



FACE-VS-SEGMENT COMPARISON: A NEW METHOD FOR COMPARING AS-PLANNED AND AS-BUILT MODELS

Georgios Tsimpiskakis¹, Ekaterina Petrova¹, Elena Torta², and Pieter Pauwels¹

¹Department of the Built Environment, Eindhoven University of Technology, Eindhoven, The Netherlands

²Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract

Recently, new technologies such as Building Information Modeling (BIM) have significantly impacted the construction progress monitoring performance. However, mistakes during construction still occur. To address this challenge, an improved Scan-vs-BIM methodology is proposed, particularly focusing on a new method for comparing the as-planned and as-built model termed *Face-vs-Segment comparison*. The two models are compared at a building element surface level, which constitutes its novel part. This method incorporates semantic information, documents the magnitude and direction of deviations, and provides customized reporting. This approach can potentially reduce monitoring time, increase overall productivity, and improve decision-making processes.

Introduction

In recent decades, the world has witnessed an outburst in technological advancements, often referred to as the Fourth industrial revolution or Industry 4.0. These advancements primarily focusing on computer science, digitisation and the Internet of Things (IoT), and they have radically changed the everyday life (Forcael et al., 2020). Within the Architectural, Engineering and Construction (AEC) industry the term Construction 4.0 has been adopted to indicate such outburst. Emerging technologies such as sensors, laser scanners, drones, and Building information modeling (BIM) software, can now be utilised across all project phases—from design to asset management—to increase efficiency (Jiang et al., 2022; Wang and Kim, 2019). However, even with the implementation of these advanced tools, construction projects remain problematic, often characterised by low productivity (Casini, 2022). The inherent complexity of construction processes at various project phases frequently leads to errors, which can lead to low productivity and in turn can cause the projects to exceed their calculated costs and projected deadlines (Jiang et al., 2022).

The construction phase, like any other project phase, is vulnerable to errors. Well-designed project drawings, and well-planned activities can be poorly executed due to the complexities of the construction environment in which they are implemented (Jiang et al., 2022). Although the adoption of the BIM approach incorporates standardised

procedures, international exchange frameworks, and innovative software programs which enhance construction process coordination, mistakes can still arise (Abreu et al., 2023). Building elements such as walls, columns, beams, etc might diverge from their original as-planned state due to mistakes in project-drawings, poor communication and collaboration among stakeholders, lack of expertise, accidents, and other factors (Abreu et al., 2023; Wang and Kim, 2019). The inability to promptly and accurately capture errors during construction, identifying the nature of these errors, and efficiently update the as-planned BIM model (Wang and Kim, 2019) are key drivers for poor construction process (Kavaliauskas et al., 2022). Thus, efficient construction progress monitoring is essential for the successful execution of the construction projects (Graganiello et al., 2024).

Under this context, this paper aims to propose an improved construction progress monitoring methodology, by focusing specifically on the comparison between as-planned and as-built models of a building's indoor environment. The primary objective of this research is to introduce a new comparison method to compare these two models on a per-element basis. This will help in: understanding the nature of errors occurring during construction, customising error reporting, and supporting the decision-making process during corrective actions. Broadly, this study seeks to enhance the literature, by investigating how Scan-vs-BIM approach can be improved. The remainder of the paper is structured as follows: Section 2 presents a review of key related literature. Section 3 discusses the improved methodology and the proposed comparison method. Section 4 demonstrates the results of this method using a case study, and section 5 the conclusions.

Literature review

Construction progress monitoring (CPM) involves evaluating project activities to ensure that they align with their as-planned state, and to maintain project schedules and budgets (Graganiello et al., 2024; Krijnen and Beetz, 2017). The emergence of BIM has improved collaboration, communication, and decision-making among project stakeholders during the execution phase (Pauwels and Petrova, 2020; Badenko et al., 2019). Additionally, innovations such as Augmented reality (AR), Digital twins

(DT), laser scanners, and drones have initiated the revitalisation of construction progress monitoring (Casini, 2022). Moreover, Lean Construction practises, such as the Last Planning System (LPS) facilitate scheduling and work monitoring throughout the project life-cycle (Porwal et al., 2010). However, recent studies revealed that more than half of construction projects fall behind schedule, and nearly two-thirds of them experience cost overruns (Han et al., 2018). To conclude, there is a need for more sophisticated construction progress monitoring methodologies that exploit current technologies and increase overall project's efficiency (Graganiello et al., 2024).

To satisfy this need SCAN-vs-BIM approach was introduced by Bosché et al. (2015) to address the occurrence of errors during the construction phase. This innovative construction progress monitoring methodology facilitates site monitoring as it utilises advanced building representations such as, Point cloud and BIM models (Pučko et al., 2018). Point clouds are considered as as-built model, while BIM models serve as as-planned models. These two models, depicting the same building environment in different project phases, are compared to identify deviations, which indicate errors during construction. Scan-vs-BIM is a promising approach for error detection and can be integrated in current construction progress monitoring methodologies to improve overall project performance (Graganiello et al., 2024).

A point cloud is a collection of points captured using a laser scanner device, representing tangible objects in the building environment (Wang and Kim, 2019). Typically, each point contains information about the X, Y, and Z coordinates, RGB colour values, and is exported in various file formats such as OBJ, LAS, RCP (for Revit), and PLY (Dinis et al., 2020). Generally, the point cloud models capture a snapshot of the built environment in certain time during construction. In contrast, a BIM model provides a much richer representation of the building environment (Graganiello et al., 2024). It consists of 3D objects enriched with valuable information, offering a comprehensive depiction of project plans and drawings (Pauwels and Petrova, 2020). BIM is more than just a model; it represents a novel philosophy that standardises and revolutionises processes within construction projects. It introduces international standards, such as ISO 19650, Industry Foundation Classes (IFC), BIM Collaboration Format (BCF), Information Container for linked Document Delivery (ICDD), and others, to manage information throughout the building life cycle (Ashworth et al., 2022). Such a novel approach aims to enhance the communication, collaboration, and decision making process among project stakeholders, ultimately contributing to the successful achievement of project goals.

The literature on the Scan-vs-BIM approach presents a variety of methods for comparing BIM and point cloud models. One method, involves comparing the comprehensive point cloud model with the comprehensive BIM model (Graganiello et al., 2024; O'Keeffe et al., 2017).

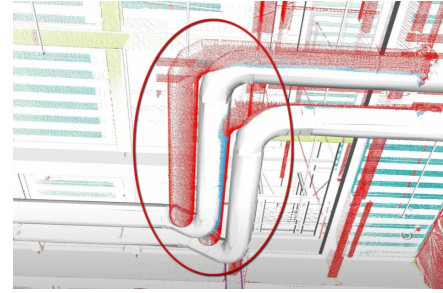


Figure 1: Example of duct differences within point cloud model and IFC, visualised by AutoCorrTM

This method presents deviations in the form of raw data. Specifically, deviations are visualised by colourising the points in the point cloud that exceed a predefined threshold (O'Keeffe et al., 2017) (Figure 1). While effective, this approach requires considerable user effort to manually analyse the properties of each deviation. Another comparison method involves point cloud-to-point comparison (Kavaliauskas et al., 2022). Now, the point cloud model is compared with specific points from the BIM model that can represent for instance individual columns. This allows for an efficient identification of whether these columns are missing from the as-built state of the project (Kavaliauskas et al., 2022). This method offers the advantage of documenting metadata for the building elements participate in the comparison process. The analysis of existing comparison methods in Scan-vs-BIM methodology highlights the need for further improvements.

Methodology

This research presents an improved version of the construction progress monitoring feedback loop system proposed by Jiang et al. (2022). The enhanced Scan-vs-BIM methodology includes the following steps: data acquisition, data pre-processing, comparison, progress reporting, while the loop closes by the execution of essential corrective actions (Figure 2). The corrective actions are utilised to update the initial as-planned model (BIM model) by incorporating the deviations identified through this methodology. However, the final two steps of this methodology remain conceptual and are reserved for future research.

The remaining steps of the methodology are analysed in detail to support the development of a new comparison method termed *Face-vs-Segment*. Particularly, this method investigates the efficiency of comparing the two models in a surface level. The analysis exploits only the visible outer surfaces of rectangular building elements that are perpendicular to a predefined Cartesian Coordinate System of the construction project. In this method, a building element surface (Figure 3a) from the BIM model termed as *Face*, is compared with the corresponding surface from the point cloud model, termed as *Segment* (Figure 3b). These two surfaces represent the same building element surface in reality, but are captured in different states—as-planned and as-built respectively. The *Face-*

vs-Segment method aims to precisely identify deviations between these states, document them in a more explanatory and structured manner, and finally facilitate decision-making process during construction phase.

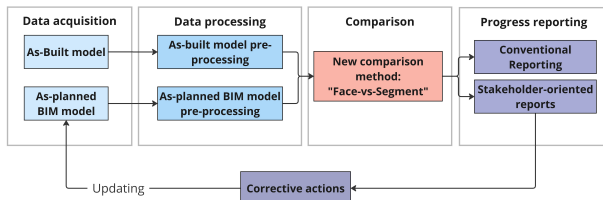


Figure 2: The conceptual construction progress monitoring methodology proposed by this research

The steps towards the development of the new comparison method are implemented in Python¹. The process begins with acquiring the as-built and as-planned models in specific file formats to be compatible for python-based analysis and visualisation purposes. Then, both models are processed to get prepared for the comparison step, where the new proposed *Face-vs-Segment* method is implemented. The deviations that are identified are in a form of displacements and are documented in meters (m) with a certain sign indicating their direction in terms of the specified coordinate system. By finding the direction of each deviation, information about the nature of the deviation arise (if it is for, change in building element’s dimensions, or catholic displacement, etc) which constitutes the novel part of this research.

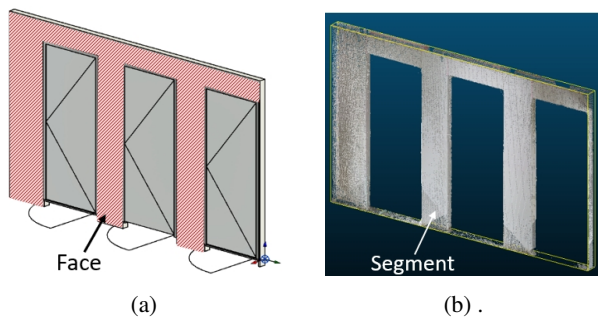


Figure 3: (a) A surface in the BIM model specifying a Face, (b) A surface in the Point cloud model specifying a Segment.

Data acquisition

Specifying the correct file format for the acquired datasets is critical for implementing an accurate Scan-vs-BIM methodology. For testing the proposed methodology, the as-built and as-planned models are acquired. These models always represent the same building environment of a building under construction but in different states, the as-planned model reflects the design phase, while the as-built model captures a state during the construction phase. The as-built model is captured using the terrestrial laser scanner (TLS), Leica BLK360, and is exported in a point cloud

¹The complete python algorithm: https://github.com/GTsimpiskakis/Improved_Scan-vs-BIM_methodology

model using .LAS file format. At the same time, the as-planned model of the same building environment is acquired in an .IFC file format. By utilizing these standardized file formats—commonly recognized in the literature—the efficiency of subsequent methodology steps, such as data processing and comparison, is significantly enhanced.

Point cloud model pre-processing

The pre-processing of the point cloud model aims to transform the raw data in editable and comparable form. Initially, the point cloud model is converted from .LAS to .PLY file format to ensure compatibility with software tools such as Blender 4.0 and Meshlab, which are utilized in this methodology. Then, Statistical Outlier Removal (SOR) method is applied to reduce arbitrarily located points that constitute outliers in the model. To further reduce the point cloud density, the voxelisation method is employed. After successful point cloud model size reduction, the computational power needed for point cloud processing is substantially reduced, which increases the Python algorithm efficiency. Thus, manageable point clouds can be easily decoupled into segments which constitutes the next step towards the point cloud pre-processing step.

Point cloud segmentation is an essential step towards point cloud processing for element-oriented Scan-vs-BIM matching. During this process, the uniform point cloud model is divided into 2D surfaces or *Segments*. A segment is a 2D plane which can also be considered as 3D due to its negligible thickness, typically ranging from 20mm to 40mm. The challenging task of the segmentation procedure lies in accurately decoupled into segments all the visible surfaces of each rectangular building element represented in the point cloud model.

While numerous segmentation algorithms exist in the literature, applying them in complex building environment is difficult task. After extended experimentation with these algorithms, a different segmentation method is proposed termed *RANSAC-DBSCAN segmentation method*. This method combines two established segmentation algorithms, RANSAC and DBSCAN. First, RANSAC clustering algorithm is applied to divide the model into segments. The user then intervenes, to categorize these RANSAC segments into *Correct*, or *Wrong* segments based on how accurately represent the reality. Wrong segments are further subdivided based on common point densities by utilizing the DBSCAN algorithm to better represent the reality, and finally label them as correct or wrong again. This combination produces more efficient results than applying each algorithm individually (See the orange segment in Figure 7a, and how it has been subdivided into two, after applying DBSCAN algorithm in Figure 7b). The RANSAC-DBSCAN method is demonstrated by the lines 8-30 of the pseudo-code of Algorithm 1. A significant limitation of the proposed method is its semi-automated nature, as requires substantial user intervention. More research needs to be done to minimise this user intervention. After suc-

Successful point cloud segmentation, the final step in point cloud model pre-processing is to transform the point cloud model into the same Cartesian Coordinate System as the IFC model. This transformation requires the origin point and orientation of the IFC model, and is applied by using regular translation and rotation operations.

IFC model pre-processing

To compare the IFC and point cloud models, both must be georeferenced within the same coordinate system. This research adopts a local Cartesian Coordinate System (CCS). The IFC model is loaded into Revit software and the user specifies the origin point of the cartesian coordinate system anywhere in the model by positioning the Revit's project base point (PBP). Subsequently, the coordinates of the four corner points of each visible building element face are identified based on this specified CCS, using Python. At the end, the IFC model and the point cloud model are ready for the comparison step, as the position of each building element face (from the IFC model) and each segment (from the point cloud model) is defined within the same Cartesian Coordinate System.

Face-vs-Segment comparison

The methodology developed so far aims to support a new approach of comparing the as-built and as-planned models in an element-oriented manner. Particularly, by comparing a *Face* with a *Segment*, more efficient identification of deviation between these two models can be identified. However, a face can only be compared with a segment if both represent the same surface of a rectangular building element of the measured environment. For this reason, we apply a two-step matching process between a face and a segment. This process is a complex task as the as-planned and as-built models can deviate from each other by infinite possible ways, reflecting various construction errors. To pass this obstacle, nine conditions and twelve rules representing the most common error cases are formulated based on experts' opinion. Conditions and rules govern the quantity of deviation between a face and a segment in order to be considered as matched. Thus, these two terms are used in this research to establish some norms in the two-step matching process between a face and a segment. A possible error case which is governed by condition 1 is illustrated in the schematic floor plan view in Figure 4.

During the first step of the two-step matching process, for each face, all its possible matching segments are grouped together. Nine conditions, are applied sequentially to narrow down the group of segments, until zero, one or two segments remain. If a Face is not matched with a segment, then it is labeled as *Missing*, and if is matched with one and only segment is labeled as *Existing*. If the Face is matched with two segments, the matching process continues to step two, by applying the rules to identify between the two remaining segments (See lines 31-50 of pseudo-code in Algorithm 1). The two-step matching process is primarily a part of a proof of a concept pipeline, thus further tests is

needed.

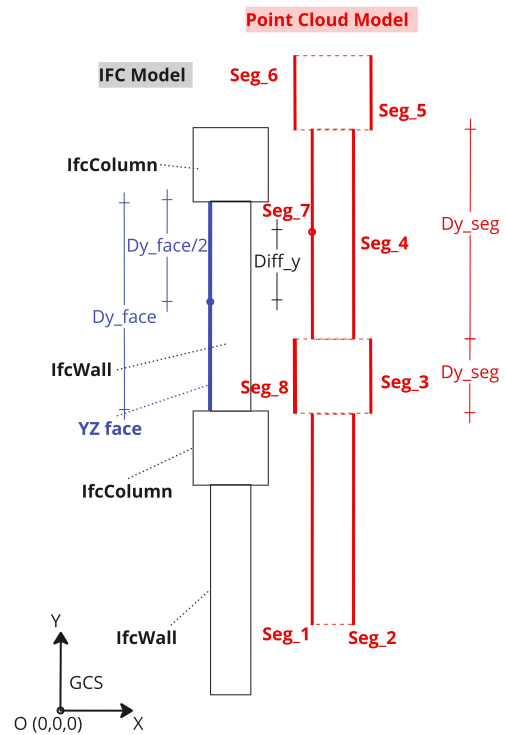


Figure 4: Condition 1: Difference in Axis Y. (a) Floor plane view of an IFC model (Black) and point cloud model (Red), initial group of possible matching segments (Seg₁ to Seg₈), and visualisation of $Diff_y$.

Figure 4 illustrates a possible error during construction. Even if it is obvious that YZ Face matches with Segment 7, it is not obvious for a computer. To automate this process in Python, nine conditions are established. For instance, Condition 1 showcases the matching process for differences between as-planned (faces) and as-built (segments) models in terms of Y axis. A maximum allowable distance is specified with Equation 1 which sets a threshold to prevent the YZ face from being matched with very distant segments that probably represent another face. Specifically, condition 1 indicates that *the distance between the centers of mass of the face and a segment on Axis Y must be less or equal than the Max_Diff_y , for a segment to remain in the possible matching segment group.* Max_Diff_y , constitute a limitation as the algorithm can identify deviations up until this limit. Finally, the segments that satisfy the condition 1 are Segment 7, 4, 8, and 3. For this new group of segments, Condition 2 is applied which is now referred to differences between faces and segments in Z axis. Then, condition 3 refers to norm that can distinguish if a segment represent a wall, column or beam surface. To conclude the remaining conditions are applied one by one (further explanations for conditions can be found in the work: Tsimpiskakis (2024)). Finally, after all conditions are applied, two segments remain in the initial possible matching segment group for the YZ Face, as demonstrated in Figure 5.

$$\text{Max_Diff}_y = \frac{d_{y_{\text{face}}}}{2} \quad (1)$$

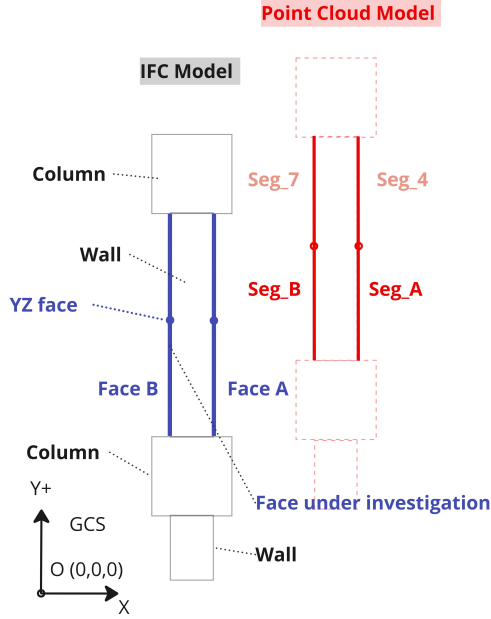


Figure 5: Remaining possible matching Segments for Face YZ after applying the nine conditions.

When two matching segments are identified for a face in the first step of the matching process, a challenge arises in distinguishing between them. This issue typically occurs in cuboid building elements with two of their surfaces in close proximity (e.g. a wall with a thickness of 10cm). These surfaces, share similar characteristics, which make them indistinguishable after the application of the conditions. The issue now is that both surfaces may be considered valid matching segments for a face. To address this problem, twelve rules are formulated, for segments located in X, Y, or Z axes. These rules will address all the possible arrangements or layouts between the face under investigation, the other twin face and the two segments (Face B, Face A, and Seg_A, and Seg_B respectively, of the example in Figure 5).

$$\left\{ \begin{array}{l} \text{if, } \text{Seg_A}_{\text{Loc}} \geq \text{Face_A}_{\text{Loc}} \text{ and} \\ \text{Seg_A}_{\text{Loc}} \geq \text{Face_B}_{\text{Loc}} \text{ and} \\ \text{Seg_B}_{\text{Loc}} \geq \text{Face_A}_{\text{Loc}} \text{ and} \\ \text{Seg_B}_{\text{Loc}} \geq \text{Face_B}_{\text{Loc}} \text{ and} \\ \text{Face_A}_{\text{Loc}} \geq \text{Face_B}_{\text{Loc}} \end{array} \right\} \text{ Then: } \text{Seg_B} \implies \text{Face_B} \quad (2)$$

Continuing the previous example, Figure 5 demonstrates the application of the first rule. Segments 4, and 7 are now referred to as Segment A, and Segment B, respectively, with Segment A always positioned at an average location greater than Segment B, along the common x-axis. In this particular example, Face B is investigated, and its location along axis-x is less than that of Face A. It is evident that Face B should be matched with Segment B. To translate this in a machine-readable form, five inequalities are for-

mulated in Equation 2. If these inequalities are satisfied, the algorithm matches Face B with Segment B. Otherwise, the process proceeds to the next rule, which accounts for different arrangements between these two faces and two segments. Finally, after all the rules have been applied (see Tsimpiskakis (2024)), it is secured that the face under investigation is correctly matched with the correct segment.

Results

To validate the above method, this work includes a case study on a complex high-rise building under construction in Eindhoven, the Netherlands. Currently under construction, the tower is expected to be completed by the end of 2025. As of April 2024, construction progress was approximately in one-third of the completion. The as-built state of a portion of the building's 4th floor was captured (Figure 6b) using a TLS BLK360, resulting in a .LAS point cloud model. At the same time, the as-planned state of the same area was

Algorithm 1 Segmentation and Matching Workflow

```

1: Input: PointCloudModel as PLY, asPlannedModel as IFC
2: segmentedModelRANSAC ← ApplyRANSAC(PLY)
3: segmentedModelDBSCAN ← ApplyDBSCAN(PLY)
4: DisplayResults(segmentedModelRANSAC, segmentedModelDBSCAN)
5: preferredAlgorithm ← GetUserDecision()
6: if preferredAlgorithm is RANSAC or DBSCAN then
7:   finalSegmModel ← preferredAlgorithm
8: else
9:   ▷ RANSAC-DBSCAN segmentation process applies
10:  segmentedModelRANSAC ← ApplyRANSAC(PLY)
11:  CorrectSegments ← []
12:  WrongSegments ← []
13:  for all seg in segmentedModelRANSAC do
14:    userDecision ← ClassifySegm(seg)
15:    if userDecision is Correct then
16:      Add seg to CorrectSegments
17:    else
18:      Add seg to WrongSegments
19:    end if
20:  for all wSeg in WrongSegments do
21:    RefinedSegments ← ApplyDBSCAN(wSeg)
22:    for all rSeg in RefinedSegments do
23:      userDecision ← ClassifySegm(rSeg)
24:      if userDecision is Correct then
25:        Add rSeg to CorrectSegments
26:      end if
27:    end for
28:  end for
29:  finalSegmModel ← MergeSegments(CorrectSegments)
30: end if
31: for all face in IFC do
32:   matchSegGrp ← FindInitialMatchingSegments(face, finalSegmModel)
33:   for condID = 1 to 9 do
34:     matchSegGrp ← ApplyCondition(face, matchSegGrp, condID)
35:   end for
36:   segCount ← Count(matchSegGrp)
37:   if segCount == 0 then
38:     LabelFace(face, "Missing")
39:   else if segCount == 1 then
40:     LabelFace(face, "Existing")
41:     LinkFaceToSegment(face, matchSegGrp)
42:   else if segCount == 2 then
43:     for ruleID = 1 to 12 do
44:       matchSegGrp ← ApplyRule(face, matchSegGrp, ruleID)
45:     end for
46:     CorrectSegments ← SelectFinalSegment(matchSegGrp)
47:     LabelFace(face, "Existing")
48:     LinkFaceToSegment(face, CorrectSegments)
49:   end if
50: end for

```

obtained (Figure 6a) from the construction company in

.IFC format. Since the as-planned model of the building may be updated throughout the construction process, it is very important to compare its most recent version with the corresponding as-built state to ensure accurate results. Moreover, before applying the Face-vs-Segment comparison method (the third step in Figure 2), it is necessary to obtain reliable outcomes from the point cloud model segmentation and IFC model pre-processing steps.

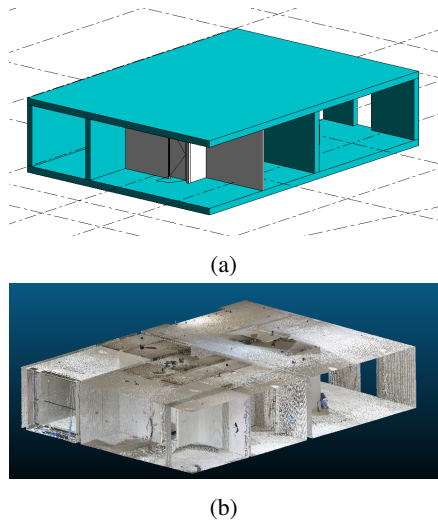


Figure 6: Case study data: (a) As-planned model (IFC model) (b) As-built model (Point cloud model).

Point cloud segmentation results

This research executes analysis into point cloud segmentation algorithms and proposes a new RANSAC-DBSCAN segmentation process. This process delivers significantly improved results compared to applying the RANSAC and DBSCAN algorithms individually. The RANSAC-DBSCAN process generates segments that more accurately represent the actual surfaces in the measured environment (Figure 7b). As demonstrated in Figure 7a, applying the RANSAC algorithm alone, does not yield correct segmentation results because some produced segments might represent two or more building element surfaces from the building environment (see orange segment in Figure 7a, when compared with the corresponding results of Figure 7b).

As-planned model pre-processing results

Secondly, this research contributes to the parsing of an IFC file, to enable the accurate identification of the coordinates of the four vertices of each visible building element face. These coordinates are determined in the local CCS specified by the user in Revit software. Consequently, IFC model pre-processing together with point cloud model segmentation contribute to determining the coordinates of segments and faces that accurately represent building element surfaces of the measured environment. This sets the foundation for applying the Face-vs-Segment comparison method.

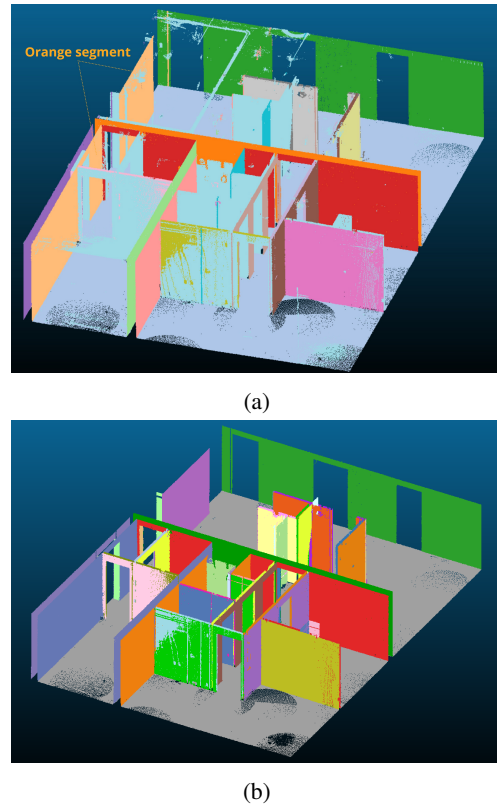


Figure 7: Resulting segmentation in case study point cloud model (a) RANSAC segmentation algorithm (b) RANSAC-DBSCAN segmentation process.

Face-vs-Segment comparison results

The Face-vs-Segment comparison method consists of two steps. The first step involves matching faces with Segments, while the second step deals with the comparison of each face with its corresponding matching segment (if a match exists) to identify discrepancies. So, first the matching process applies a series of conditions and rules until every building element face in the model is labeled as *Existing*, *Missing* or *Excluded*. This labeling is depending on whether it is matched with a segment, is not, or is excluded from the analysis, respectively. For each Existing face the process continues to the second step, namely to the Face and Segment comparison.

The comparison step is the final and most critical part of this research. The discrepancies between a face and its matching segment are identified in terms of displacements by documenting not only their magnitude but also their direction regarding the local CCS. Particularly, five types of deviations are documented as illustrated in a hypothetical wall example in Figure 8. This example focuses on faces located in the YZ plane of the CCS. Deviation 1 illustrates the displacement of the wall face along the X-axis. Deviations 2 and 3 correspond to the displacements of the Ymax and Ymin face edges, while Deviation 4 and 5 refer to the displacement of the corresponding face edges in Z axis. Not only the magnitude of the displacement, but also its direction is documented, by reporting its sign (+ or

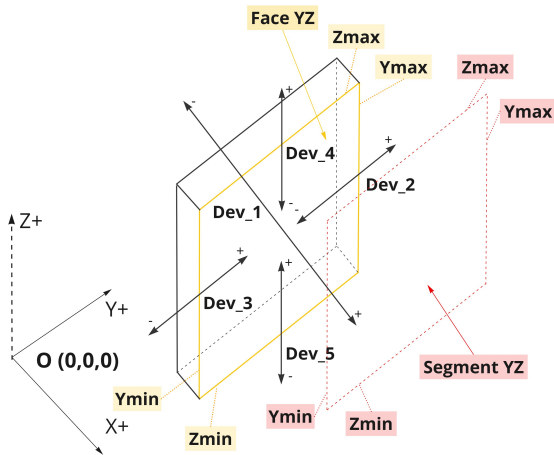


Figure 8: The five type of deviations that are investigated

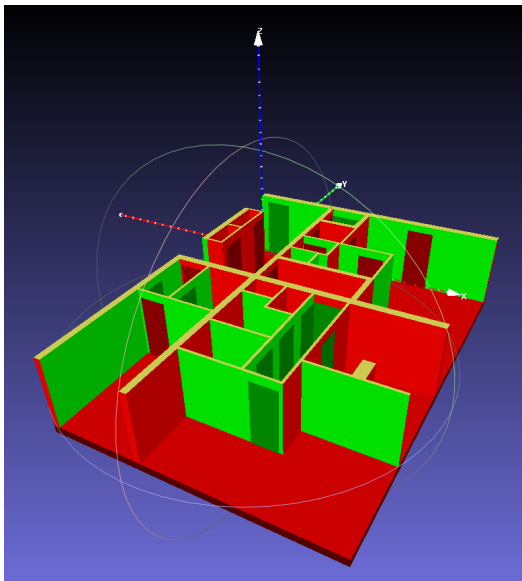


Figure 9: Final form of visualisation of comparison results. Correct faces (Green colour), Incorrect faces (Red colour).

-) along the corresponding axis. Some comparison results from case study data are demonstrated in Table 1. The same principle applies when a face is located in XZ, and XY planes of CCS. A tolerance of five centimeters is used to distinguish between acceptable and unacceptable deviations between faces and segments. When the deviation exceeds the tolerance the face is labeled as *Incorrect* and is coloured red, while otherwise it is labeled as *Correct* and is coloured green. For each face, the final comparison results are visualised in Figure 9 using Meshlab software by exporting an .OBJ file. Documenting the comparison results in both table format and colourful visualisation constitutes the final form of reporting proposed in this research. Of course, these results can form a baseline for implementing more automated updates or actions, which was not in the scope of this work.

Conclusions

This research demonstrates the feasibility of applying a detailed comparison method within the general methodological framework of the SCAN-vs-BIM approach (Figure 2). By examining existing comparison methods, potential gaps are identified and a new approach termed as Face-vs-Segment comparison is proposed. This approach aims to improve the detection, reporting and visualisation of discrepancies between as-planned and as-built state of indoor building elements under construction. Moreover, a new RANSAC-DBSCAN segmentation process is proposed, enabling the efficient segmentation of a point cloud representing an indoor building environment. This advancement facilitates the effective implementation of the Face-vs-Segment comparison method.

This new comparison method offers more effective error reporting. The discrepancies identified through the comparison are documented both quantitatively and qualitatively. Quantitative reporting includes documenting the magnitude and direction of a building element displacement, while qualitative reporting incorporates metadata, and other information about each face in final results. The combination of the quantity of deviation together with metadata (i.e. GUID, tag, IFC type, etc) from the associated IFC element in which the face belong (Table 1) constitute important contributions in the SCAN-vs-BIM framework. Furthermore, by visualising the comparison results with different colours in a model (.OBJ file) facilitates potential updating and error-handling by the end user. However the proposed Face-vs-Segment method comes with several limitations deriving from the application of conditions and rules. In conclusion, the proposed comparison method contributes in a reliable identification of mistakes during construction, which with further research, this method can be automated by applying it in robotics and digital twin applications.

Acknowledgments

The authors would like to thank Mr. Selahattin Dulger, as well as Mr. Dimitrios Koumaris and Mr. Daniel Escobar Valdivieso, members of the Twin Lab at TUE, for their significant support in this research.

References

- Abreu, N., Pinto, A., Matos, A., and Pires, M. (2023). Procedural Point Cloud Modelling in Scan-to-BIM and Scan-vs-BIM Applications: A Review. *ISPRS International Journal of Geo-Information*, 12(7):260. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.
- Ashworth, S., Dillinger, M., and Körkemeyer, K. (2022). BIM guidance to optimise the operational phase: defining information requirements based on ISO 19650. *Facilities*, 41(5/6):337–356. Publisher: Emerald Publishing Limited.
- Badenko, V., Fedotov, A., Zotov, D., Lytkin, S., Volgin, D.,

ID (1)	IfcType (2)	GUID (3)	Tag (4)	Dev_X (m) (5)	Dev_Y_max (m) (6)	Dev_Y_min (m) (7)	Dev_Z_max (m) (8)	Dev_Z_min (m) (9)	State (10)
1	IfcWall	...	364352	-	-	-	-	-	Excluded
1	IfcWall	...	364352	-	-	-	-	-	Excluded
2	IfcWall	...	364370	-0.015	0.013	0.012	-0.041	0.004	Correct
2	IfcWall	...	364370	-0.006	0.014	0.013	-0.041	0.003	Correct
3	IfcWall	...	364626	0.022	0.001	0.003	-0.043	0.05	Incorrect
3	IfcWall	...	364626	0.122	0.032	0.073	-0.043	-0.05	Incorrect
4	IfcWall	...	364665	-	-	-	-	-	Missing
4	IfcWall	...	364665	-	-	-	-	-	Missing
5	IfcOpeningElement	...	364373	-	0.015	0.03	-0.02	-	Correct
5	IfcOpeningElement	...	364373	-	0.015	0.03	-0.02	-	Correct
6	IfcOpeningElement	...	364609	-	-	-	-	-	Excluded
6	IfcOpeningElement	...	364609	-	-	-	-	-	Excluded
7	IfcOpeningElement	...	364646	-	-	-	-	-	Excluded
7	IfcOpeningElement	...	364646	-	-	-	-	-	Excluded

Table 1: Final results regarding testing the improved SCAN-vs-BIM approach with the case study data regarding elements on YZ plane of the CCS

- Garg, R. D., and Liu, M. (2019). Scan-to-bim methodology adapted for different application. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-5/W2:1–7.
- Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., and 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. *Automation in Construction*, 49:201–213.
- Casini, M. (2022). *Construction 4.0: Advanced technology, tools and materials for the digital transformation of the construction industry*. Woodhead Publishing.
- Dinis, F. M., Sanhudo, L., Martins, J. P., and Ramos, N. M. M. (2020). Improving project communication in the architecture, engineering and construction industry: Coupling virtual reality and laser scanning. *Journal of Building Engineering*, 30:101287.
- Forcael, E., Ferrari, I., Opazo-Vega, A., and Pulido-Arcas, J. A. (2020). *Construction 4.0: A Literature Review. Sustainability*, 12(22):9755. Number: 22 Publisher: Multidisciplinary Digital Publishing Institute.
- Graganiello, C., Salzano, A., Asprone, D., Nicoletta, M., and Zitiello, E. P. (2024). Scan-vs-bim approach for construction site monitoring optimization. In *International Conference of Ar. Tec.(Scientific Society of Architectural Engineering)*, pages 367–379. Springer.
- Han, K., Degol, J., and Golparvar-Fard, M. (2018). Geometry- and Appearance-Based Reasoning of Construction Progress Monitoring. *Journal of Construction Engineering and Management*, 144(2):04017110. Publisher: American Society of Civil Engineers.
- Jiang, Z., Shen, X., Ibrahimkhil, M., Barati, K., and Linke, J. (2022). Scan-vs-bim for real-time progress monitoring of bridge construction project. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10:97–104.
- Kavaliauskas, P., Fernandez, J. B., McGuinness, K., and Jurelionis, A. (2022). Automation of Construction Progress Monitoring by Integrating 3D Point Cloud Data with an IFC-Based BIM Model. *Buildings*, 12(10):1754. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.
- Krijnen, T. and Beetz, J. (2017). An IFC schema extension and binary serialization format to efficiently integrate point cloud data into building models. *Advanced Engineering Informatics*, 33:473–490.
- O’Keeffe, S., Hyland, N., Brodie, S., and Dore, C. (2017). *The BIM & Scan® Platform: A Cloud-Based Cyber-Physical System for Automated Solutions Utilising Real & Virtual Worlds*.
- Pauwels, P. and Petrova, E. A. (2020). *Information in construction*.
- Porwal, V., Fernandez-Solis, J., Lavy, S., and Rybkowski, Z. (2010). Last planner system implementation challenges. pages 548–556.
- Pučko, Z., Šuman, N., and Rebolj, D. (2018). Automated continuous construction progress monitoring using multiple workplace real time 3D scans. *Advanced Engineering Informatics*, 38:27–40.
- Tsimpiskakis, G. (2024). Face-vs-segment comparison: a new method for comparing as-planned and as-built models. Master’s thesis, Technical University of Eindhoven.
- Wang, Q. and Kim, M.-K. (2019). Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, 39:306–319.