

Vision-Based Progress Monitoring of Building Structures Using Point-Intensity Approach

Varun Kumar Reja ^{a,b}, Parth Bhadaniya ^a, Koshy Varghese ^a, Quang Ha ^b

^a Department of Civil Engineering, IIT Madras, India

^b Faculty of Engineering & Information Technology, UTS, Australia

E-mail: varunreja7@gmail.com, ce19m008@gmail.com, koshy@iitm.ac.in, quang.ha@uts.edu.au

Abstract –

Progress quantification of construction projects is critical for project managers to manage projects effectively. The trade-off between computation time and accuracy is a key aspect while selecting the quantification method. Though accuracy is essential, project managers require real-time information about quantities of work completed on various building components to make timely decisions. Several researchers have developed individual pipelines using vision-based technologies for automated progress quantification. However, they face significant implementation challenges, including higher computational complexity, skilled personals, and costly equipment. Hence, this study aims to define an easy-to-implement pipeline to quantify the holistic progress and element-wise progress of a building. For executing this, the method utilizes point intensity as a fundamental parameter. The progress of a specified element is calculated by comparing the number of points in the as-built model and the number of anticipated points from the as-designed model of the same point intensities. The method directly integrates the as-designed BIM model with the scan data through a user-friendly visual programming tool - Grasshopper3D for progress quantification. The workflow provides one-click progress report generation with minimal inputs from the users for basic alignment of the imported as-built data. As the point clouds are directly addressed as a single entity, and to-be intensity has been calculated through an intensity-based randomized approach, the method utilizes less computation for the whole process.

Keywords –

Automated Progress Monitoring, Point-Intensity, Scan-vs-BIM, Progress Quantification, Visual Programming, Single-Click Report Generation, BIM-Based Tracking, Point Clouds, Automation in Construction Management

1 Introduction

Monitoring of construction projects is a key aspect of management. Construction progress monitoring has been attempted through various technologies [1]. These technologies are based on bar codes, QR codes, RFIDs, range imaging, photogrammetry, videogrammetry, laser scanning, etc. These technologies are much advanced than primitive pen and paper-based or Daily Progress Report (DPR) based progress monitoring.

Though several advanced technologies have been developed, there has been little evidence of their application at construction sites. Currently, a majority of construction projects predominantly use pen and paper-based or daily updated Excel sheet-based (DPR-based) methods for progress monitoring. However, these methods are time-consuming, non-systematic, less accurate, and require too much human intervention making the process inefficient [2].

The on-site adoption of advanced methods like Scan-to-BIM or Scan-vs-BIM has been to a lesser extent. These methods require significant computation of data, certain specific skills as well as domain knowledge for implementation. In addition, their current industrial version requires substantial human intervention, making them tedious to run at frequent intervals.

Lately, most of the sites have used 3D spatial data to visualize and monitor progress. For quantification of progress, Scan-to-BIM and Scan-vs-BIM are the two-key techniques found in the literature. These methods process the point cloud data to generate an as-built BIM model or compare it through a computationally extensive process. However, due to the complex task of as-built modeling and quantification of progress, on-site implementation of the vision-based progress monitoring pipelines has been the bare minimum.

This study recognizes the gaps discussed above and aims to present a novel pipeline for progress quantification that can be easily implemented at project sites. The pipeline is focused on detecting the volumetric progress using point cloud data directly. For this paper, the scope is limited to quantifying the

progress of building projects. Hence, the two objectives of this study are as follows:

1. To develop an on-site implementable pipeline to utilize as-built 3D point clouds and their derived properties for progress quantification.
2. To experimentally validate the pipeline and present the key factors which make the pipeline more adoptable.

This paper is organized as follows. After the introduction, Section 2 discusses the state-of-the-art quantification methods and their challenges for on-site implementation. Section 3 presents the methodology. Section 4 presents the system architecture for the proposed pipeline. Section 5 presents the experimental results and discussion. This is followed by a conclusion highlighting the features of the proposed pipeline along with future work given in Section 6.

2 Existing progress quantification approaches

Progress quantification is essential for managers to make decisions about the use and allocation of project resources. However, quantification on construction sites is still a tedious and time-consuming process due to conventional techniques. Moreover, the automated and semi-automated methods present in the existing literature have not been widely adopted in construction projects due to various factors.

Recently there has been a variety of progress monitoring approaches pipelines that have been developed in recent years. Although many implement progress visualization using Augmented Reality (AR) [3], Virtual Reality (VR) [4], Mixed Reality (MR) [2], and Extended Reality (XR) [5] environments, only a few attempted quantification.

Existing approaches have used quantification using image processing [6], object detection [7], model-based recognition [8] based methods. Apart from the image-based techniques, these methods are generally part of the two main methods, i.e., Scan-to-BIM and the Scan-vs-BIM approach. Quantification has also been attempted by integrating the two approaches [9]. Existing methods in both techniques require converting into BIM or comparing the point cloud model by overlapping, which is computationally expensive and requires complex algorithms. The state-of-the-art of both methods is discussed in this section with the various challenges faced for on-site implementation.

2.1 Scan-to-BIM

The Scan-to-BIM approach is based on converting an as-built point cloud to a BIM model. This is

implemented by detecting the elements from the point clouds using various heuristics [10][11] or learning-based algorithms [4][12] and then replacing them with corresponding BIM elements. The progress quantification can be easily done by directly generating the bill of quantities [13].

The Scan-to-BIM approach uses a lower order point cloud model to convert to a more informative BIM model. This process is computationally expensive, time-consuming and expert experience is required to implement it successfully.

In heuristic-based modeling, a set of primary input data about the common elements in the point cloud is required to initiate the process. This data must be manually fed into the logic to detect elements requiring significant time and effort. Secondly, the heuristics-based method cannot be expected to recognize all the elements as some are unique to a particular project.

Few studies have recently explored learning-based methods for detecting elements from point clouds [14] [15]. However, the site-wide implementation of these approaches has not been possible due to various factors. Firstly, for learning-based methods, a large dataset of existing elements for training is required, which is not available currently and is difficult to generate for various elements. Through an approach to systematically generate training data have been suggested [16]. Secondly, computational requirements are high to process and test the large dataset. Lastly, a skilled workforce is required to implement and operate these pipelines regularly.

2.2 Scan-vs-BIM

On the other hand, the Scan-vs-BIM approach detects progress by overlapping the as-designed and as-built BIM models [17]. This is done either by applying thresholds [2] or point occupancy-based methods [4]. Once the elements are detected, they are labeled as complete or incomplete in the as-designed model. The quantification can be performed by generating the quantities of completed elements.

Likewise, Scan-vs-BIM uses a comparative approach. Though the comparison in this process is computationally less costly than Scan-to-BIM, there is a lot of pointwise processing to apply thresholds or check the model occupancy.

On the one hand, progress quantification is critical to project managers for efficient management. On the other hand, existing quantification methods are not easily implementable at construction sites. Therefore, this study intended to fill this gap by proposing a pipeline for vision-based progress monitoring, which is easily implementable at construction projects, with much less computation and skills. The details of the pipeline are discussed in the next section.

3 Methodology

An easy-to-use pipeline has been developed in this study for holistic and element-wise progress monitoring in the construction execution phase. The primary focus of this study is to compute the progress using the point intensity approach; however, the essential data acquisition and preprocessing steps are mentioned to complete the pipeline structure. The overview of each step and the core concept of the pipeline have been discussed in this section.

3.1 Data acquisition, preprocessing & registration

Figure 1 shows the overall pipeline adopted in this study. As an input, the pipeline receives site-acquired as-built point cloud data, captured using suitable vision-based technology.

Next, for preprocessing, firstly, the point clouds are extracted in a suitable file type format. Secondly, if multiple point clouds are to be stitched together, they should be registered using existing registration approaches. For our pipeline, noise and outlier removal is recommended but not mandatory as these points do not lie within the boundary region of any elements. However, it is suggested to sub-sample the point cloud through the scope-box method to a range of interest.

Lastly, the as-built point cloud and as-designed model can be registered using manual, automatic, or semi-automatic registration approaches. For this, our method uses the developed 3-point align script, which is a semi-automated approach and works on aligning a defined plane between the two models.

3.2 Progress quantification

Figure 1 shows the overall concept of this part graphically. The key concept of quantification in this pipeline is based on a spatial comparison of as-built point clouds against the as-designed BIM model. The as-built point intensity (points per unit area) can be derived through the scan data, and the equivalent intensities are transferred to as-designed BIM for point count calculation. Using arithmetic ratios of these point counts, holistic or elementwise progress can be derived in percentage completion.

The method first utilizes the as-designed BIM model to derive boundary representatives (B-Reps) of all the elements in the as-designed model. In the BIM model, the elements are usually represented by 3D meshes that show the boundary of the solid elements with infinite thin surfaces called B-Reps.

To calculate points that should be present on the surfaces of element B-reps, a specific threshold has been considered to formulate an offset geometry that

can include the points within specified spatial limits. This offset geometry has been addressed as the boundary region of the element in this study.

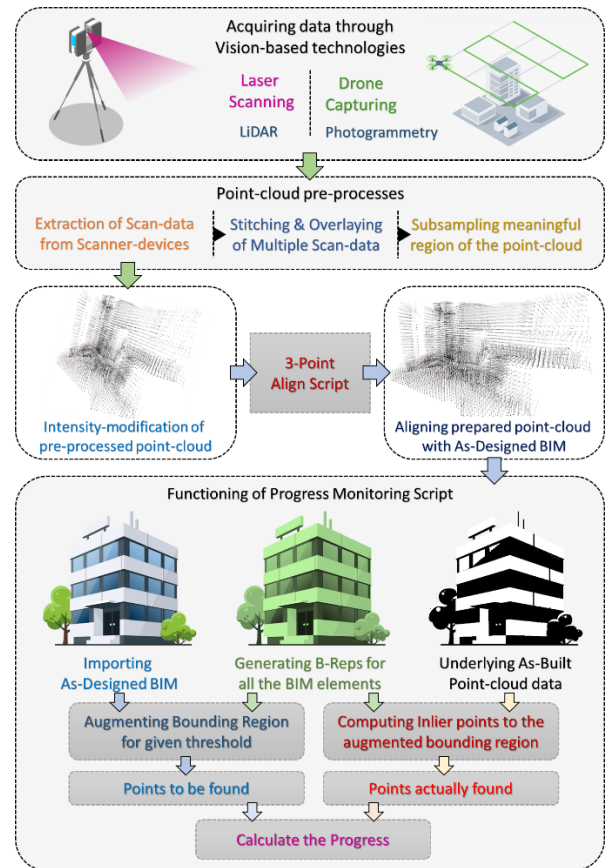


Figure 1 Overall pipeline for point-intensity based progress monitoring

Using few parametric toolsets, B-Rep surfaces or mesh-faces are populated with the given number of randomized points. Multiple researchers are widely using this method to produce accurate synthetic point clouds for experimentation. For uniform generation on each surface, point count is defined by surface area multiplied by the desired intensity. This study utilizes this concept to anticipate point counts on the required surfaces by directly performing basic multiplication of area and intensity instead of actually producing all synthetic points. For anticipating the point count in the as-designed model, users are expected to note and feed the same intensity that has been used while performing the laser-scan or cloud construction through photogrammetry.

Additionally, anticipated points were calculated only through the area of exposed surfaces of the elements. It has been considered that overlaying a portion of such surfaces usually being occluded by exposed surfaces when a realistic scan is being performed in real. To eliminate such partial or complete surfaces, surface

intersection with co-planarity has been utilized as basic logic. Figure 2 shows this graphically.

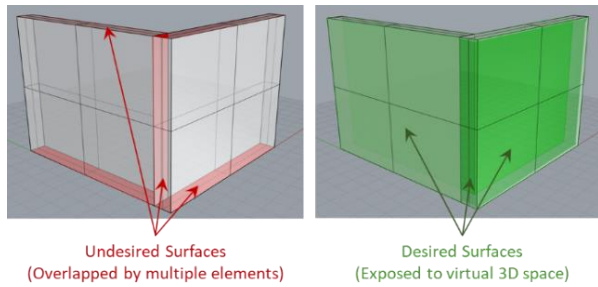


Figure 2 Exposed and occluded surfaces

After aligning the imported as-built point cloud, volumetric bounding regions are overlaid on the scan data. Each bounding region is engaged with an element. The portions of the as-built point cloud being covered were sub-sampled and classified with the particular element identity. Then, as-built point counts for all the elements were derived from the cloud statistics of all the relative sub-samples.

The point counts derived from the BIM model and the scan data are addressed here as 'As-designed Point Counts (PC_{AD})' and 'As-built Point Counts (PC_{AB}).' Graphical visualization of these and their overlay is shown in Figure 3. Both of these entities have been utilized to calculate progress status in three different levels of detail:

1. Holistic progress tracking (computes progress of overall project)
 - a) Cumulative progress calculation (provides unified progress of all the elements)
 - b) Element-wise progress calculation (provides progress elementwise as a list)
2. As-requested progress tracking (provides the progress of the specifically requested elements)

In the following sub-section, the idea is technically elaborated with a logic-flow diagram and pseudocode created for the developed algorithm.

3.3 Core concept

Figure 4 shows the logic in sequence to compute the progress quantities for holistic and as-requested progress tracking. Suppose the preprocessed as-built point cloud and an as-designed BIM are available to compute the progress of a particular element E_i . In that case, the core idea is to compare the number of points that are in the as-built point cloud (N_{ABE_i}) with the expected number of points with the same intensity for the as-designed point cloud (N_{ADE_i}), which can be computed using the equation 1:

$$N_{AD}E_i = \text{surface area of } E_i * \text{intensity of points} \dots \quad (1)$$

Then, the percentage progress for of a particular element E_i can be obtained as

$$\% \text{ Progress of } E_i = \left\{ \frac{N_{AB}E_i}{N_{AD}E_i} \right\} * 100 \dots \dots \quad (2)$$

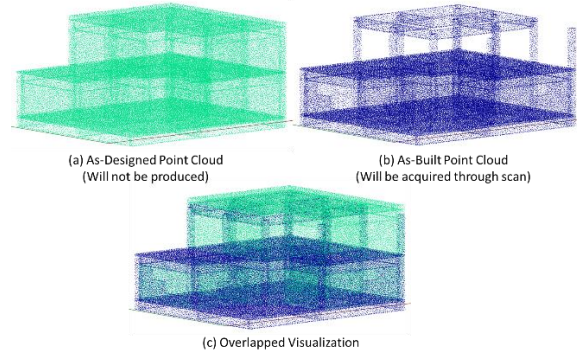


Figure 3 Visuals of as-built and as-designed point clouds with equal point intensities

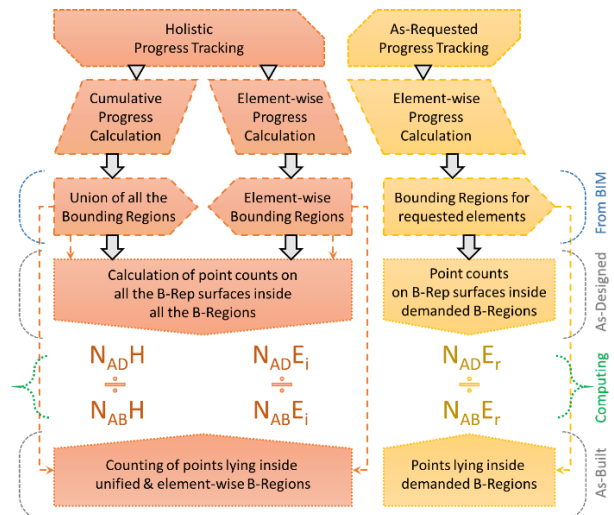


Figure 4 Logic-flow Diagram

3.3.1 Pseudocode

The concept described above has been implemented using the following pseudocode.

Computing as-built progress quantification

Inputs: BIM_{AD} , E_r , PC_{AB} , I

Outputs: PH , PE_i , PE_r

Geometry BReps (GB) extraction & Offset Bound (OB) generation for all as-designed BIM - (BIM_{AD}) elements into a list

- 1 For i in range (count.Elements(BIM_{AD}))
- 2 ... $GB_{AD} [j] = \text{append} (GB_{AP} (i))$

```

3 ...OBAD[ ] = append (offsetbrep.(GBAD(i))
4 UGBAD = BRep.Union(GBAD)

```

Exploding BReps into surfaces (Surf)

```

5 For i in range (length.list (GBAD))
6 ...Explode B-Rep into m surfaces → Surfi(i),...,Surfm(i)
7 ...For j in range (m)
8 .....SurfList [ ] = append (Surfi(i))

```

Retaining only exposed surfaces (ExpSurf)

```

9 For i in range (length.list(SurfList))
10 ...SurfList' = pop(SurfList(i))
11 ...overlay=Intersection.(SurfList(i), SurfList') && Co-
    planarity.(SurfList(i), SurfList')
12 ...If overlay=FALSE
13 .....ExpSurf [ ] = append(SurfList(i))
14 ...SurfList' = SurfList

```

Computing point-counts on exposed surfaces (ExpSurf) as per adopted point intensity (I) and defining to-be point counts for elements

```

15 For j in range (length.list(ExpSurf))
16 ...PCExpSurf [ ] = append(Area.ExpSurf(j) * I)
17 For i in range(length.list(GBAD))
18 ...For j in range (length.list(ExpSurf))
19 .....Inclusion = inlying.surfinbrep(ExpSurf(j), GBAD(i))
20 .....If Inclusion=TRUE
21 .....PCAD [ ] = append(PCExpSurf(j))

```

Counting inlying points from as-built point cloud (PC_{AB})

```

22 For i in range (length.list(GBAD))
23 ...partcloud = subsample.brep(GBAD(i), PCAB )
24 ...PCAB [ ] = append(count.partcloud)

```

Computing holistic progress (PH) , Element-wise progress (PE_i), and As-requested progress (PE_r) (for the requested elements E_r)

```

25 PH = count.subsample.brep(UGBAD,PCAB )/sum.PCAD [ ]
26 PEi = append (PCAB (i)/ PCAD (i))
27 Find i for each rth element in Er & add to r [ ]
28 PEr = append (PCAB (r)/ PCAD (r))

```

4 System architecture

As shown in Figure 5, the system architecture consists of three basic types of 3D processing environments to accomplish multiple technical tasks.

1. BIM modeling environment – For operating and visualizing as-designed BIM model. Autodesk Revit application has been utilized as a BIM platform in this method.
2. Point cloud processing environment – For visualizing and preprocessing the extracted point clouds. An open-source toolset called Cloud Compare has been used in this method.
3. Parametric 3D modeling and Visual Programming – The core logic of the method has been developed

in terms of logical loop sequence to find inlier points within the bounding region of particular element geometries. A combination of a 3D modeling application called Rhinoceros3D and visual programming tool called Grasshopper3D have been utilized to produce the entire script sets.

4.1 Technical execution

As shown in Figure 5, the method primarily takes two inputs in terms of a BIM model and a point cloud into a .xyz /.e57 file type format. Specific additional toolsets to the Grasshopper3D called 'Rhino.inside.Revit' and 'Tarsier' have been utilized to integrate BIM functionalities from Revit and the inbuilt cloud processing functionalities of Tarsier. Additional toolset called 'TT Tool-box' have been employed for enabling the script to auto-generate cumulative / elementwise and as-requested progress monitoring report in the form of a spreadsheet.

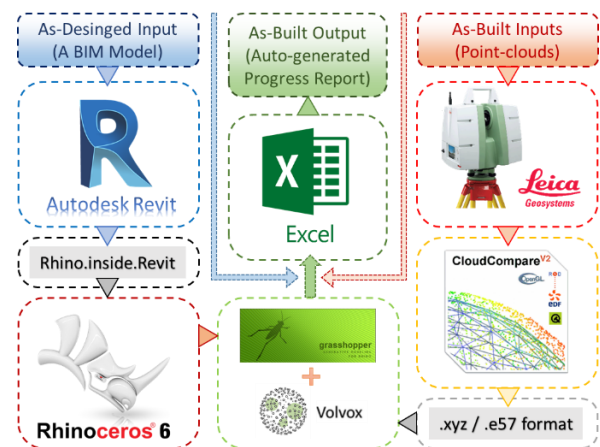


Figure 5 System Architecture Diagram

Technical logic has been fully developed as visual code scripts inside the Grasshopper3D environment. Users are first required to align imported point cloud with as-designed BIM file parallelly opened into Revit BIM environment. For better alignment, the script has been equipped with semi-automated 3-Point align functionality that asks the user to feed three planner points defining a relatively large planner surface in the point cloud and the same planner face in the as-designed BIM model. After the alignment, the user can compute holistic and element-wise progress through a single toggle and can produce a spreadsheet report with a single click.

To produce a partial progress report covering specific elements only, users need to choose the desired element or a set of multiple elements for which an as-requested progress report is required to be generated.

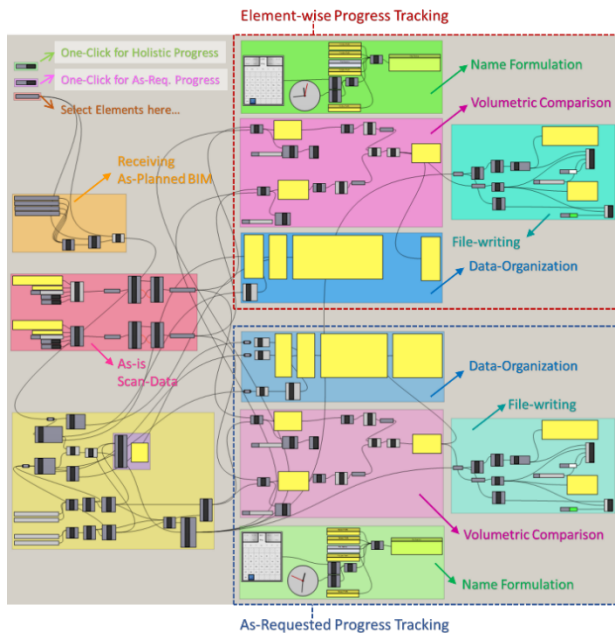


Figure 6 Grasshopper Script Anatomy

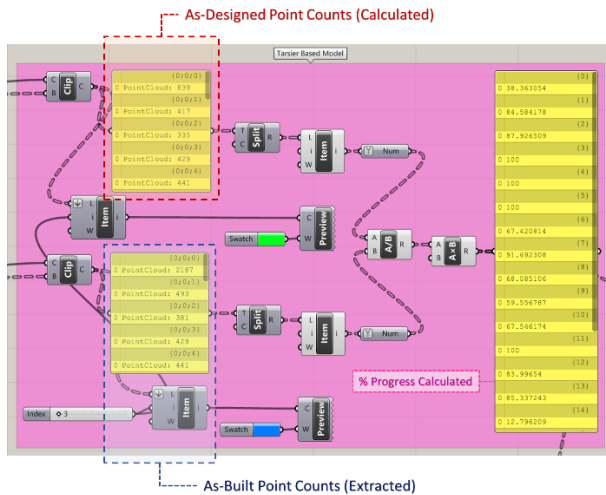


Figure 7 Script for Volumetric Comparison

The developed grasshopper script consists of five different sections shown with different colors. Each of them is automating one particular task in the workflow, as shown in Figure 6.

1. Obtaining BIM Model and current element selection to generate request list.
2. Producing lists of element ID, UUID, and Types for all and requested elements. (Orange & Yellow)
3. Importing scan data from the given storage location and semi-automated alignment. (Red)
4. Deriving counts of spatially present points inside bounding region of elements and percentage progress calculation (Pink)
5. Integration of data into a spreadsheet.
 - a. Data-tree management (Blue)

- b. Automated workbook writing (Pine Green)
- c. Formulating filename (Green)

In the script, the volumetric comparison is a core-functional part that checks for points' spatial presence and prepares element-wise count lists. It finally performs all required arithmetic operations and frames results in terms of percentage completion. A detailed version has been reflected in Figure 7.

4.2 Data-flow

The data flow in the pipeline is as shown in Figure 8. Algorithmic logic embedded in the script first derives bounding regions of element geometries in terms of shell geometries.

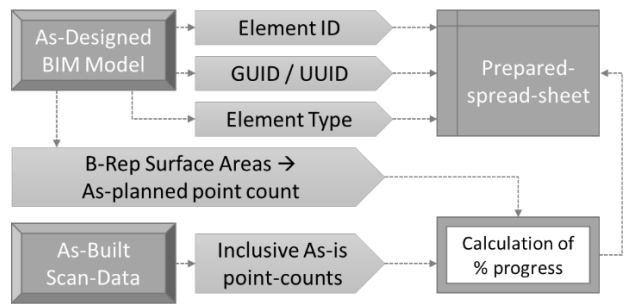


Figure 8 Data-flow diagram

All the BIM elements from the Revit environment usually contain GUIDs (Globally Unique Identifiers) and UUIDs (Universally Unique Identifiers). These identities were extracted using the element passport function from the RiR toolset. Bounding regions of the elements have been generated first by converting mesh-geometries into B-Reps made with infinitely thin surfaces and then offsetting the produced B-Reps at a limited distance to both sides of the solid element surfaces. The offset region is aimed to include representative points of the real scan data for the particular elements. Thus, an inclusion logic to determine the count of inlying points has been included in the workflow through the crop-box subsampling functionality of the tarsier toolset. To calculate point intensity on the exposed surfaces, deconstruct B-Rep functionality has been utilized to find surface areas to multiply them with scan-intensity of the point clouds.

5 Experimentation

The developed method has experimented with an as-designed BIM model (Figure 9a) of a hypothetical building structure and a synthetically produced point cloud (Figure 9d). The actual progress on a given date was formulated through an incomplete version of the same BIM model (Figure 9b). As shown in Figure 9d, the assumed as-built model was produced using the

progress, and elemental progress gives an element-wise progress chart of a building. Both these extents provide detailed information to project managers to make timely and informed decisions.

The pipeline was developed using a user-friendly visual programming tool - Grasshopper3D. The core concept used here was to compare the point intensity in the as-built and as-designed point cloud models.

The pipeline provides a door to wider adoption, less computation cost, and quicker progress quantification. The pipeline is completely automatic and works using minimal computation and skills.

A limitation of this method is that currently, the operational state of the element cannot be recognized. However, with some modifications, temporary fittings like shuttering and formwork can be detected. This will be included in the future study as it will make the pipeline more effective. Also, an effort to implement the current pipeline on point cloud data obtained from the site will be made to see the robustness of the results.

7 References

- [1] Omar T. and Nehdi M. L., Data acquisition technologies for construction progress tracking, *Autom. Constr.*, vol. 70, pp. 143–155, 2016.
- [2] Kopsida M. and Brilakis I., Real-Time Volume-to-Plane Comparison for Mixed Reality-Based Progress Monitoring, *J. Comput. Civ. Eng.*, vol. 34, no. 4, pp. 1–15, 2020.
- [3] Zollmann S., Hoppe C., Kluckner S., Poglitsch C., Bischof H., and Reitmayr G., Augmented Reality for Construction Site Monitoring and Documentation, *Proc. IEEE*, vol. 102, no. 2, pp. 137–154, Feb. 2014.
- [4] Pour Rahimian F., Seyedzadeh S., Oliver S., Rodriguez S., and Dawood N., On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning, *Autom. Constr.*, vol. 110, no. October 2019, p. 103012, Feb. 2020.
- [5] Khairadeen Ali A., Lee O. J., Lee D., and Park C., Remote Indoor Construction Progress Monitoring Using Extended Reality, *Sustainability*, vol. 13, no. 4, p. 2290, Feb. 2021.
- [6] Arif F. and Khan W. A., Smart Progress Monitoring Framework for Building Construction Elements Using Videography–MATLAB–BIM Integration, *Int. J. Civ. Eng.*, vol. 3, 2021.
- [7] Wang Z. *et al.*, Vision-Based Framework for Automatic Progress Monitoring of Precast Walls by Using Surveillance Videos during the Construction Phase, *J. Comput. Civ. Eng.*, vol. 35, no. 1, p. 04020056, 2021.
- [8] Maalek R., Lichti D. D., and Ruwanpura J. Y., Automatic recognition of common structural elements from point clouds for automated progress monitoring and dimensional quality control in reinforced concrete construction, *Remote Sens.*, vol. 11, no. 9, 2019.
- [9] Bosché F., Ahmed M., Turkan Y., Haas C. T., and Haas R., 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.*, vol. 49, pp. 201–213, 2015.
- [10] Ochmann S., Vock R., and Klein R., Automatic reconstruction of fully volumetric 3D building models from oriented point clouds, *ISPRS J. Photogramm. Remote Sens.*, vol. 151, no. October, pp. 251–262, 2019.
- [11] Nikoohemat S., Diakité A. A., Zlatanova S., and Vosselman G., Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management, *Autom. Constr.*, vol. 113, no. May, p. 103109, 2020.
- [12] Chen J., Kira Z., and Cho Y. K., Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction, *J. Comput. Civ. Eng.*, vol. 33, no. 4, pp. 1–10, 2019.
- [13] Mahami H., Nasirzadeh F., Ahmadabadian A. H., and Nahavandi S., Automated progress controlling and monitoring using daily site images and building information modelling, *Buildings*, vol. 9, no. 3, 2019.
- [14] Zeng S., Chen J., and Cho Y. K., User exemplar-based building element retrieval from raw point clouds using deep point-level features, *Autom. Constr.*, vol. 114, no. September 2019, p. 103159, 2020.
- [15] Li Y., Li W., Tang S., Darwish W., Hu Y., and Chen W., Automatic indoor as-built building information models generation by using low-cost RGB-D sensors, *Sensors (Switzerland)*, vol. 20, no. 1, pp. 1–21, 2020.
- [16] Bhadaniya P., Reja V. K., and Varghese K., Mixed Reality-Based Dataset Generation for Learning-Based Scan-to-BIM, in *International Conference on Pattern Recognition (ICPR 2021)*, 2021, vol. 12667, pp. 389–403.
- [17] Golparvar-Fard M., Peña-Mora F., and Savarese S., Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models, *J. Comput. Civ. Eng.*, vol. 29, no. 1, p. 04014025, Jan. 2015.