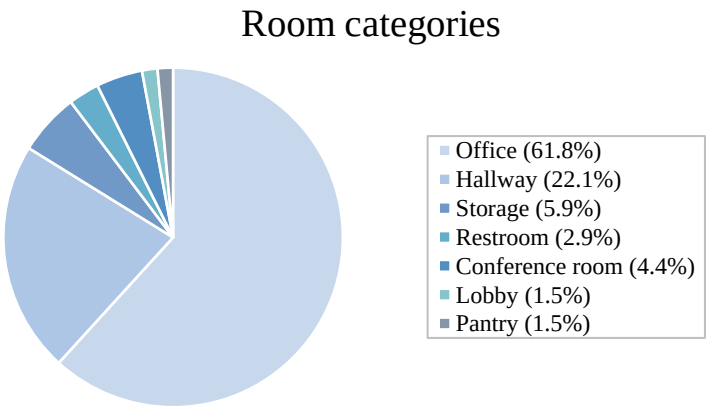

Figure 7: Configuration of rooms in area five of S3DIS.

4.1 Structural elements

Structural elements encompass walls, floors, and ceilings. For walls, the process involves forming centerlines by retrieving start and end point data. Floors and ceilings employ a similar strategy, using point data to establish the outlines and then translating these outlines vertically to represent the ceiling. Figure 8 shows an example of the designed logic for wall objects.

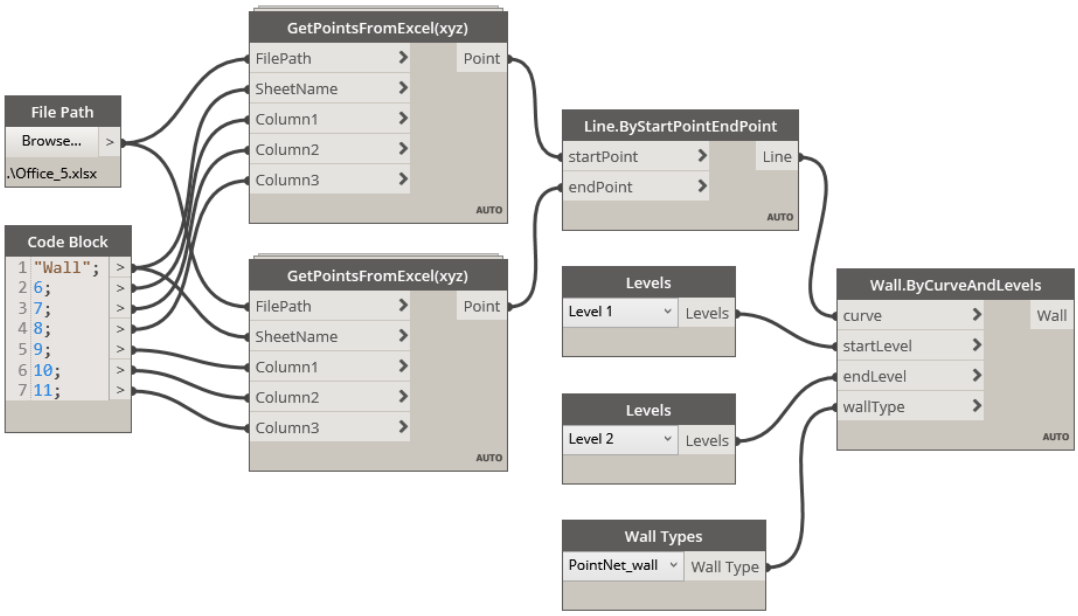

Figure 8: The designed logic for wall objects.

4.2 Hosted elements

Hosted elements include items that are attached or related to structural elements, such as windows, doors, or shelves. The geometric information, such as host walls' start and end points, is crucial in this process. Figure 9 provides an example of the required geometric information of a hosted object, where the first six columns indicate the location coordinates of the start and end points of host walls. The term “chord length” represents the distance between the object’s center and the starting point of its respective host wall. With the extracted information, the designed logic for hosted elements locates the center of the objects on host wall baselines and places the objects at these specified locations. It is important to note that object dimensions are automatically adjusted post-placement based on object dimensions.

StartPoint.X	StartPoint.Y	StartPoint.Z	EndPoint.X	EndPoint.Y	EndPoint.Z	ChordLength	FamilyType	Bookcase_DistoffFloor	Bookcase_Width	Bookcase_Depth	Bookcase_Height
-12.773	17.249	0	-7.748	17.249	0	0.5	lBased_Shelvir	0	1	0.35	2.8
-12.773	17.249	0	-7.748	17.249	0	1.5	lBased_Shelvir	1.4	1	0.35	1.4
-12.773	17.249	0	-7.748	17.249	0	2.5	lBased_Shelvir	1.4	1	0.35	1.4
-12.773	17.249	0	-7.748	17.249	0	3.5	lBased_Shelvir	0	1	0.35	2.8
-12.773	17.249	0	-7.748	17.249	0	4.5	lBased_Shelvir	0	1	0.35	2.8
-7.748	20.404	0	-12.773	20.404	0	2.725	lBased_Shelvir	1.4	0.9	0.35	1.4
-7.748	20.404	0	-12.773	20.404	0	3.625	lBased_Shelvir	1.4	0.9	0.35	1.4
-7.748	20.404	0	-12.773	20.404	0	4.525	lBased_Shelvir	1.4	0.9	0.35	1.4
-12.773	17.249	0	-7.748	17.249	0	1.5	ılBased_Cabin	0	1	0.45	0.75
-12.773	17.249	0	-7.748	17.249	0	2.5	ılBased_Cabin	0	1	0.45	0.75

Figure 9: An example of geometric information of hosted objects.

4.3 Non-hosted elements

Non-hosted elements, such as furniture, provide more flexibility as they are movable and rotatable. Their placement requires only the location coordinates of the object's center, along with dimensions and rotation angles if necessary.

4.4 Code Blocks

Instead of using substantial nodes to create the designed scripts, code blocks can be deployed to minimize the number of nodes so as to reduce computation. A code block in Dynamo is a node containing numbers, strings,

formulas, and scripts to perform more sophisticated tasks that allow users to customize and interact with all the functionalities in Dynamo (Jezyk et al., 2019). Moreover, customized functions are stored in a code block and can be recalled anywhere within Dynamo. For example, Figure 10 illustrates a code block being used to locate and select host wall elements.

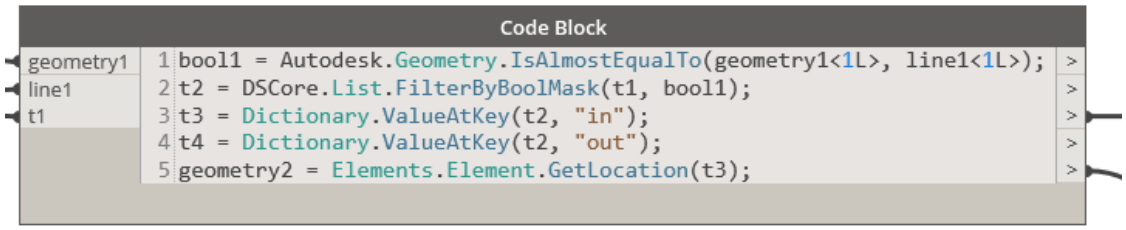

Figure 10: Example of code block application.

4.5 Results

Figure 11 presents the established logic for non-hosted elements. Figure 12 provides an example of the as-built models generated by the proposed framework. The results highlight the potential of creating 3D as-built BIMs from well-segmented segmented point clouds with structural, hosted, and non-hosted elements given the necessary information (as displayed in Figure 13). In addition, this generic modeling process is versatile and applicable to various element types.

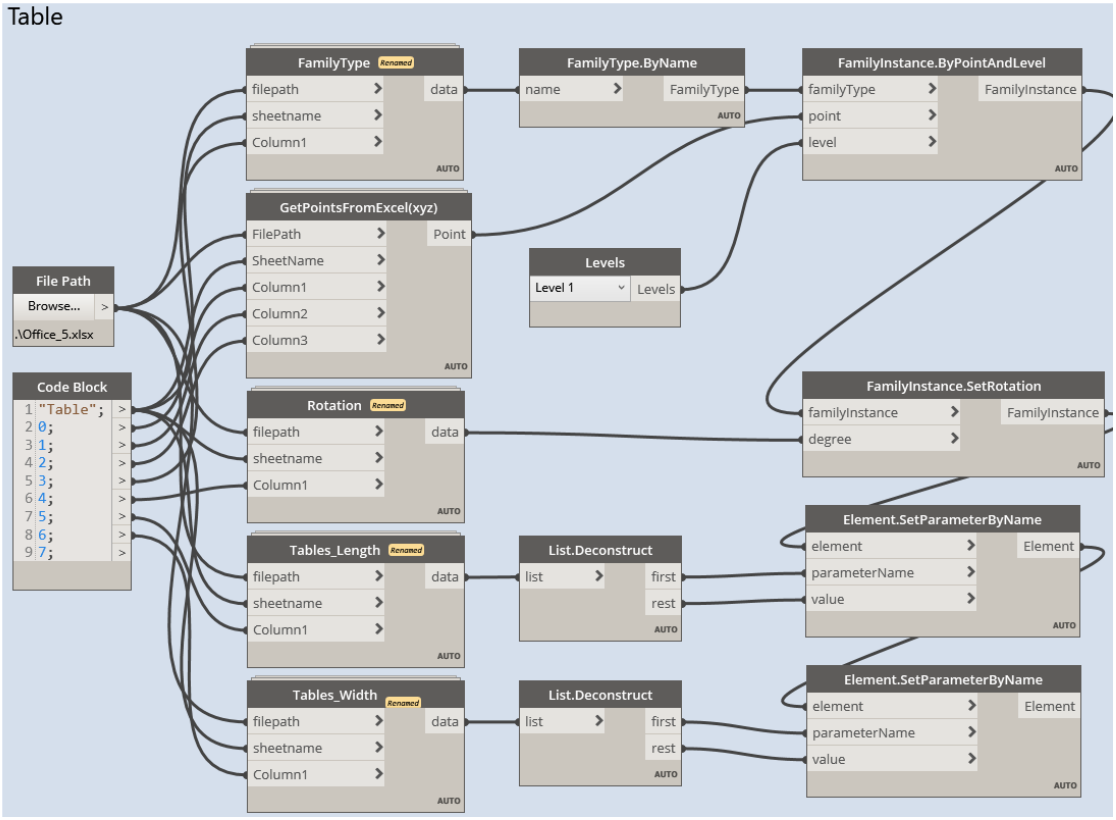

Figure 11: The designed logic for non-hosted elements.

Figure 12: Examples of the as-built BIMs generated by the proposed methodology.

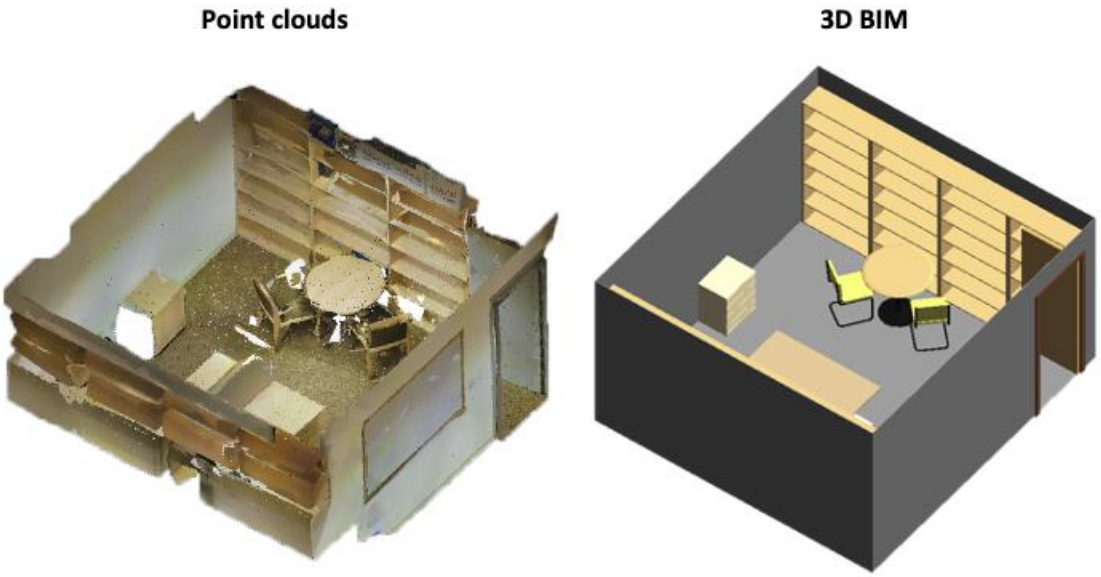

Figure 13: A comparison of point clouds and 3D BIM.

Figure 14 compares the modeling time between the manual modeling approach and the proposed method. The manual modeling process is time-consuming in that it generally takes up to four hours to construct a 3D BIM for a room. On the other hand, the parametric approach is more efficient and less labor-intensive. The time needed for creating as-built BIMs leveraging the parametric approach for a room decreased over time. The initial stage of designing a logical workflow takes a few hours to construct, test, and modify for one room. After accomplishing the generic logic, the average time spent for a single room is significantly reduced. Take an office room as an example; the average modeling time decreased to 35 minutes after establishing the designed logic. In addition, the dimensions and rotation angles of objects are modified simultaneously.

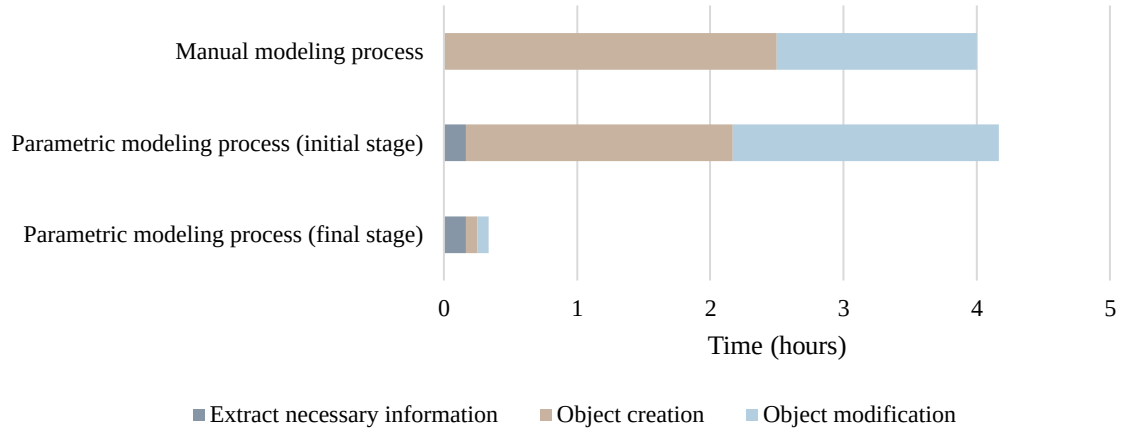

Figure 14: Modeling time comparison chart.

Table 4: RMSE for various parameters of hosted and non-hosted objects.

Object Type	**Family Type**	**Parameter**	**RMSE (mm)**
Hosted	Windows	Width	4.28
		Height	1.94
		Center Location	3.72
	Doors	Width	0.55
		Height	0.35
		Center Location	3.81
	Boards	Width	2.16
		Height	0.25
		Center Location	1.54
	Bookcase	Width	3.32
		Height	5.05
		Depth	1.07
		Center Location	5.47
Non-hosted	Table	Width	1.98
		Height	2.73
		Length	0.66
		Center Location	4.93
	Chair	Width	2.63
		Height	1.07
		Depth	3.66
		Center Location	2.28

The Intersection over Union (IoU) is a widely recognized metric in computer vision, designed to quantify the degree of overlap between predicted data and ground truth, thereby providing a measure of accuracy. Within the framework of Case Study 1, the authors employed an IoU analysis to compare geometric information, which was autonomously extracted using Python scripts, with the ground truth data, as illustrated in Figure 15. The ground truth data was derived from as-built BMIs that were manually modeled by an experienced Revit user. This data is assumed to represent the ground truth. As a result, the overall IoU for structural elements was 96.35%. For hosted and non-hosted elements, the researchers calculated the errors associated with each parameter (such as width, height, and center location), and these findings are compiled in Table 4. The root-mean-square error (RMSE) for hosted and non-hosted elements were 3.634 and 2.607 mm, respectively.

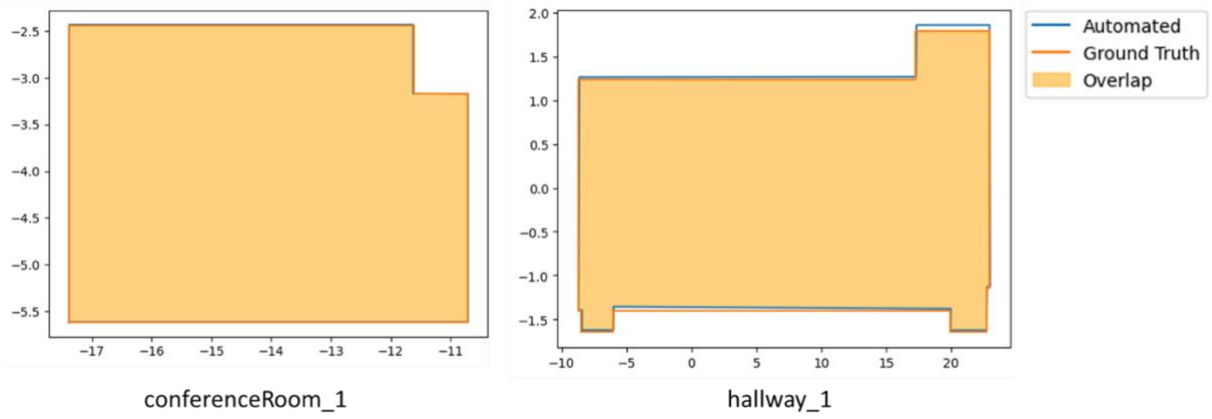

Figure 15: Illustration of IoU for structural elements in Case Study 1.

5. PARAMETRIC MODELING PROCESS – CASE 2

The Medical Center Mechanical Room dataset was chosen to verify the proposed framework's applicability to varied element types. The dataset is a 3D model containing structural, mechanical, and plumbing elements. This second case implementation primarily focuses on mechanical and plumbing system elements, elaborating on the designed logic for their creation. 'Clockwork for Dynamo', 'Springs for Dynamo', and 'MEPover' (MEPover, 2016) packages are mainly utilized in the designed logic for the second case study.

5.1 Parametric Information extraction

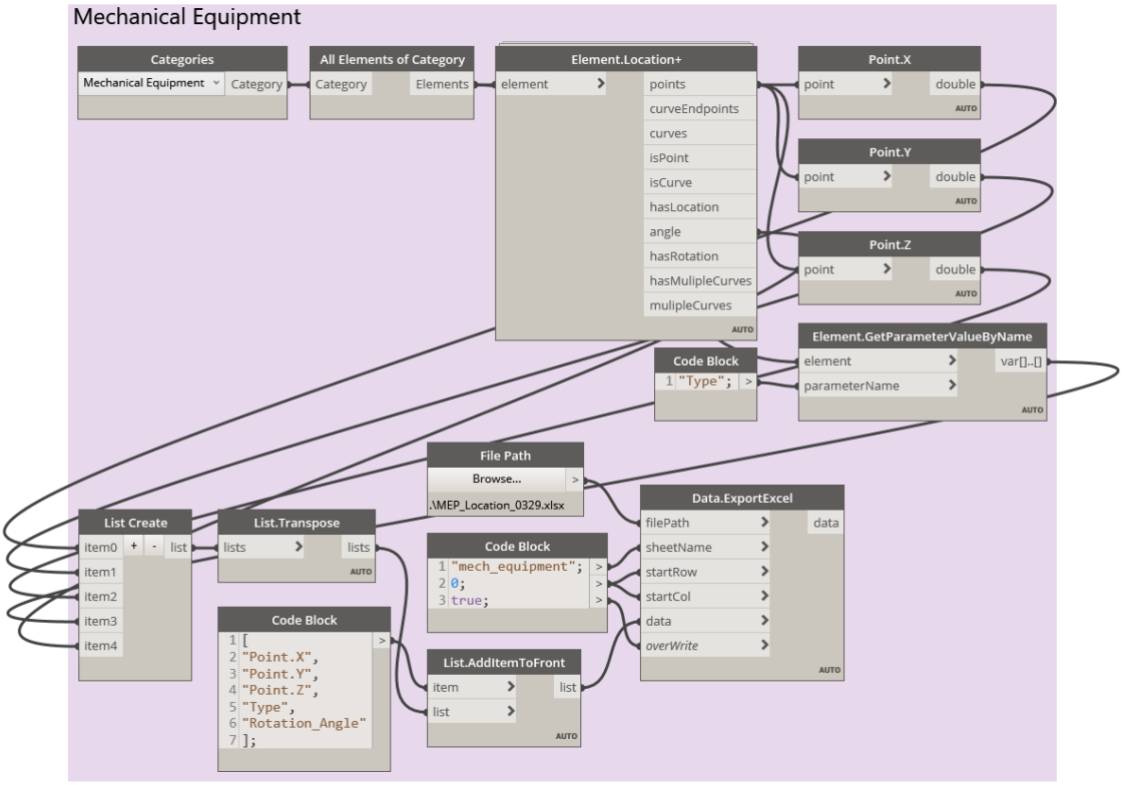

Figure 16: The designed logic for information extraction of MEP elements.

In this second case, Autodesk® Dynamo is utilized for extracting the required information. The designed logic selects all elements from a specified category (for instance, duct elements) and extracts their location coordinates and other information. This information is organized into a list and store in a spreadsheet for future use. Figure 16 presents the designed logic for extracting necessary information of mechanical equipment elements.

5.2 Mechanical and Plumbing Elements

The creation of mechanical equipment elements follows the workflow for non-hosted objects, requiring the location coordinates of the object's center, family type, and dimensions. On the other hand, plumbing elements can be divided into two categories: (1) ducts and pipes, and (2) fittings.

5.2.1 Ducts and pipes

Creating pipes and ducts requires the coordinates of the start and end points, representing their centerlines. The designed logic generates duct and pipe elements after forming these centerlines, as depicted in Figure 17.

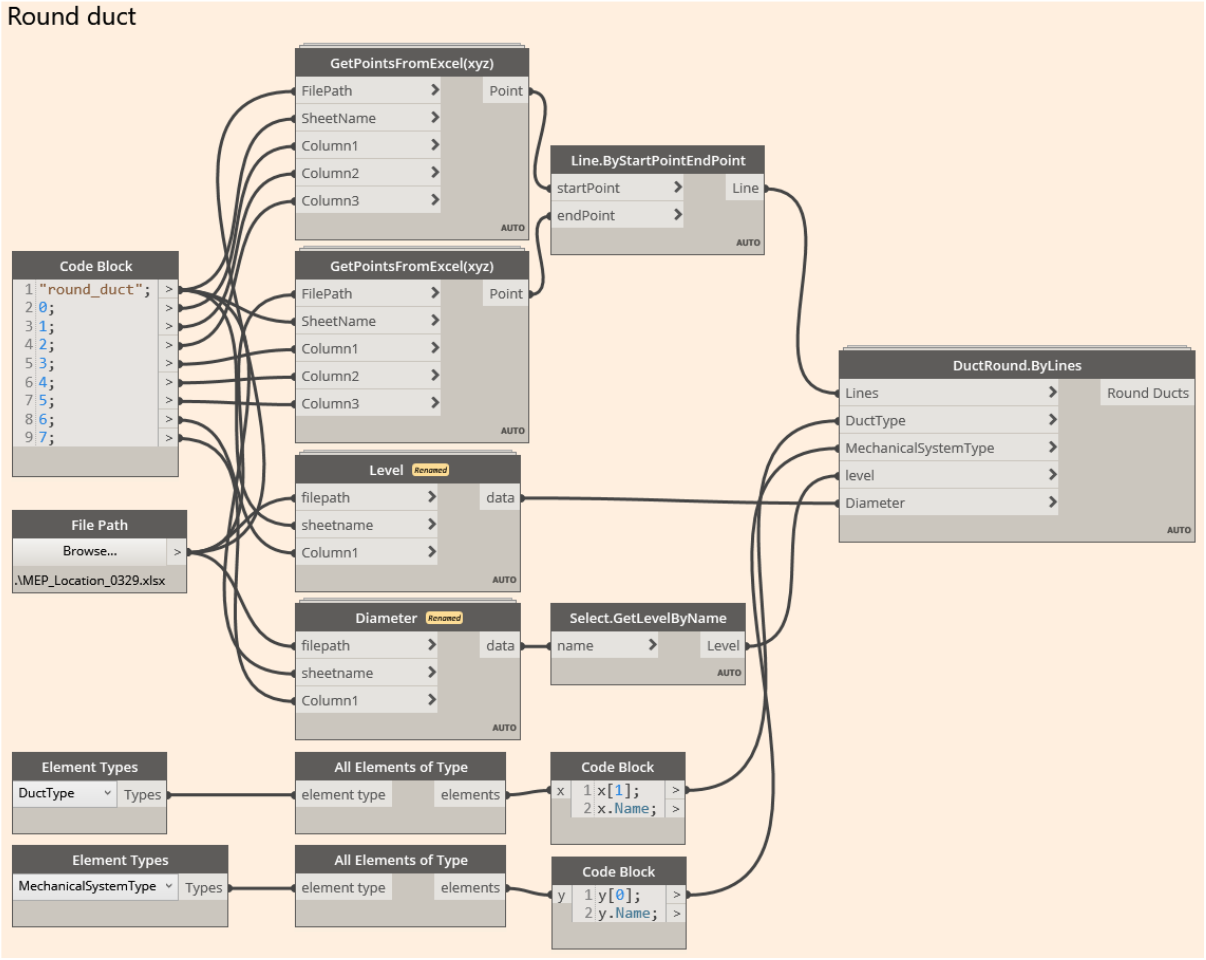

Figure 17: The designed logic for duct and pipe elements.

5.2.2 Fittings

After creating ducts and pipes, fittings that connect sections of ducts or pipes are generated manually. Since existing Dynamo nodes are not capable of creating fittings with specified size and family type, the authors developed a Dynamo script to serve the need. Dynamo Player is applied to execute the designed script more efficiently since it allows users to operate the script in Revit without Dynamo. Figure 18 describes how Dynamo Player works in Revit. Users can significantly expedite the modeling process by selecting pipes or ducts to be connected and assigning family types with Dynamo Player.

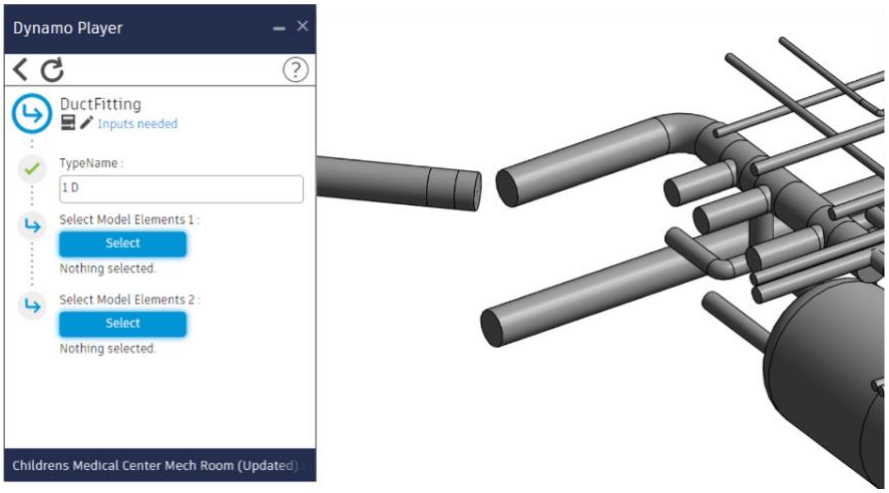

Figure 18: Dynamo script execution with Dynamo Player.

5.3 Results

Figure 19 shows the model generated by the proposed framework. The designed logic can collect the necessary information into a spreadsheet with well-organized worksheets presenting each object type. The outcome highlights the potential of the proposed parametric approach to be versatile across different disciplines, including mechanical and plumbing systems. However, it is important to note that further automation of this method is currently limited due to the restricted availability of Dynamo packages for MEP.

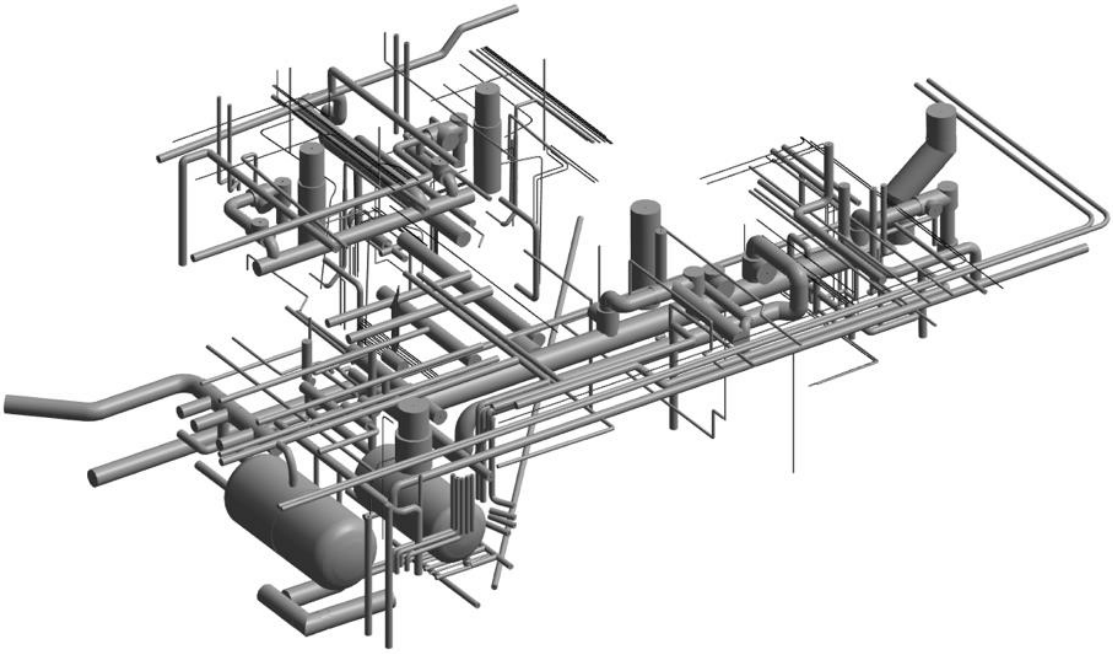

Figure 19: Mechanical and plumbing elements generated by the proposed method.

In the presented case study, all computations were performed on a system equipped with an Intel® Core™ i7-8700 CPU, 64 GB RAM, operating on a 64-bit system. The process of extracting geometric information was completed in 12 seconds, while the creation of 3D BIMs was accomplished in 37 seconds. The authors utilized Dynamo Players to manually create connections for pipes and ducts, adding an additional 41 minutes to the total time. Accordingly, the overall time spent in generating this model was slightly under 42 minutes.

The Industry Foundation Classes (IFC) is an object-oriented data model that is standardized and platform neutral,

thus can be used in any BIM software. In this study, we utilized the quantity of elements in each IFC type to obtain the accuracy rate between the as-designed and the generated BIM, as presented in Table 5. The comparative analysis was restricted to mechanical and plumbing elements, with an overall retrieval rate of 93.9%.

Table 5: Accuracy comparison for various IFC Types in Case Study 2.

IFC Type	Ground Truth	Generated BIM	Retrieval Rate
IfcDistributionPort	1711	1612	94.2%
IfcDuctFitting	293	242	82.6%
IfcDuctSegment	556	555	99.8%
IfcPipeFitting	6	5	83.3%
IfcRelConnectsPorts	574	550	95.8%
IfcSystem	275	243	88.4%
Overall	3415	3207	93.9%

6. CONCLUSIONS

This study presents a 3D semi-automated as-built modeling framework by applying a parametric approach, which is divided into pre-processing and parametric modeling stages. The proposed method extracts necessary modeling information through the integration of scripts and CloudCompare. Most of the required information is extracted automatically, while CloudCompare is utilized to manually determine family types and orientation angles. The parametric modeling is performed by Dynamo, which allows users to create 3D models with high efficiency and provide high flexibility with powerful built-in functions and can also execute Python scripts for more sophisticated tasks. Also, the proposed approach leverages built-in family objects in Revit allows widespread access without searching for objects from other sources. This study elaborates the design process for parametric modeling in Dynamo with details, including the required parameters, the entire designed logic for some element types, to serve as a guideline for future research in Scan-to-BIM. Building on our previous study (Ma et al., 2022), this semi-automated method has been applied to two real-world datasets to validate the practicality of implementing the designed workflow for modeling structural, architectural, furniture, mechanical, and plumbing elements. Based on the results, it can be inferred that the proposed framework has promising potential in generating 3D BIMs and significantly reduce modeling time, thus indicating its practicality and benefits for Scan-to-BIM applications.

Unlike the parametric modeling approach, the manual modeling process is time-consuming and error-prone. Experienced modelers have to visually identify and generate objects by referencing imported point clouds. Meanwhile, they have to create multiple new family types to fit the actual sizes of recognized objects. That is, they may neglect some objects or create objects with incorrect sizes. Once a change has been made to the dimension of an object, modelers have to revisit the created family object to reflect the change. On the other hand, the parametric modeling approach with Dynamo has the potential to improve modeling efficiency by defining required parameters and manipulating nodes and scripts, as it can reduce human effort in creating tons of family objects and save time spent on updating the parameters of multiple objects individually. Consequently, the proposed framework has the potential to create as-built models efficiently and facilitate Scan-to-BIM.

The proposed framework, while promising, does have a few areas of improvement. First of all, the proposed method relies on accurate point cloud segmentation, which may not always be feasible or achievable. Secondly, existing Dynamo packages for assisting MEP system modeling are too limited to perform specific tasks. Plus, the proposed research method utilizes built-in family objects in Autodesk® Revit® with limited specifications. Lastly, the devised workflow should be examined to validate its applicability and accuracy in other disciplines, such as electrical elements. These limitations highlight areas for enhancement within our framework and underscore potential future research in the Scan-to-BIM field.

To further advance the parametric modeling approach and overcome the aforementioned limitations, future studies could focus on several key areas. One effective approach is to employ deep learning-based instance-level semantic segmentation algorithms, which can not only classify points at a category level but also at an instance level, enabling more accurate modeling of distinct objects within the same category. Another focus area would be the development of customized nodes and packages for MEP system elements, which could expand the versatility of Dynamo in this context. Furthermore, expanding the family object library within Revit to accommodate a broader

range of specifications can help reflect real-world objects more accurately. It is important to note that increasing the number of object types might increase the recognition process and potentially compromise accuracy, highlighting the need of maintaining a balance between versatility and accuracy in future development. Lastly, investigating the adaptability of the developed workflow across other disciplines would verify its broad applicability. These focus areas have the potential to considerably enhance the modeling process, paving the way toward achieving a fully automated Scan-to-BIM process.

In conclusion, this study highlights the applicability and adaptability of the parametric modeling approach. The proposed method encompasses a wide variety of indoor objects, aiming at reducing modeling time and human intervention, which is a significant contribution to the field of Scan-to-BIM. The extraction of geometric information and creation of 3D BIM elements, traditionally considered labor-intensive and error-prone, are mostly automated in our semi-automated approach. In the proposed method, manual intervention is primarily required for the determination of family types and orientation angles using CloudCompare and the selection of ducts for connection. Although manual intervention is still necessary, this method substantially diminishes the likelihood of human errors and reduces modeling time. The workflow represents one of the early attempts to model a variety of indoor objects using parametric modeling at once, which adds to its strength and uniqueness. Overall, the findings from this research support the use of parametric modeling as a promising technique for Scan-to-BIM, and future research can build upon this work to improve and expand the methodology further.

ACKNOWLEDGMENTS

The authors would like to thank G4 Spatial Technologies for their support in providing the Medical Center Mechanical Room dataset that facilitated the development of this research.

REFERENCES

Alizadehsalehi, S., Hadavi, A., Huang, J.C., 2020. From BIM to extended reality in AEC industry. Autom. Constr. 116, 103254. https://doi.org/10.1016/j.autcon.2020.103254

Allegra, V., Di Paola, F., Lo Brutto, M., Vinci, C., 2020. Scan-To-BIM For The Management Of Heritage Buildings: The Case Study Of The Castle Of Maredolce (Palermo, Italy), in: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Presented at the XXIV ISPRS Congress, Commission II (Volume XLIII-B2-2020) - 2020 edition, Copernicus GmbH, pp. 1355–1362. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1355-2020

Andriasyan, M., Moyano, J., Nieto-Julián, J.E., Antón, D., 2020. From Point Cloud Data to Building Information Modelling: An Automatic Parametric Workflow for Heritage. Remote Sens. 12, 1094. https://doi.org/10.3390/rs12071094

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1534–1543. https://doi.org/10.1109/CVPR.2016.170

Bansal, V.K., 2021. Integrated Framework of BIM and GIS Applications to Support Building Lifecycle: A Move toward nD Modeling. J. Archit. Eng. 27, 05021009. https://doi.org/10.1061/(ASCE)AE.1943-5568.0000490

Barazzetti, L., 2016. Parametric as-built model generation of complex shapes from point clouds. Adv. Eng. Inform. 30, 298–311. https://doi.org/10.1016/j.aei.2016.03.005

Bosché, F., Ahmed, M., Turkan, Y., Haas, C.T., Haas, R., 2015. The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. Autom. Constr., 30th ISARC Special Issue 49, 201–213. https://doi.org/10.1016/j.autcon.2014.05.014

Brumana, R., Oreni, D., Barazzetti, L., Cuca, B., Previtali, M., Banfi, F., 2020. Survey and Scan to BIM Model for the Knowledge of Built Heritage and the Management of Conservation Activities, in: Daniotti, B., Gianinetto, M., Della Torre, S. (Eds.), Digital Transformation of the Design, Construction and Management Processes of the Built Environment, Research for Development. Springer International Publishing, Cham, pp. 391–400. https://doi.org/10.1007/978-3-030-33570-0_35

Chen, J., Cho, Y.K., Kim, K., 2018. Region Proposal Mechanism for Building Element Recognition for Advanced Scan-to-BIM Process 221–231. https://doi.org/10.1061/9780784481264.022

Davila Delgado, J.M., Oyedele, L., Ajayi, A., Akanbi, L., Akinade, O., Bilal, M., Owolabi, H., 2019. Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. J. Build. Eng. 26, 100868. https://doi.org/10.1016/j.jobe.2019.100868

Dieckmann, A., 2013. Clockwork For Dynamo Package [WWW Document]. URL https://github.com/andydandy74/ClockworkForDynamo (accessed 10.18.21).

Esfahani, M.E., Rausch, C., Sharif, M.M., Chen, Q., Haas, C., Adey, B.T., 2021. Quantitative investigation on the accuracy and precision of Scan-to-BIM under different modelling scenarios. Autom. Constr. 126, 103686. https://doi.org/10.1016/j.autcon.2021.103686

Gruber, C., Weiner, T., Zuchtriegel, R., 2018. BIM for tunnelling for a company – Approaches and strategies. Geomech. Tunn. 11, 366–373. https://doi.org/10.1002/geot.201800017

Heaton, J., Parlikad, A.K., Schooling, J., 2019. Design and development of BIM models to support operations and maintenance. Comput. Ind. 111, 172–186. https://doi.org/10.1016/j.compind.2019.08.001

Hough, P.V.C., 1962. Method and means for recognizing complex patterns. US3069654A.

Jezyk, M., Mode Lab, John Pierson of Parallax Team, 2019. What's a Code Block | The Dynamo Primer [WWW Document]. URL https://primer.dynamobim.org/07_Code-Block/7-1_what-is-a-code-block.html (accessed 4.11.22).

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

Kontoudaki, A., Georgopoulos, A., 2022. HBIM Library Development For A Doric Temple Column. Presented at the The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences, Nice, France.

Leite, F., 2019. BIM for Design Coordination: A Virtual Design and Construction Guide for Designers, General Contractors, and MEP Subcontractors, 1st ed. Wiley.

Ma, J.W., Lee, Y.-C., Leite, F., 2022. A Practical Application Using Parametric Modeling for As-Built BIM Generation from Point Clouds 830–838. https://doi.org/10.1061/9780784483961.087

Macher, H., Roy, L., Landes, T., 2021. Automation of windows detection from geometric and radiometric information of point clouds in a scan-to-BIM process. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLIII-B2-2021, 193–200. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-193-2021

MEPover, 2016. MEPover Package [WWW Document]. URL https://sites.google.com/site/bimstallatie/dynamo/mepover-package (accessed 10.18.21).

Moyano, J., Gil-Arizón, I., Nieto-Julián, J.E., Marín-García, D., 2022. Analysis and management of structural deformations through parametric models and HBIM workflow in architectural heritage. J. Build. Eng. 45, 103274. https://doi.org/10.1016/j.jobe.2021.103274

Pepe, M., Costantino, D., Restuccia Garofalo, A., 2020. An Efficient Pipeline to Obtain 3D Model for HBIM and Structural Analysis Purposes from 3D Point Clouds. Appl. Sci. 10, 1235. https://doi.org/10.3390/app10041235

Son, H., Kim, C., Turkan, Y., 2015. Scan-to-BIM - An Overview of the Current State of the Art and a Look Ahead. ISARC Proc. 1–8.

Tang, P., Huber, D., Akinci, B., Lipman, R., Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. Autom. Constr. 19, 829–843. https://doi.org/10.1016/j.autcon.2010.06.007

Ullah, K., Lill, I., Witt, E., 2019. An Overview of BIM Adoption in the Construction Industry: Benefits and Barriers, in: Lill, I., Witt, E. (Eds.), 10th Nordic Conference on Construction Economics and Organization, Emerald Reach Proceedings Series. Emerald Publishing Limited, pp. 297–303. https://doi.org/10.1108/S2516-285320190000002052

U.S. Bureau of Labor Statistics, 2021. Labor Productivity for Selected Construction Industries [WWW Document]. URL https://www.bls.gov/lpc/construction.htm#footnote1 (accessed 3.3.22).

Venkov, D., 2015. Springs for Dynamo Package [WWW Document]. URL https://github.com/dimven/SpringNodes (accessed 10.18.21).

Volk, R., Stengel, J., Schultmann, F., 2014. Building Information Modeling (BIM) for existing buildings — Literature review and future needs. Autom. Constr. 38, 109–127. https://doi.org/10.1016/j.autcon.2013.10.023

Wang, B., Yin, C., Luo, H., Cheng, J.C.P., Wang, Q., 2021. Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data. Autom. Constr. 125, 103615. https://doi.org/10.1016/j.autcon.2021.103615

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