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

AUTOMATING CONSTRUCTION SITE GUARDRAIL SAFETY INSPECTION USING BIM AND POINT CLOUD SEGMENTATION

SUBMITTED: January 2026
PUBLISHED: June 2026
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2026.030

Vimal Bharathi, Ph.D. Candidate
Department of Civil and Mechanical Engineering, Technical University of Denmark (DTU)
vvebh@dtu.dk

Jochen Teizer, Professor
Department of Civil and Mechanical Engineering, Technical University of Denmark (DTU)
teizer@dtu.dk

SUMMARY: *Due to the dynamic nature of construction sites, constant installation and removal of safety equipment is a required practice. This to date requires manual, thus infrequent and still time-consuming inspections to ensure the safety measures are correctly in place when needed. This paper introduces a novel end-to-end pipeline that integrates SafeConAI, UAV-collected point cloud data, and 4D BIM to automate safety inspections, overcoming the limitations of fragmented prior approaches by enabling real-time deviation mapping and comprehensive safety monitoring. This approach aims to bridge the gap between as-built data integration with autonomous navigation by providing location-aware compliance checking. Autonomous ground or aerial vehicles collect point clouds, and a segmentation model is trained to detect and segment safety guardrails and other essential building elements. The system is built and tested in a laboratory environment first by creating a typical fall-from-height protection case. The information generated from the point cloud segmentation is then compared with the original BIM to monitor and point out deviations, and finally provides an analysis report in a user-friendly SafeBIM. The method has been evaluated on several test cases from laboratory to real construction site settings. The results from this work indicate a promising method that can assist practitioners in (semi-) automating construction site safety inspection and making workplaces safer.*

KEYWORDS: *point cloud segmentation, compliance control, fall-from-height hazards, protective guardrails, IFC building elements, SafeBIM, unmanned aerial vehicles.*

REFERENCE: *Bharathi, V. & Teizer, J. (2026). Automating construction site guardrail safety inspection using BIM and point cloud segmentation. Journal of Information Technology in Construction (ITcon), 31, pg. 674-698. https://doi.org/10.36680/j.itcon.2026.030*

1. INTRODUCTION

Construction sites are dynamic workspaces where complex tasks often involve frequent repositioning of labor, materials, and machinery (Soygaonkar & Bhangale, 2014). This creates a significant challenge in maintaining up-to-date safety measures and ensuring that safety equipment aligns with the required standards. Unlike static environments in fabrication, construction sites rarely remain the same for even two consecutive days, and their configurations continuously evolve as the building process progresses (Pinto et al., 2011). In 2022, falls from height accounted for approximately 81% of fatal slips, trips, and falls and 20% of nonfatal falls across all industries, with construction workers comprising nearly half (49%) of fatal occupational slips, trips, and falls (BLS Stats, 2023; BLS Chart, 2023). In particular, roofing contractors, residential building construction, and commercial or institutional building construction reported among the highest fatality rates in 2022 compared to prior years and other sectors (CPWR Data, 2023). Notably, 70% of fatal falls in construction involved workers at companies with fewer than 10 employees.

The continuity in work task performance requires frequent updates to safety protocols and equipment placement, making safety management an ongoing and complex challenge for responsible humans. This variability also means that safety plans can quickly become outdated. Ensuring that hazard mitigation measures, such as guardrails and fall prevention systems, always comply with the current safety plan is an even greater challenge. Factors such as the repetitive nature of manual inspections, unplanned removal of safety equipment during work, workers' failure to reinstall it, and the rapid emergence of new hazardous conditions make these potentially lifesaving tasks more difficult. Despite the best efforts of entire safety teams, manual processes are prone to errors, particularly when workers are fatigued or under time pressure. Therefore, frequent safety inspections are critical to ensure that safety protocols are followed and that appropriate safety equipment is installed correctly. Yet, these inspections can also be labor-intensive, time-consuming, and highly susceptible to human error. Given their importance in preventing accidents, inspection tasks can not be neglected or compromised (Xu et al., 2023). This makes it imperative to explore more reliable and efficient methods, such as automation, to address these challenges effectively.

One promising avenue is the use of advanced technologies, such as Autonomous Ground Vehicles (AGVs) and Unmanned Aerial Vehicles (UAVs) for automated data collection (Siebert & Teizer, 2014). These tools can capture comprehensive, up-to-date 3D representations of construction sites through point cloud data, providing an accurate overview of current conditions. By eliminating reliance on manual data collection, these technologies streamline the inspection process, enabling faster and more reliable assessments of site safety. At the same time, Building Information Modeling (BIM) serves as a digital representation of the as-planned construction environment, supporting planning and compliance checking. However, temporary safety elements such as guardrails are typically not explicitly modeled in conventional BIM workflows, as BIM primarily focuses on permanent structural components. This creates a gap between planned safety requirements and actual on-site conditions.

Figure 1: Guardrail protection rule in Denmark (HÅNDBOGEN, 2025).

To address this gap, recent research has explored the use of point cloud data and machine learning techniques to identify construction elements and safety features. These approaches allow automated detection of objects such as walls, floors, debris, and guardrails from scanned data. The detected elements can then be compared with BIM models to assess compliance with safety plans. However, most existing studies focus on individual components of the problem, such as guardrail detection or BIM-based rule checking, without integrating them into a unified workflow.

While prior studies have explored individual components such as point cloud segmentation (Yin et al., 2021) or BIM-based safety checks (Johansen et al., 2022), and others have focused specifically on guardrail detection (Bharathi et al., 2024; Johansen et al., 2021), these approaches remain fragmented. In contrast, this research incorporates guardrail detection as one component within a broader end-to-end pipeline that integrates an enhanced SafeConAI algorithm, UAV-collected data, and 4D BIM for automated safety inspections. This enables real-time deviation mapping of safety-critical elements, particularly guardrails (Figure 1), between the as-planned BIM model and the as-built site conditions, offering a comprehensive solution tailored to the dynamic nature of construction sites.

The research questions addressed in this work are:

- How can point cloud segmentation and BIM integration automate construction site guardrail safety inspections?
- What is the effectiveness of using a point cloud segmentation model in the detection of guardrails under varying realistic workplace conditions?
- What are the practical benefits, and how far can potential issues be overcome by implementing automated safety inspections in dynamic and complex construction environments?

The paper is organized as follows. The literature is reviewed in the next section. The research methodology is described in Section 3, and the process implementation is presented in Section 4. Section 5 presents the analysis and statistical report of the process on a real construction site. And finally, conclusions and future direction of the study are mentioned in Section 6.

2. BACKGROUND

For this work, a detailed literature review was conducted to understand current trends in the use of BIM, point clouds, and AI techniques to improve site safety and compliance. We investigated various areas, such as BIM to enhance safety, recent point cloud registration algorithms, AI models for point cloud segmentation, and a method for as-built data integration.

2.1 The role of BIM in enhancing safety and compliance

BIM serves as a centralized source of essential building data, encompassing much more than just geometry. It supports compliance checks across various critical areas, such as fire safety simulations (Sun & Turkan, 2020), accessibility standards (Eastman et al., 2009), and visibility analyses (Li et al., 2022). Within the realm of safety, BIM has become a cornerstone for rule-checking processes. Researchers have extensively explored how BIM and its geometric data can be utilized to identify hazards, including fall risks, which can be mitigated directly through geometric analysis of BIM elements (Johansen et al., 2022; Qi et al., 2014). Incorporating safety considerations into BIM models has been approached in several ways. Some methods focus on embedding safety guidance within the design process, guided by formalized input from construction workers (Qi et al., 2011). Others adopt a semantic strategy, sharing identified hazards with designers or safety managers for appropriate intervention (Kim et al., 2016; Jin et al., 2019). Another methodology involves directly integrating mitigation measures into the BIM model, enabling automated identification and extraction of safety equipment locations and geometries (Johansen et al., 2023; Zhang et al., 2013). In Johansen et al. (2023), automated approaches for identifying and extracting safety equipment in BIM models achieved 100% completeness, soundness, and spatial correctness on the simpler benchmark Model A (a low-complexity high-rise building scenario) and 91–96% across all three metrics on the more challenging benchmark Model B (an edge-case model designed to test algorithm robustness). This direct integration streamlines safety processes, focusing on specific hazard types and enabling efficient management of mitigation strategies. By embedding safety measures into the BIM model, the digital representation can guide the automation of safety-related tasks, such as inspections. Knowing the type and location of the expected safety equipment allows for more systematic and reliable monitoring. Furthermore, Teizer et al. (2022) introduce the concept of a Digital Twin for Construction Safety (DTCS), extending BIM capabilities by integrating real-time sensor data and AI-driven analytics to create a dynamic safety management system. The DTCS framework includes modules for hazard prevention through design, real-time risk monitoring, and continuous safety performance improvement, transforming BIM into a proactive tool for construction site safety.

2.2 Advancements in point cloud registration algorithms

To effectively track construction progress, acquiring accurate as-built data is essential. A highly reliable approach to obtaining this data is to use laser scanners, which generate point clouds. Point cloud registration is crucial role in most point cloud processing tasks. Typically, global registration techniques rely on geometric feature matching, whereas local registration methods generally operate without using features. The study by Fontana et al. (2021) examined various point cloud registration algorithms using Generalized Iterative Closest Point (G-ICP), Normal Distribution Transform (NDT), and Probabilistic Point Cloud Registration (PPCR) for local registration, as well as TEASER++, Fast Global Registration, and Random Sample Consensus (RANSAC) for global registration, with TEASER++, G-ICP, and PPCR demonstrating strong performance (median error as low as 0.11 across diverse benchmarks) while NDT showed higher variance (median error 0.72). Additionally, Ge (2017) focused on sampling-based algorithms that utilize RANSAC to address challenges in coarse registration during automated alignment of two-point clouds without markers from building scenes. A widely recognized traditional approach is the ICP algorithm, which is often employed to align two point clouds or match a point cloud to a building information model (Liu et al., 2021; Masood et al., 2020; Zhang et al., 2021). For precise registration, a coarse alignment must first be completed to obtain the necessary transformation parameters. ICP assumes that the point clouds are roughly aligned and then iteratively refines the alignment by finding the best rigid transformation. During this process, the algorithm identifies correspondences by calculating the shortest distances between points, improving the alignment with each iteration. In Bouaziz et al. (2013), a new version of the ICP algorithm was introduced that enhances robustness through sparsity optimization, achieving significantly lower RMSE values (e.g., as low as 4.8e-4) in the presence of substantial outliers and noise compared to standard ICP. The Sparse ICP technique addresses issues by using sparsity-inducing norms, which improves robustness to noise and outliers, although this can come at the cost of significant performance degradation (Mavridis et al., 2015).

2.3 Guardrail detection using image-based and point cloud methods

Recent advancements in construction technology have leveraged point cloud data for enhancing scaffolding safety and efficiency, as demonstrated by Kim et al. (2023), who developed a method to transform point clouds into images for analyzing scaffold joint safety, offering a robust framework for 3D-to-2D safety assessments and achieving 82.1% accuracy in scaffold joint safety analysis while successfully localizing unsafe joints on the 3D point cloud data. Similarly, Khan et al. (2021) exploration of deep convolutional neural networks for monitoring mobile scaffolds highlights the potential of integrating advanced computational techniques with safety rule correlations, paving the way for real-time construction site oversight.

In the context of detecting safety equipment, particularly guardrails, previous research has focused on both data collection and compliance verification in as-built environments. Similar to methods used for general construction progress monitoring, these studies employ either image-based approaches or point clouds collected from various sources. For instance, one image-based study utilizes two-dimensional (2D) images combined with Convolutional Neural Networks (CNN) for detection tasks (Kolar et al., 2018). This approach leverages transfer learning, allowing pre-trained models to be adapted for specific tasks, such as guardrail detection. The Visual Geometry Group (VGG-16) model (Simonyan & Zisserman, 2014) was retrained on a dataset of 4.000 augmented images and an additional 4.000 real-world images sourced from construction sites and online platforms like Google. This method achieved an impressive 96.5% detection rate for a specific type of factory-produced guardrails. Bharathi et al. (2024) show that guardrail detection can be performed from images under very different weather conditions using deep learning and domain adaptation techniques. However, this work lacks information on the depth of the guardrails. Johansen et al. (2021) detect guardrails using point clouds and some traditional computer vision methods. However, this work lacks information on geo-referencing the detections back to the BIM model. The current work is being performed to address issues encountered in the previous work and to build a robust safety compliance checking system. Another notable study also employs image-based techniques but relies on video footage captured by UAVs rather than traditional static images (Li et al., 2022). Keyframes are automatically extracted from the video, and a topology-based method is applied to detect guardrail posts. Detection is refined by introducing constraints, such as defining spatial relationships and distances between rail posts and floors. Unlike deep learning approaches, this method uses traditional machine learning techniques, reducing the size of the training dataset required to achieve reliable results.

While these studies provide valuable insights into guardrail detection, they address isolated components of the

safety inspection process. In contrast, this work proposes a novel end-to-end pipeline that integrates advanced point cloud segmentation (ResPointNet++), UAV-based data collection, and a BIM-based SafeCONAI_v2 framework. This holistic approach not only addresses depth and geo-referencing limitations but also enables automated deviation mapping back to the IFC model, offering a unified solution for construction safety monitoring.

2.4 BIM for data integration and automated navigation

BIM serves as a versatile platform for both data integration, fusing heterogeneous sources such as point clouds and sensors into as-built models, and also automated navigation, enabling semantic-aware path planning and localization in dynamic construction environments. By parameterizing as-built geometries from real-time scans and extracting navigable networks from IFC schemas, BIM bridges the gap between static design models and operational realities, reducing discrepancies that hinder autonomy and compliance monitoring. In practice, BIM models rarely include temporary safety elements such as guardrails, as they are often installed and removed dynamically during construction. Recent works highlight BIM's dual utility, particularly in integrating LiDAR and UAV data for accurate parameterization while supporting graph-based routing for robots and vehicles.

For navigation, BIM provides semantic maps that optimize paths while incorporating integrated data for adaptability. Bahreini et al. (2024) extended this with the OBRNIT ontology, enriching IFC models for navigable graphs that support A*-replanning via RGB-D mismatches, reducing errors by 25% and enabling asset localization. Husemann et al. (2025) demonstrated BIM-to-simulation pipelines in Unreal Engine 5, aligning LiDAR scans for sub-centimeter localization and task-oriented paths (e.g., marking), with 95% success in 100 m² trials.

Emerging integrated works leverage BIM for concurrent data assimilation and navigation. Akhavian et al. (2025) introduced BIM2RDT, an agentic framework that converts static BIM into digital twins via 4D point-cloud integration, generating voxelized spaces for potential-field routing that adapts to hazards, yielding 20-35% energy savings. Amani & Akhavian (2024) fused BIM-derived potentials with multi-heuristic A* and NLP constraints to generate collision-free paths, incorporating as-built updates from scans to enforce buffers, accelerating planning by 40%. Braga et al. (2025) presented a modular platform that queries IFC for spatial relations, integrates semantic UIs with ROS for real-time obstacle mapping, and reduces manual oversight in navigation by 50%.

Collectively, these advancements illustrate BIM's efficacy in parameterizing as-built data for fidelity and in guiding navigation for efficiency. However, they typically use broad scanning, inefficient for transient safety elements like guardrails. The proposed pipeline in this work addresses this by pre-annotating geo-referenced hazard locations in SafeBIM, localizing inspections, allowing autonomous agents to condition paths using IFCSpaceIDs containing the safety element(guardrails), and eliminating redundant routes via ResPointNet++ validated scans, enabling proactive, compliance-centric workflows.

3. METHODOLOGY

The pipeline of processes illustrated in Figure 2 outlines a thorough and organized method for comparing as-planned and as-built construction data, with the primary objective of ensuring that the constructed elements conform precisely to the original design specifications. This methodology leverages the capabilities of BIM and point cloud data to achieve high accuracy in construction verification. The process commences with the creation and utilization of the "as-planned" 4D BIM model, a comprehensive digital representation that encapsulates not only the three-dimensional spatial details of the structure but also integrates time-sequence information, effectively adding a fourth dimension. This 4D BIM model serves as the foundational reference point, embodying the envisioned final structure as per the architectural and engineering design plans, complete with all intended components and their respective installation timelines.

From this initial 4D BIM model, a derivative known as the "SafeBIM" is generated. While the precise definition of "SafeBIM" may vary depending on project-specific requirements, it is reasonable to infer that this iteration of the model is enriched with critical safety features, most notably guardrails integrated into the design to ensure compliance with safety standards and regulations. The BIM model is automatically enriched with safety elements using the SafeConAI_v2 algorithm. These safety elements are paramount, as they protect workers and future occupants by mitigating risks such as falls from elevated platforms. The SafeBIM model, typically formatted in the widely adopted Industry Foundation Classes (IFC) standard, provides a structured and interoperable framework for representing building data. However, to facilitate a direct, meaningful comparison with the as-built conditions,

this IFC-based BIM model must be converted into a point cloud format. Point clouds are used directly instead of derived surface models to preserve fine-grained geometric details and avoid information loss during surface reconstruction, particularly in cluttered construction environments. This conversion process involves discretizing the continuous surfaces and volumes of the BIM model into a dense collection of individual data points, each defined by x, y, and z coordinates in a 3D space, thereby rendering it compatible with the as-built point cloud data.

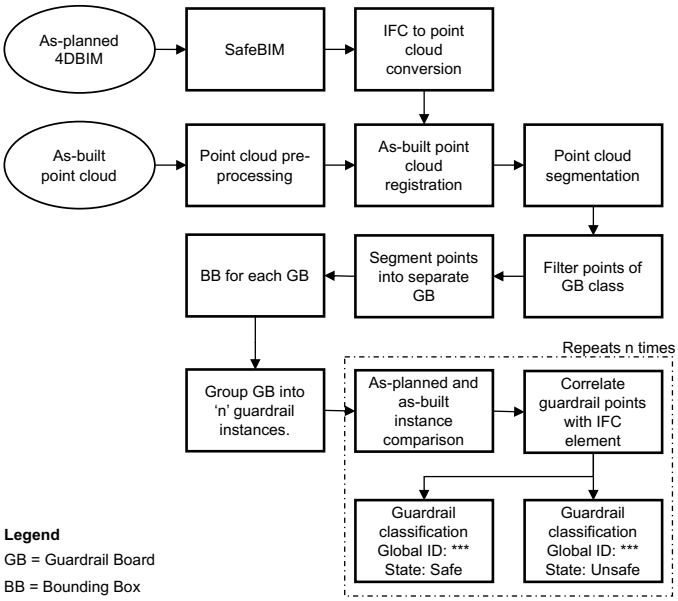


Figure 2: Pipeline of processes performed for the automated safety monitoring system.

In parallel, the as-built point cloud is generated to capture the real-world, physical state of the constructed structure at a given point in time. This 3D representation is typically acquired through advanced on-site data collection techniques such as Laser Detection and Ranging (LaDAR), Light Detection and Ranging (LiDAR), as well as photo- or videogrammatry, where photographic images are processed to reconstruct a 3D model. In this work, the Ouster OS0 sensor is used, which employs a 128-channel digital LiDAR array to emit rapid laser pulses and measure time-of-flight distances to surfaces, producing millions of high-resolution points that collectively form a detailed point cloud of the constructed environment. However, the raw as-built point cloud data, while rich in detail, is often imperfect and requires extensive pre-processing to ensure its reliability for subsequent analysis. This pre-processing phase involves several labor-intensive steps, including the manual removal of extraneous elements such as human figures, vegetation (e.g., trees), and other temporary objects captured during scanning but not part of the permanent structure. Additionally, sparse points that are too isolated to contribute meaningfully to the model, as well as outliers, which may result from measurement errors or reflections, are meticulously filtered out to enhance the dataset's integrity. The point cloud is aligned to a standardized coordinate system during this stage, a critical step to ensure that it can be accurately overlaid with the as-planned point cloud later in the process.

Once both the as-planned and as-built point clouds have been adequately prepared, the next step in the pipeline involves their registration, or alignment, within a unified coordinate system. Registration is a complex procedure that typically employs algorithms such as the ICP method to minimize the spatial discrepancies between the two datasets. By aligning the point clouds, corresponding features such as walls, floors, and guardrails can be directly compared, enabling a precise assessment of how closely the as-built structure mirrors the as-planned design. Following successful registration, the aligned point cloud data undergoes segmentation, a process that partitions the vast collection of points into distinct, manageable subsets or classes based on the structural elements they represent. For example, points corresponding to floors, walls, columns, and guardrails are grouped into separate categories. This segmentation is often achieved using machine learning techniques (Qi et al., 2017; Hu et al., 2019; Thomas et al., 2019; Zhao et al., 2020; Yin et al., 2021) or geometric algorithms (Johansen et al., 2021; Kanan et al., 2018) that identify patterns, shapes, or spatial relationships within the data. By breaking the point cloud into these discrete segments, the analysis becomes more focused and efficient, allowing evaluators to zoom in on specific elements of interest, such as the guardrails, which are critical to safety.

With the point clouds segmented, the pipeline proceeds to identify deviations between the as-planned and as-built models. This step entails a detailed comparison to detect any disparities, such as construction errors, missing components, or deviations beyond acceptable design tolerances. For instance, a wall built slightly off its intended position or a guardrail omitted entirely would be flagged during this phase. The deviations are quantified by comparing the positions, orientations, or densities of points between the two models, measured in millimeters (mm) or centimeters (cm). Once these discrepancies are identified, the results are thoroughly analyzed to verify the presence and integrity of critical safety elements. A key focus here is the "guardrails" class in the as-built model, which is scrutinized to confirm that guardrails are present at the specified locations and configurations in the SafeBIM. Missing or improperly installed guardrails could pose significant safety hazards, making this verification step a cornerstone of the safety inspection process.

To predict this deviation analysis, a custom Python algorithm has been developed, as detailed in the Appendix. Leveraging the segmentation results from Section 3.2 of the referenced methodology, the script first isolates points classified as guardrail points in the as-built point cloud. These points are then further processed to delineate individual guardrail boards, a task that requires distinguishing discrete structural units within the broader guardrail class. According to the safety criteria established for this project, a guardrail assembly is deemed safe only if it comprises exactly three boards at any given location, a standard that likely reflects safety regulations (B 100, 2021), as shown in Figure 1 or engineering requirements for stability and fall protection. To facilitate this assessment, the algorithm calculates the bounding box for each identified guardrail board, determining the minimum and maximum x, y, and z coordinates that enclose the points constituting that board. This bounding box serves as a spatial envelope, encapsulating the board's position and extent in 3D space.

With the bounding box data for each guardrail board in the as-built models, the script executes an iterative algorithm to perform a one-to-one comparison. For every guardrail board specified in the as-planned model, the algorithm searches the as-built model to determine whether an equivalent board exists at or near the same location, within a predefined vicinity or tolerance threshold (e.g., a few centimeters). If a corresponding board is found, the associated set of points is labeled as "safe," indicating compliance with the design intent. Conversely, if no matching board is detected or if the board's position, size, or orientation deviates significantly, the points are labeled as "unsafe," signaling a potential construction defect or safety violation. This procedure, visually represented in Figure 2, exemplifies a robust framework for automating quality control and safety assessments in construction projects. Pinpointing discrepancies with precision to the “SafetyGuardrail ID” and “IFCSpace ID” of the faulty guardrail location would empower project managers to address issues proactively, ensuring that essential design elements, particularly safety-critical features like guardrails, are implemented as intended.

The automation of safety inspections in this work hinges on two pivotal components: the SafeBIM (Section 3.1), which provides authoritative reference for safety features, and point cloud segmentation, which enables the granular analysis of as-built conditions. If the analysis reveals missing elements, such as guardrails absent from the as-built model, these discrepancies are mapped back to the SafeBIM IFC model using a custom-built Python algorithm as mentioned in Section 4.4 and shown in the Appendix. This reverse-mapping process involves correlating the deviant point cloud data with the corresponding entities in the BIM model, using specific guardrail instances tagged with unique identifiers. By doing so, project managers gain a clear and actionable understanding of which design elements were not constructed as planned, down to the exact locations and components affected. Armed with this information, they can prioritize corrective actions, whether that involves installing missing guardrails, adjusting misaligned structures, or revising construction processes to prevent future deviations. Ultimately, this detailed pipeline not only enhances the accuracy of construction verification but also reinforces safety and quality standards, aligning the built environment with its digital model (as-planned SafeBIM).

3.1 Enhanced version of SafeBIM generation pipeline

The term “SafeBIM” refers to the output generated by using a BIM model after the placement of safety elements (such as guardrails) and the various spaces, namely movement space, fall space, and hazard space. Johansen et al. (2022) introduced the concept of SafeBIM, in which the authors used the BIM model to generate safety guardrails for simple IFC elements. But for the industry standard, IFCs need more robust guardrail placement. In this work, the preliminary version of SafeConAI was used and modified to address complex building and construction structures, and subsequently updated to version 2, which was ultimately used. The comparison between the older and modified versions of SafeConAI is shown in Figure 3. The modified, more advanced version is better in the

placement of guardrails, eliminating duplicate placements, removing false cases of guardrail placement, and moreover, reducing the inference time for the SafeBIM generation. The process time for the BIM model seen in Figure 3 in the previous version is approximately 290 seconds, which is much worse than the enhanced version 2, which takes approximately 17 seconds (≈94% reduction). This update is critical as it significantly improves the accuracy and efficiency of guardrail placement, which directly affects the reliability of subsequent safety analysis. The number of guardrail entities generated by SafeConAI serves as the basis for comparison with the as-built guardrail installation. In the previous version, the number of unwanted and falsely generated guardrail elements has been reduced by 75% in the new version, which significantly improves the safety analysis produced by this work in the later sections.

The modified SafeBIM generation by version 2 has been described in detail in this section. An example of a BIM model is shown in Figure 4a, which will be used to demonstrate the steps involved in SafeConAI to generate safety guardrails. The SafeBIM generation rests on four different stages, as shown in Figure 4b-e.

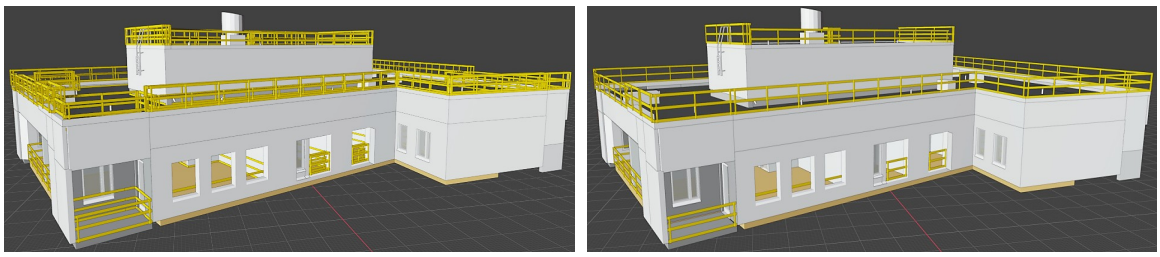


Figure 1: Visual comparison between SafeConAI's previous version and the modified version 2.

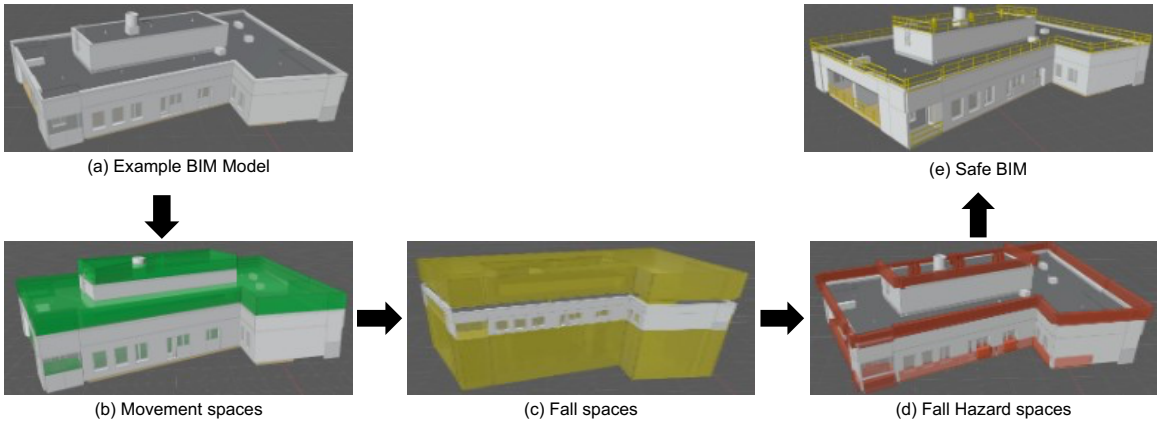


Figure 2: Steps involved in generating SafeBIM for the seventh and eighth floors of the building.

3.1.1 Extracting movement space

This step extracts walkable spaces from BIM objects (e.g., floors, walls, etc.) meeting geometric criteria such as height, width, and connectivity based on predefined thresholds from the configuration. The process begins by initializing with a BIM model and an optional object list, defaulting to all model objects if unspecified. Objects are sorted by their maximum z-coordinate (height) for a bottom-up approach. The configuration parameters (Figure 5) defining spatial constraints are based on analyzing the safety regulations in Denmark (HÅNDBOGEN, 2025), Germany (B 100 2021), and the United States (OSHA, 2019):

- Walkable height = 2m;
- Minimum walk width is 0.56m, and
- Maximum walkable slope = 0.8 equivalent to ~39 degrees.

The analysis iterates over objects as a potential base surface, retrieving their upward-facing surface and minimum height z_{on}. A nested loop checks all other objects for obstructions: if another object's minimum height z_{down} satisfies

0.0m ≤ z_{down} - z_{on} ≤ 2m, its downward surface is subtracted from the base surface, removing obstructed areas like ceilings or overhead structures. The result buffers inward by -0.56m, and the valid space is recorded as walkable, linked to its obstructing objects. If the buffered shape becomes invalid (None) but the original shape persists, it's flagged for a connectivity check.

The connectivity step splits shapes into polygons, comparing average z (z_1) to the existing spaces z_0, and if -0.1 m ≤ z_1 - z_0 ≤ 0.5m is true, the overlap test expands the footprint 0.2m (minimum object size) for intersection. If an overlap exists, the polygon is deemed accessible. Accessible polygons are grouped and added to the walkable space list, with a navigable network via vertical clearance (0.0m ≤ z_{down} - z_{on} ≤ 2m), buffering, and a 0.2m overlap check. These conditions ensure spaces are at least 2m tall, 0.56m wide, and connected within a 0.5m vertical step limit as seen in Figure 4b. The process assumes static geometry and nuances that stairs require further refinements.

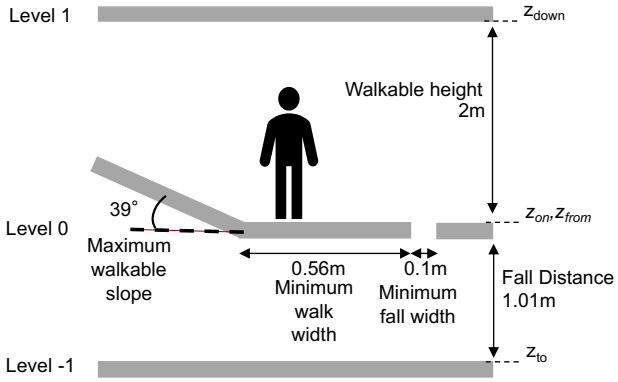


Figure 3: Configuration parameters for the SafeBIM calculation.

3.1.2 Extracting fall space

This step detects fall spaces (vertical drops exceeding the threshold) from BIM objects using top-down processing. The process starts with a BIM model and an optional object list, defaulting to all model objects if none is provided. Objects are sorted by maximum z-coordinate (height) and reversed for a top-down approach. Configuration parameters defining spatial constraints:

- Fall distance = 1.01m;
- Minimum fall width = 0.1m; and
- An artificial "top" object is created, positioned above the model's highest point by the model's z-range plus 3m, ensuring all potential falls are bounded from above.

The analysis iterates over each object, starting with its downward surface and minimum height z_{from}. A nested loop examines all other objects to identify fall scenarios, considering both intra-object and inter-object falls. For intra-object falls (same object, matched by ifc_guid), if multiple surfaces exist, the downward surface intersects the upward surface. If the vertical gap z_{to} - z_{from} exceeds 1.01m, the intersection is refined with a 0.1 m buffer, translated downward by -(z_{from} - z_{to}) - 1.01m, and buffered again by 0.1m. If valid (not None), this space is recorded as a fall within the object.

For inter-object falls (different objects), if the lower object's maximum height zmax exceeds the upper object's minimum height zfrom, it's skipped. Otherwise, the downward surface of the upper object intersects the upward surface of the lower object. If the vertical drop z_{from} - z_{to} exceeds 1.01 m, the intersection is translated downward by -(z_{from} - z_{to}) - 1.01m and buffered by 0m. If the result and its bounding box are valid (not None), it's added as a fall space. The upper object's surface is then subtracted by the intersection, refined with a 0.1 m buffer, and discarded if it becomes None.

This step identifies fall risks by checking whether z_{to} - z_{from} > 1.01m (intra-object) or z_{from} - z_{to} > 1.01m (inter-object), using intersection, translation, and buffering to define spaces as seen in Figure 4c. It also assumes static geometry and a top-down perspective for completeness.

3.1.3 Extracting fall hazard space

This step identifies fall hazard spaces by analyzing the spatial relation between movement spaces and fall spaces. The process starts with lists of movement spaces and fall spaces, using configuration parameters such as:

- Coverable distance = 1m;
- Space height = 1m;
- Space width = 0.5m.

The analysis iterates over each movement space, extracting its footprint and bounding box heights: minimum ms_z_{low} and maximum ms_z_{high}. For each movement space, it examines all fall spaces, retrieving their footprints and bounding box heights: minimum fs_z_{low} and maximum fs_z_{high}. A fall hazard is identified if $fs\_z_{low} \leq ms\_z_{low}$, $ms\_z_{high} \leq fs\_z_{high}$, and $ms_z_{low} > 0$m hold true.

When these conditions are met, the shared edges between the movement space and fall space are computed. If shared edges exist (not empty), a candidate fall hazard space is created as a line string along these edges, with a fixed height of 1 m (space height), linked to the movement space (parent) and fall space (dependency). The candidate's bounding box is calculated with a height of 1m from ms_z_{low}, and its dimensions x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, and z_{max} are compared to existing fall hazard spaces. If all dimensions match within a 0.01m tolerance (e.g., abs ($candidate\_bb.x_{min}$ - $existing\_bb.x_{min}$) < 0.01), it's flagged as a duplicate and skipped. Non-duplicates are added to the fall hazard list, with their z-range (e.g., ms_z_{low} to [ms_z_{low} + 1m]) logged for verification (Figure 4d).

3.1.4 Placing safety guardrail

This step integrates fall protection by placing guardrails along identified fall hazard spaces. It processes hazard spaces to generate IFC-compliant guardrail elements, using precise geometric placement based on configuration parameters, namely:

- Railing height = 1m;
- Board height = 0.1m;
- Board thickness = 0.04m;
- Number of boards = 3;
- Maximum distance between poles = 1.5m; and
- Pole thickness = 0.05m.

It handles two input types: a list of hazard spaces or a single space with time-based changes. For each hazard space, the process extracts a line string representing the fall edge. If it is a valid LineString, it iterates over its constituent lines. Each line, if a Line object, triggers guardrail creation with a default or specified time range (tadd, tremove) and an optional parent element.

Guardrail construction starts with a line's vector (vec_L) and length (mag_L). If mag_L < 0.1m, no guardrail is created (minimum length 10 cm, below minimum object size = 0.2 m but kept as original). Normalized vectors (*vec_0*, *vec_90*, *vec_180*, *vec_270*) define directions along and perpendicular to the line. Poles are spaced based on mag_L: if [mag_L > 2 x 1.5m] (maximum distance between poles), points are interpolated using (mag_L / 1.5m) segments; otherwise, only endpoints are used. For each point:

- First pole: Offset by -0.05m (negative pole thickness, into the slab) and 0.1m from the start;
- Last pole: Offset by -0.05m and -0.1m from the end;
- Middle poles: Offset by -0.05m; and
- Pole corners (e.g., lower left) are calculated using half pole thickness (0.025m), extruded to railing height = 1m.

Boards span the full line length, placed at heights from 0 to 0.9m (1m - 0.1m) in number of boards = 3 steps (0m, 0.45m, 0.9m). Corners use half board thickness = 0.02m, extruded to board height = 0.1m. Both poles and boards are positioned via IFC extrusion placements, centered by their mean coordinates, with poles and boards colored

yellow as seen in Figure 4e.

3.2 Point cloud segmentation

We have used point cloud segmentation models trained on custom datasets to detect all the points that belong to guardrails in various scenarios, with ground truth for the detection of safety guardrails. For the detection of guardrails to work with good accuracy, we trained the model with various scenarios. We have used seven different point clouds containing indoor and outdoor scenarios of guardrail placement. The training data also contains the SafeBIM point cloud labeled for the same outdoor location (as seen later in Section 4 in Figure 11a).

We trained the model with a sample dataset containing a mock guardrail built and placed in a real-world scenario. We collected seven different point clouds using the BLK360 Laser scanner, covering the guardrail scenarios shown in Figure 6. Then the dataset was labeled manually into six different classes: ceiling, debris, floor, guardrail board, guardrail pole, and wall.

For detecting guardrails in point clouds, this study adopts "ResPointNet++", which is the successor of "PointNet". The PointNet model achieves high performance on the benchmark dataset, namely ModelNet40. However, it lacks the ability to capture local context at different scales. "ResPointNet++" introduces a hierarchical feature-learning framework with improvements by introducing three layers, namely a sampling layer, a grouping layer, and a point net layer. The sampling layer downsamples the point cloud to reduce complexity and select key points, whereas the grouping layer groups the points into local neighborhoods for feature extraction and spatial context. The point layer used for local pattern learning is similar to "PointNet".

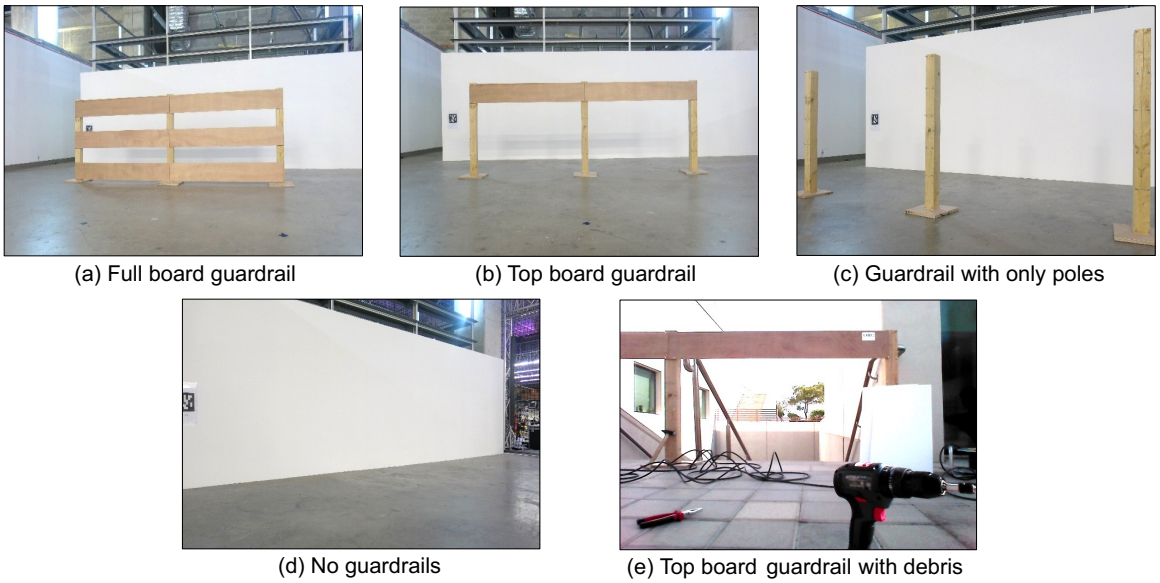


Figure 4: Different cases for guardrail placement include both safe and unsafe conditions.

We used the "ResPointNet++" model trained on the PSNet5 dataset, which contains around 80 million data point clouds collected from four different industrial scenes and covers 4,000 m2. The PSNet5 dataset was collected as part of the research in Yin et al. (2021), which contains objects such as pipes, pumps, tanks, I-shaped beams, and rectangular beams. This dataset is very similar to construction scenarios with multiple structures and objects. Hence, we performed transfer learning using the weights of ResPointNet++ trained on PSNet5 and further trained the model for our objective by using our guardrail dataset and fine-tuned. The custom data used for the training segmentation model includes both indoor (Figure 7a and b) and outdoor (Figure 7c and d) scenarios. To increase computational efficiency in training, the point clouds were spatially subsampled with a minimum distance between points of 0.02m. In order to train the point clouds, each point must have a set of input features for the model to learn. In this work, the points are trained with 6 sets of features, namely [x, y, z, n_1, n_2, n_3]. Here, x, y, z are the positional coordinates of the points, and n_1, n_2, n_3 are the normal vectors to each point. The normals are generated using surface approximation techniques called local surface models, meaning the normals are perpendicular to the

local surface on which the points are located. Since the point clouds are captured in different environments and under different conditions, the color [R G B] of the points is not used as a training feature.

We trained the model on the guardrail dataset for 300 epochs with various hyperparameters set for the dataset. The part Intersection Over Union (IOU) and mean IOU for the test data are in Table 1.

We evaluated the trained model with validation data and achieved results with great accuracy, as shown in the confusion matrix in Figure 8. The point cloud segmentation model gives the resulting segmentation of the points into separate classes, as seen in Figure 9. The validation data did not contain any ceiling class. The model predicts the guardrail board and guardrail pole with great accuracy, even with debris lying in front.

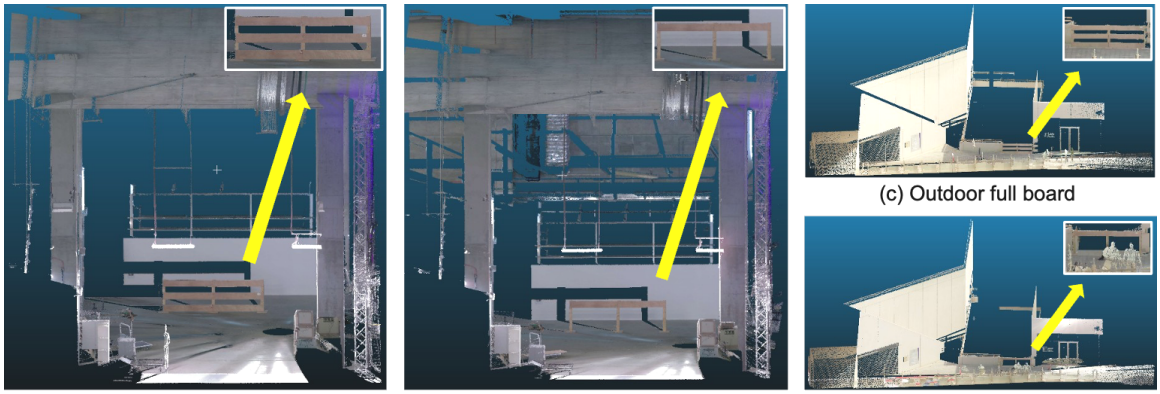


Figure 5: Point cloud training and test data. The inset image shows the guardrail scenario captured. The indoor point clouds were used as a training dataset, and the outdoor dataset was used as the validation dataset.

Table 1: IOU metrics for the test data at the end of training.

	Ceiling	Debris	Floor	Guardrail board	Guardrail pole	Wall
Part IOU	0.7762	0.4394	0.9993	0.7420	0.7134	0.9867
Mean IOU	0.7762					

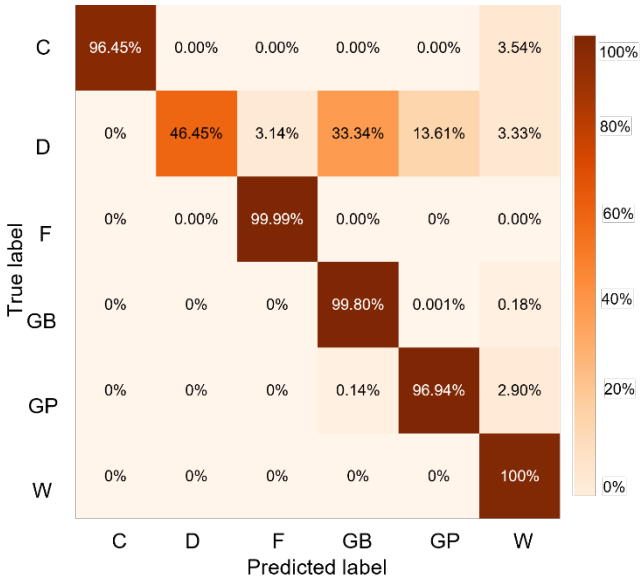


Figure 6: The confusion matrix shows the results of the model test on validation data. y-axis being the true label and x-axis being the predicted label. C - Ceiling, D- debris, F - Floor, GB - guardrail board, GP - guardrail pole, and W - wall.

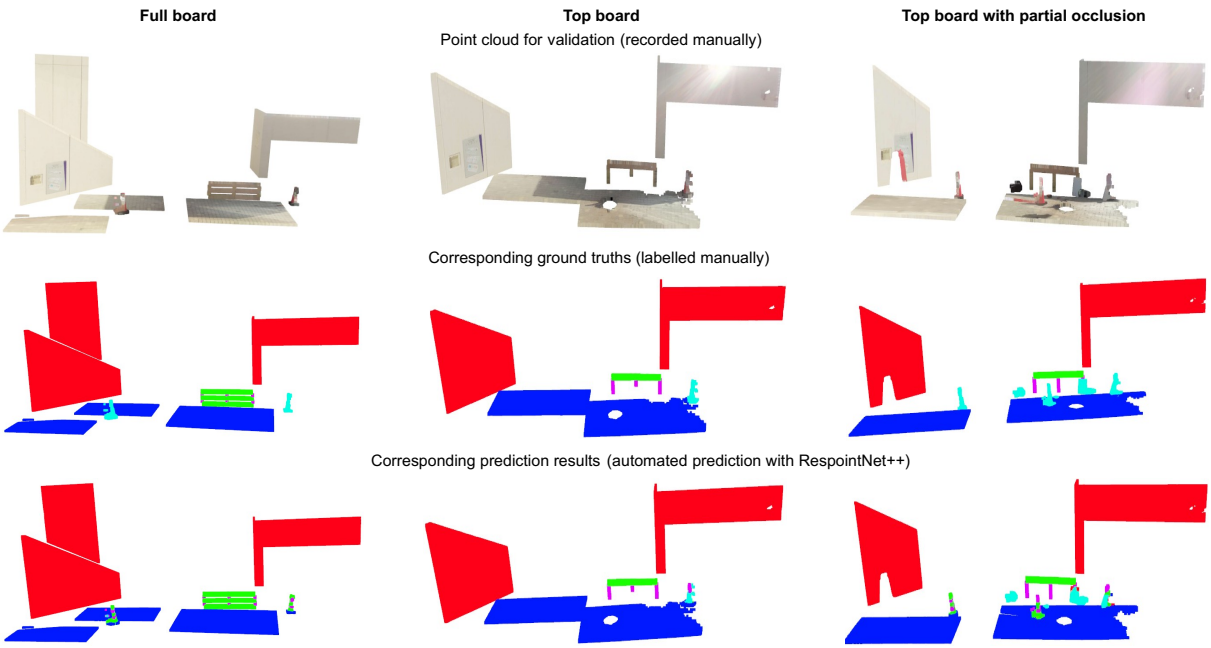


Figure 7: The predictions are shown for different classes of the guardrails' full board, top board, and top board with debris. Color values correspond to: Red – wall, Blue – floor, Green – guardrail board, Pink – guardrail pole, and Cyan – debris.

4. SAFETY INSPECTION SYSTEM IMPLEMENTATION

In this section, a more detailed explanation of the various steps involved in the pipeline, as shown in the previous section, is provided. The steps, such as SafeBIM generation, point cloud registration, point cloud segmentation, and devising the deviations between the as-built point cloud and the as-planned BIM model, are explained with some mockup scenarios.

4.1 SafeBIM prediction of staircase IFC model

For the automated safety monitoring pipeline seen in Figure 2, we need the BIM model for the area where the safety inspection needs to be conducted. The BIM model is needed for the pipeline to understand what the as-planned model of the site looks like to draw a comparison with the as-built situation. The as-planned BIM model needs all the safety guardrail elements installed by the Safety managers, which can then be used as ground truth for comparison. The BIM model (Figure 10a) is used to generate SafeBIM for the area with safety elements installed in the work placement of guardrails. This is done by using the SafeConAI_v2, which predicts all the fall-protection locations and installs the safety guardrails accordingly. The working of "SafeConAI_v2" was explained in Section 3.1. The results obtained for the area can be seen in Figure 10b. The results were validated through manual inspection by comparing the generated guardrail placements with safety regulations and expert verification.

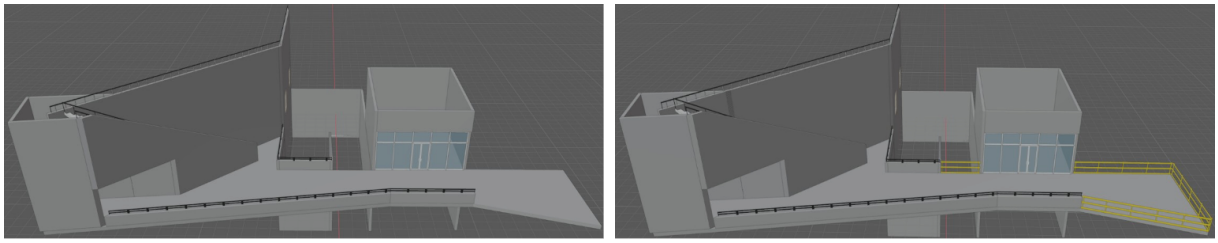


Figure 8: The result of using SafeConAI_v2 on to the model.

4.2 Point cloud registration

The next step in our pipeline is to align the point cloud captured in the Staircase area and the SafeBIM model to the same coordinate system. This is an essential step in this pipeline to extract the location of guardrails from both the as-built (captured point cloud) and the as-planned (Staircase BIM model), and perform analysis for the missing or faulty guardrail based on the location data. This step can be skipped if the SafeBIM and the point cloud collected are geo-referenced to a common coordinate system or standard. Hence, the data was not captured initially in this work; with geo-location, the point cloud registration must be performed additionally.

To align and register the captured point cloud with SafeBIM, we implemented automated point cloud registration. For this purpose, the SafeBIM was converted to a SafeBIM point cloud(ifcclouds, 2023). The raw BIM model in ".IFC" format is converted to a dense point cloud. The point cloud data is generated by extracting the meshes by "IFC-ENTITY-TYPE" and distributing points over each face using barycentric coordinates (Schneider & Eberly, 2003) and then using a weighted sampler algorithm (Efraimidis & Spirakis, 2006) to distribute points over mesh faces (triangles) extracted from IFC entities. The number of points per square meter of surface area (Num points) was set to be 10,000. All mesh-points are concatenated into a single point cloud. The point cloud extracted from this method is shown in Figure 11a.

Then, the as-built point cloud needs to be aligned with the converted as-planned SafeBIM model, as they are in different reference coordinates. There are several techniques for point cloud-to-point cloud registration. The method used in this work is the ICP registration method (Besl & McKay, 1992) using "CloudComPy" (CloudComPy, 2025). Here, the reference point cloud is the "as-planned" point cloud, and the point cloud to be aligned is the "as-built" point cloud. Once the point clouds are registered, they both lie on the same coordinate system, which makes it possible to compare the objects in the "as-planned" and "as-built" point clouds. The output of this can be seen in Figure 11c.

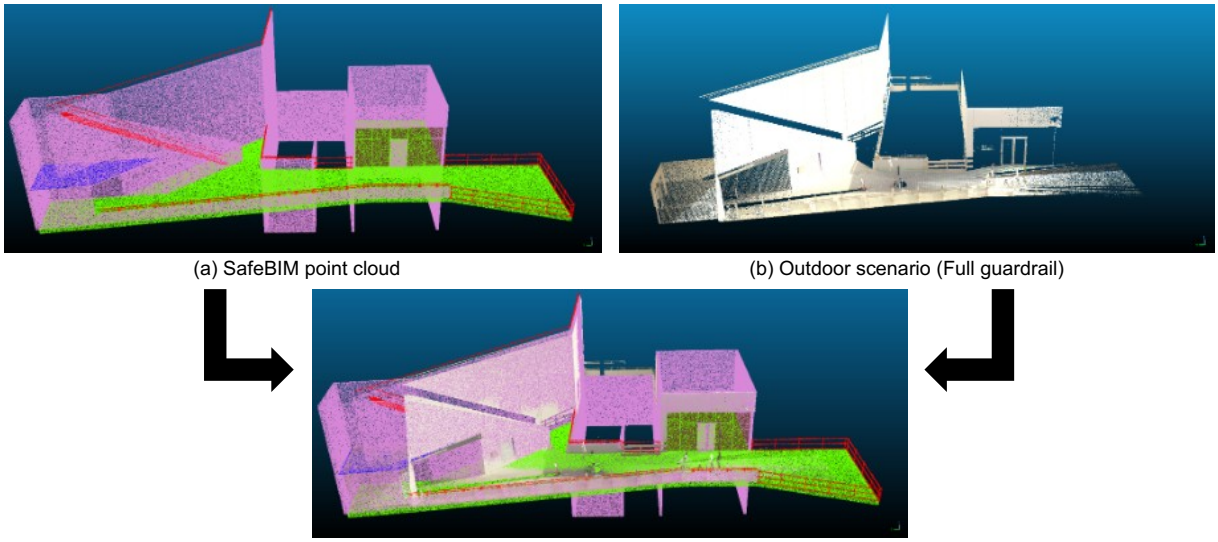


Figure 9: ICP registration of the as-built point cloud to the as-planned point cloud.

4.3 Point cloud segmentation

Once the as-built point clouds and the as-planned SafeBIM point cloud have been registered to a common coordinate system, the next step is to feed the as-built point clouds to the point cloud segmentation model that has been trained, as seen in Section 3.2. The model predicts different classes with a high accuracy of 0.77 mean IOU, as shown in Table 1 and Figure 8, and can be further processed to detect missing elements in point clouds (Figure 12a and b) by comparing it with SafeBIM. The model even predicts the guardrail pole and guardrail board with great precision, but since our focus is to count the number of guardrail boards in a given bounding box with x and y coordinates, we ignore the number of guardrail poles detected. The bounding box having 3 guardrail boards at different heights is determined as the safe class, and a bounding box that contains fewer than 3 guardrail boards is

classified as unsafe.

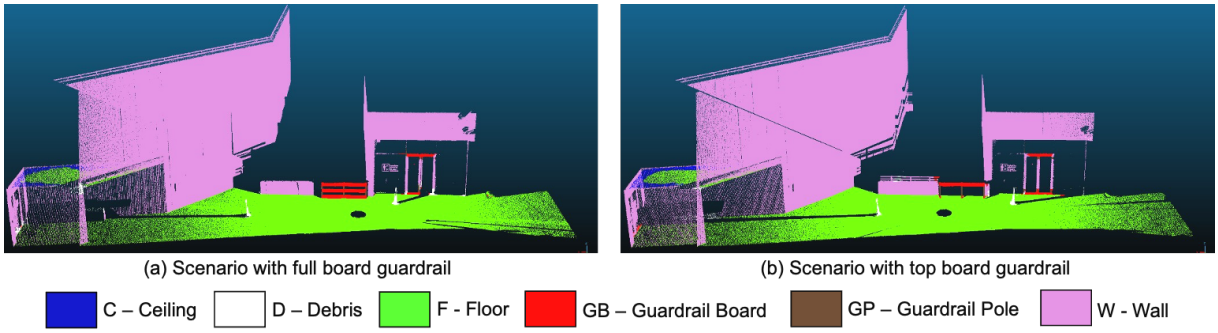


Figure 10: The class prediction of the point clouds using ResPointNet++ trained in Section 3.2.

4.4 Deviations between As-built and As-planned

To investigate discrepancies between the as-built environment and design specifications, a detailed pipeline was developed to compare guardrail configurations captured in point cloud data with those defined in a SafeBIM IFC model. The process begins with a point cloud file containing 3D coordinates and semantic labels, from which points classified as guardrail boards or poles are extracted. These points are segmented into individual instances using a density-based clustering algorithm (Ester et al., 1996), which groups spatially proximate points into clusters based on a maximum distance parameter and a minimum point threshold. For each cluster, an oriented bounding box is computed to define its spatial extent and orientation, and clusters are further analyzed to determine whether they represent boards or poles based on the predominant class within the cluster.

Next, we identify stacked guardrail configurations by grouping clusters that are vertically aligned, using tolerances for horizontal and vertical separation to distinguish multi-level systems (e.g., top rail, mid rail, toe board). The stacked group of guardrails can be seen in Figure 19. The safety of these stacked groups is assessed using two criteria: a group is considered safe if it contains exactly three instances, indicating a complete guardrail system, or if it has two instances with points other than floor, guardrail board, or guardrail pole detected directly below its bounding box footprint, suggesting additional protection like a ground surface. To detect such points, we define a bounding box for each group based on the minimum and maximum horizontal coordinates of its instances and the lowest vertical coordinate, then examine all points within this horizontal footprint that lie below the group. Groups not meeting these safety conditions are classified as unsafe, providing a clear metric for compliance with typical guardrail standards.

The pipeline then compares these point cloud-derived guardrail groups with guardrail elements specified in the SafeBIM IFC model. In the IFC file, guardrails are identified by their type (e.g., "IfcRailing" with a "safety guardrail" designation) or by names containing "guardrail." For each IFC guardrail, we compute a bounding box from its geometric representation and calculate its centroid. Matching is performed by comparing this centroid to the centroids of point cloud guardrail groups, considering a match valid if the distance between centroids is within a small threshold (e.g., 0.5m) or if the bounding boxes overlap, allowing for minor positional deviations between the as-built and as-planned data. This comparison determines whether each designated guardrail is present in the point cloud and retrieves its safety status from the earlier assessment. The result for the comparison of the guardrails between the SafeBIM and stacked groups can be seen in Figure 20.

To visualize deviations, we modify the SafeBIM IFC model by altering the color of guardrails based on their presence and safety classification. Guardrails that are either absent from the point cloud or classified as unsafe are highlighted in red. This is accomplished by accessing the geometric representation of each guardrail in the IFC file, updating existing style definitions to assign the red color (RGB: 255, 0, 0), and ensuring these changes are applied to the appropriate swept solid or body representation. If no guardrail groups are detected in the point cloud, indicating a complete mismatch with the design, all IFC guardrails are colored red to signal a critical deviation. The modified IFC is saved, preserving the original model while embedding as-built safety insights, as shown in Figure 21. This color-coded IFC output provides a visual representation of these deviations, enabling construction managers and safety inspectors to quickly identify and address issues. The algorithm for finding the deviation between the as-built point cloud and the as-planned SafeBIM has been attached to the Appendix.

5. ANALYSIS AND SAFETY REPORT

The analysis was done with real data from a real construction site in Finland, where the point cloud data was collected using an automated drone flight. Several cases of guardrail placements were collected for this study, with a total of 6 different combinations of guardrails. The entities where the guardrails were placed include windows and a door on the first floor of the residential building, which is approximately 10m high from the ground. Figure 13 shows the images from the data collection construction site.

(a) (b) (c)

Figure 11: Real residential building construction site in Finland.

Six different scenarios of guardrail placements have been captured for the analysis of the methodology in this work. The cases that were captured have been listed in Figure 14. The LiDAR data captured with the drone at the construction site includes all possible cases, such as all boards present, one or two boards missing, or a board installed diagonally. In the case of the door, the correct safety norm is to install all 3 horizontal guardrail boards (top, middle, and bottom), and for the window (top and middle boards), and any deviations from this are considered an unsafe condition.

	Window 1	Window 2	Window 3	Door 1
Case 1				
Case 2				
Case 3				
Case 4				
Case 5				
Case 6				

Figure 12: Cases for guardrail data collection using UAV LiDAR.

The UAV with LiDAR sensor, Figure 15, captures LiDAR data at 10 Hz from the start to the end of its planned flight path inside the construction site. All the frames of one UAV flight are combined to generate the complete point cloud for the site of data capture using an open-source LiDAR SLAM system (Liu, 2021). The LiDAR captures for 6 different cases can be seen in Figure 16 (viewer-based perspective mode) and Figure 17 (bubble-view mode).

(a) CATEC Cadrin: Indoor aerial robot (b) Ouster OS0 LIDAR

Figure 13: Indoor UAV and LiDAR sensor.

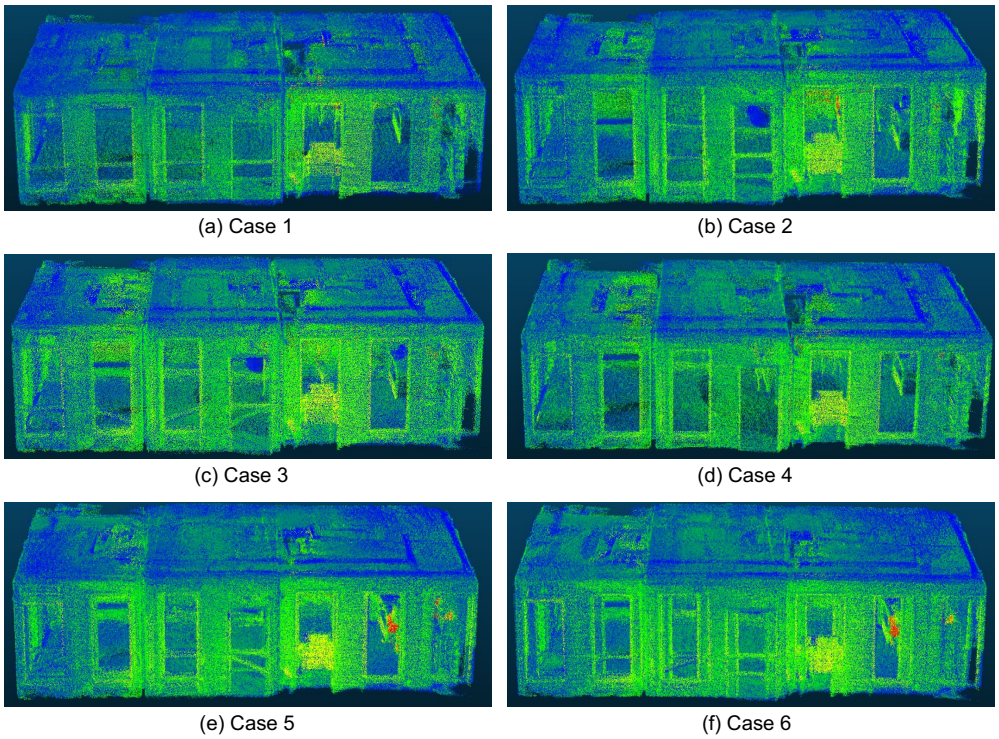


Figure 14: Full view from outside of the point cloud for 6 different scenarios of protective barrier placements.

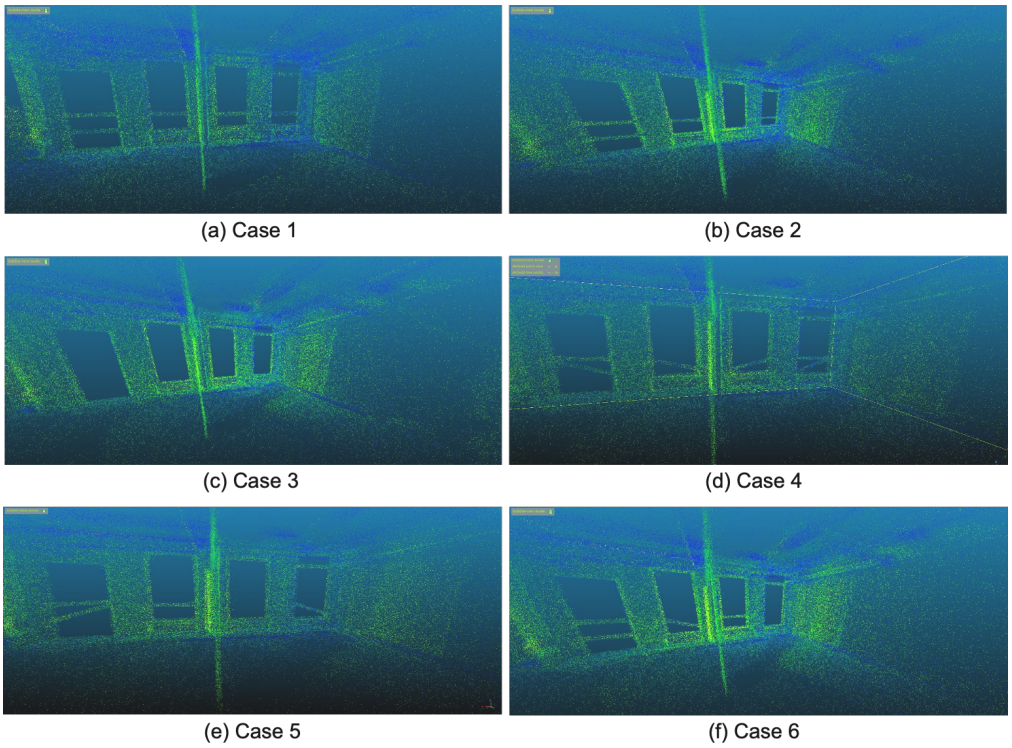
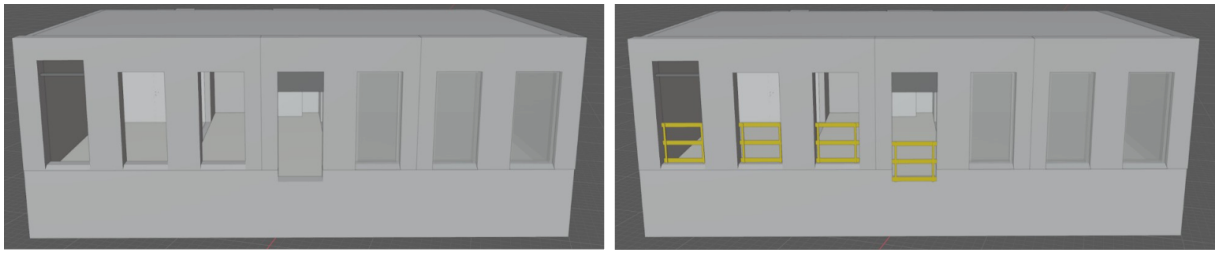


Figure 15: Bubble view from inside the point cloud for six different scenarios of protective barrier placements.

The SafeBIM is generated for the same test construction site using the pipeline mentioned in Section 3.1, as shown in Figure 18. Then, the point cloud segmentation model "ResPointNet++" trained in Section 3.2 is fine-tuned with a sample labeled dataset from this construction site. The transfer learning of the model on the particular use case

yielded better accuracy on segmentation results for 100 epochs, as shown in Table 2. Then this fine-tuned model has been used to segment all 6 scenarios and can be seen in Figure 19. The results from Figure 18 are used to isolate points that belong to the guardrail class, and guardrail boards that vertically align on top of each other are grouped together to form stacked groups of unique guardrails, which can be seen in Figure 20. After which, the guardrail elements in SafeBIM are spatially compared with the stacked guardrail groups, and the deviations are mapped back to SafeBIM as shown in Figure 21.

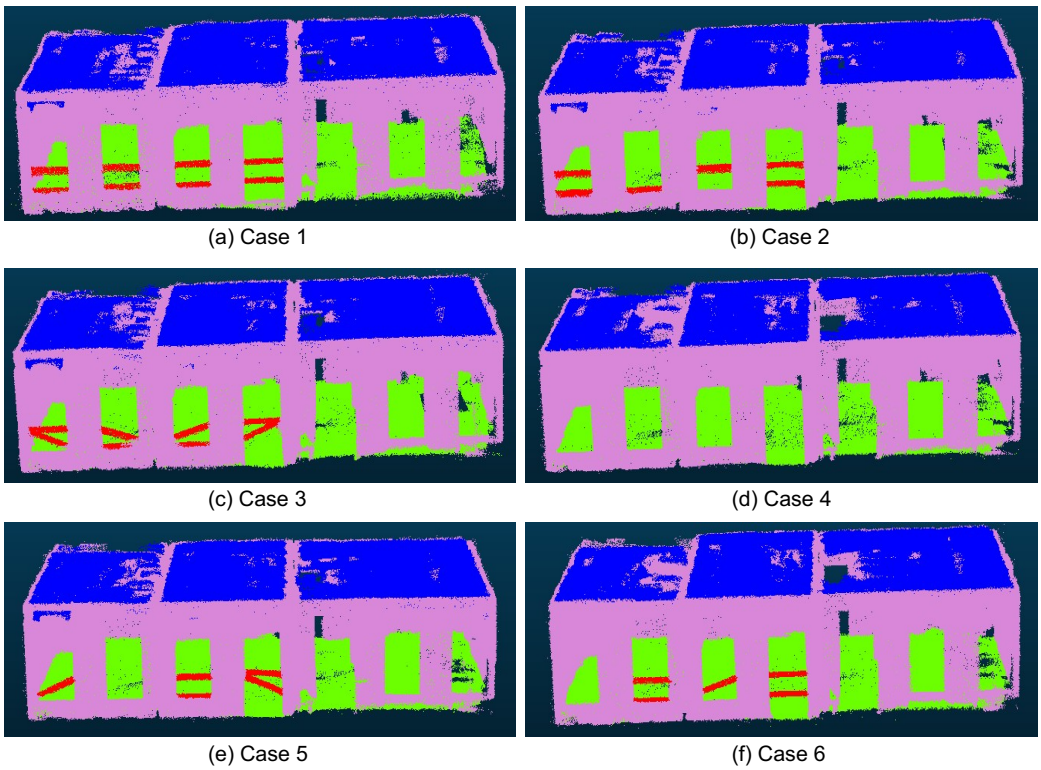


(a) BIM (b) SafeBIM

Figure 16: SafeBIM with the construction site. In total, 6 windows and 1 door, out of which 3 windows have already been installed.

Table 2: IOU metrics for the fine-tuned use case.

	Ceiling	**Debris**	**Floor**	**Guardrail board**	**Guardrail pole**	**Wall**
Part IOU	0.9563	0.9779	0.9986	0.9920	0.9546	0.9875
Mean IOU	0.9775					



(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

(e) Case 5 (f) Case 6

Figure 17: Point cloud segmentation results of all 6 cases.

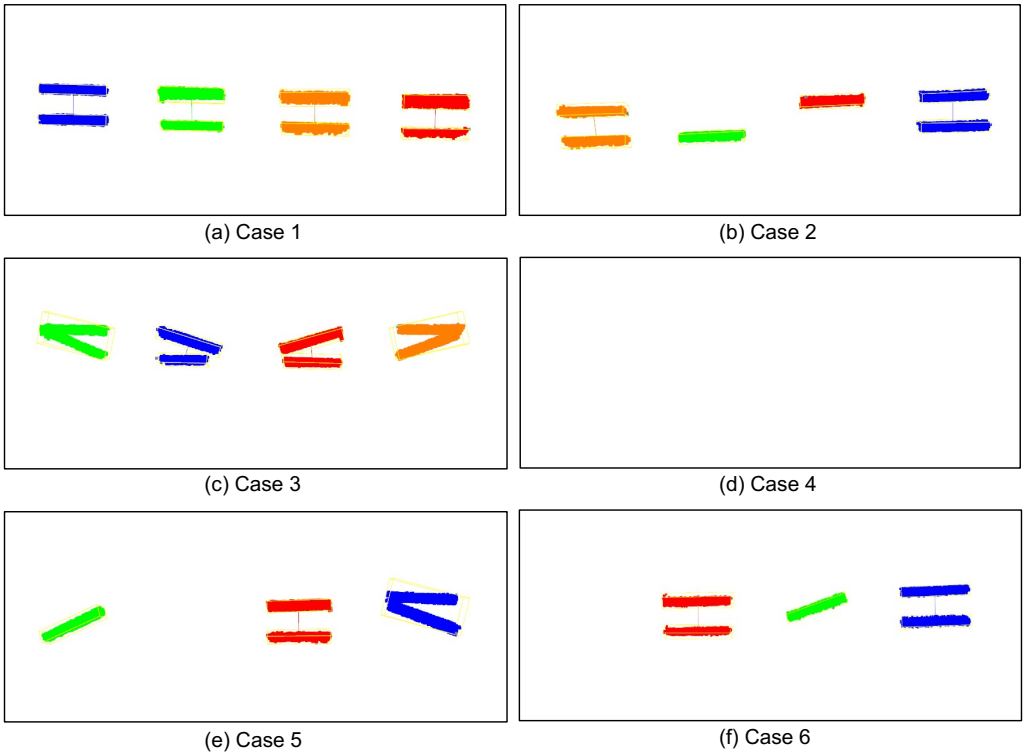


Figure 18: Guardrail boards grouped to form stacked groups of guardrails, achieving a bounding box location for each guardrail in all cases.

Figure 19: IFC file exported for each case with the corresponding safety state for each guardrail (note: Yellow refers to correct installation and red does not).

In Table 3, the classification result in green has been correctly identified, whereas the red has been wrongly classified. In case 1, the door guardrail has been wrongly classified as Safe, which is because the toe board has been wrongly classified as a wall, which can be seen in Figure 19a. This can be overcome by using additional features such as color [R, G, B] or textures for training the model to help the model to achieve higher accuracy. The pipeline built for predicting deviation between the SafeBIM and point cloud uses logic to check for classes other than floor below the guardrail to classify the guardrails with two guardrail boards (middle board and top board). Here, since the toe board is classified as a wall, the algorithm fails in this case.

Table 3: The automated analysis output shows the state of each guardrail element in SafeBIM, whether it is safe or unsafe compared to the UAV-collected as-built LiDAR data.

Element	Guardrail IFC GUID	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Window 1	3G9N4LEp58svsKnxtRiDf2	Safe	Safe	Unsafe	Unsafe	Unsafe	Unsafe
Window 2	1_AuwUzA5AJwjsuPNPZoEo	Safe	Unsafe	Unsafe	Unsafe	Unsafe	Safe
Window 3	1jk8jGYf59yOjwJguKgp3j	Safe	Unsafe	Unsafe	Unsafe	Safe	Unsafe
Door 1	0VsAeqZOD5GfsJpdL4kOAt	Safe	Unsafe	Unsafe	Unsafe	Unsafe	Unsafe

6. DISCUSSION AND CONCLUSION

The research presented in this study introduces a transformative approach to construction site safety inspections by integrating BIM, point cloud segmentation, and autonomous data collection using UAVs. The developed pipeline, centered around the "SafeBIM" framework and the ResPointNet++ segmentation model, demonstrates a robust method for automating the detection and verification of safety guardrails, addressing critical challenges associated with manual safety inspections. To ensure methodological rigor, the pipeline employs quantitative metrics for evaluation, such as part-level Intersection over Union. The findings underscore the system's potential to enhance safety compliance, reduce labor-intensive processes, and improve overall efficiency on construction sites.

The automated safety inspection pipeline successfully aligns as-built point cloud data with as-planned SafeBIM models to identify deviations in guardrail placement. By leveraging UAV-collected LiDAR data from an Ouster OS0 sensor and a fine-tuned ResPointNet++ model, the system achieved high accuracy in segmenting guardrail boards and poles, with part Intersection over Union (IOU) metrics of 0.9920 for guardrail boards and 0.9546 for guardrail poles, as shown in Table 1. The pipeline's ability to classify guardrail configurations as "safe" or "unsafe" based on predefined safety criteria (e.g., requiring three guardrail boards for doors or two for windows) was validated across six real-world scenarios at a construction site in Finland. The results, visualized in Figures 19-21, reveal that the system correctly identified safe configurations in most cases, though occasional misclassifications occurred, such as the erroneous labeling of a toe board as a wall in Case 1, which led to an incorrect "safe" classification for a door guardrail. These outcomes were rigorously assessed using precision-recall curves and confusion matrices, ensuring transparency and reproducibility.

The SafeBIM generation process, enhanced by SafeCONAI_v2, proved effective in automatically placing guardrails in fall hazard spaces, ensuring compliance with safety standards during the planning phase. The point cloud registration step, utilizing the ICP algorithm, successfully aligned as-built and as-planned data, enabling precise comparisons. The deviation analysis algorithm further streamlined the identification of missing or improperly installed guardrails, with results integrated back into the IFC model for actionable insights. These findings highlight the system's capability to reduce human error, accelerate inspection processes, and provide construction managers with clear, color-coded visualizations of safety compliance.

However, the study also identified limitations. The segmentation model's performance, while strong in controlled scenarios, occasionally struggled with complex site conditions, such as debris or misclassified elements, as seen in Case 1. Environmental factors like varying lighting, uneven terrain, and occlusions posed challenges for UAV data collection, potentially affecting point cloud quality. Additionally, the system's reliance on manual labeling for training data and the computational intensity of processing large point clouds suggest areas for optimization, including edge deployment to mitigate latency.

6.1 Potential impact on the construction industry

The automated safety inspection system offers significant benefits for the construction industry, particularly for site managers tasked with ensuring worker safety. By replacing labor-intensive manual inspections with UAV-based data collection and automated analysis, the system reduces inspection time and costs while improving consistency. The ability to frequently monitor safety compliance without disrupting construction schedules allows for proactive hazard mitigation, potentially lowering the risk of accidents, which remains a leading cause of injuries in construction (Pinto et al., 2011). The integration with 4D BIM ensures that safety measures align with project timelines, enabling managers to address deviations before they escalate into costly delays or regulatory violations.

For site managers, the color-coded IFC outputs (e.g., Figure 20) provide a visual interface that may support decision-making; however, usability evaluation with practitioners remains future work. This clarity supports rapid corrective actions, such as installing missing guardrails or retraining workers on safety protocols. The system's scalability also makes it adaptable to various project sizes, from residential buildings to large infrastructure developments, enhancing its practical utility across the industry. By using a data-driven approach to safety management, the pipeline aligns with broader industry trends toward digitalization and automation, positioning construction firms to meet stringent safety regulations more efficiently. Moreover, by generating geo-referenced deviation maps in SafeBIM, this work paves the way for bridging data integration with guided autonomous navigation: future extensions could condition UAV or AGV paths on these faulty guardrail locations as semantic waypoints, eliminating wasteful inspection routes and enabling targeted rescans, transforming static compliance checks into dynamic, proactive workflows.

6.2 Future work

To fully realize the proposed system's potential, several avenues for future research are recommended, as outlined in the Digital Twin for Construction Safety (DTCS) concept (Teizer et al., 2024). First, enhancing the robustness of the segmentation model through expanded training datasets that include diverse guardrail types, materials, and environmental conditions could improve accuracy in complex scenarios. Incorporating advanced deep learning techniques, such as unsupervised or semi-supervised learning, may reduce reliance on manual labeling, streamlining model development. Second, optimizing UAV navigation algorithms to better handle harsh site conditions such as low visibility or cluttered environments would ensure consistent data quality. Exploring hybrid data collection methods, combining UAVs with Autonomous Ground Vehicles (AGVs), could further enhance coverage and precision. Third, integrating the pipeline with real-time IoT sensor data, as suggested in prior studies (Mehranfar et al., 2024; John et al., 2020), could enable dynamic safety monitoring beyond static guardrail checks, such as tracking worker proximity to hazards. Developing a user-friendly interface for site managers to interact with the system and receive real-time alerts would enhance its adoption. Fourth, any (semi-) automatic or robotic safety compliance and inspection system may expose safety risks to end users and bystanders. As researchers, Francis et al. (2023) and Namian et al. (2021) noted that safe planning and control of unmanned or autonomous aerial and ground vehicles are key elements of good practice that have been field-tested. Finally, conducting extensive field tests across diverse construction projects spanning different climates, building types, and regulatory frameworks will validate the system's scalability and identify site-specific challenges. Standardizing the pipeline's integration with existing construction management software would also facilitate seamless adoption.

6.3 Conclusion

This research marks a significant step towards automating construction site safety inspections, demonstrating the feasibility of combining UAVs, point cloud segmentation, and BIM to monitor guardrail safety compliance. The pipeline addresses key limitations of manual inspections by offering a faster, more accurate, and cost-effective alternative, with direct benefits for site managers and the broader construction industry. While challenges remain, particularly in adapting to diverse site conditions and optimizing computational efficiency, the findings highlight a promising path toward safer and more efficient construction environments. Continued refinement and real-world validation will be critical to scaling this technology, ultimately contributing to a future where automated safety systems are a cornerstone of construction management. By reducing accidents, ensuring compliance, and enhancing operational efficiency through rigorous, data-integrated workflows, this work has the potential to reshape safety practices and support the industry's ongoing digital transformation, paving the way for novel integrations like semantic-guided registration and autonomous navigation.

ACKNOWLEDGEMENT

The research presented in this paper was in part funded by the European Union Horizon 2020 research and innovation program under grant agreement no. 101058548 (BEEYONDERS). This work also benefited from the collaboration with the NYUAD Center for Interacting Urban Networks (CITIES).

REFERENCES

Akhavian, R., Amani, M., Mootz, J., Ashe, R. and Beheshti, B. (2025). Building Information Models to Robot-Ready Site Digital Twins (BIM2RDT): An Agentic AI Safety-First Framework, arXiv, https://arxiv.org/abs/2509.20705.

Amani, M. and Akhavian, R. (2024). Safe and Trustworthy Robot Pathfinding with BIM, MHA*, and NLP, arXiv, https://arxiv.org/abs/2411.15371.

B 100 (2021). BG Bau, Absturzsicherungen auf Baustellen, Fall protection on construction sites - Side protection/barricades (in German: https://www.bgbau-medien.de/handlungshilfen_gb/daten/bausteine/b_100/b_100.htm.

Bahreini, F., Nasrollahi, M., Taher, A. and Hammad, A. (2024). Ontology for BIM-Based Robotic Navigation and Inspection Tasks, Buildings, vol. 14, no. 8, p. 2274, https://doi.org/10.3390/buildings14082274.

Besl, P.J. and McKay, N.D. (1992). A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 239–256, https://doi.org/10.1109/34.121791.

Bharathi, V., Prieto, S.A., García de Soto, B. and Teizer, J. (2024). Automating Construction Safety Inspections using Robots and Unsupervised Deep Domain Adaptation by Backpropagation, Proceedings of the 41st International Symposium on Automation and Robotics in Construction, pp. 855–862, https://doi.org/10.22260/ISARC2024/0111.

BLS Chart. (2023). Fatal occupational injuries by event or exposure, Economic News Release, U.S. Bureau of Labor Statistics. https://www.bls.gov/charts/census-of-fatal-occupational-injuries/fatal-occupational-injuries-by-event-drilldown.htm.

BLS Stats.(2023). Construction deaths due to falls, slips and trips increased 5.9 percent in 2021, The Economics Daily, U.S. Bureau of Labor Statistics. https://www.bls.gov/opub/ted/2023/construction-deaths-due-to-falls-slips-and-trips-increased-5-9-percent-in-2021.htm.

Bouaziz, S., Tagliasacchi, A. and Pauly, M. (2013). Sparse Iterative Closest Point, Computer Graphics Forum, vol. 32, no. 5, pp. 113–123, https://doi.org/10.1111/cgf.12178.

Braga, R.G., Tahir, M.O., Karimi, S., Dah-Achinanon, U., Iordanova, I. and St-Onge, D. (2025). Intuitive BIM-aided robotic navigation and assets localization with semantic user interfaces, Front. Robot. AI, vol. 12, p. 1548684, https://doi.org/10.3389/frobt.2025.1548684.

CloudComPy (2025), Python module, https://github.com/CloudCompare/CloudComPy.

CPWR Data. (2023). Injury and Incident Data, Stop Construction Falls, The Center for Construction Research and Training. https://stopconstructionfalls.com/research/injury-incident-data/.

Eastman, C., Lee, J., Jeong, Y. and Lee, J. (2009). Automatic rule-based checking of building designs, Automation in Construction, vol. 18, no. 8, pp. 1011–1033, https://doi.org/10.1016/j.autcon.2009.07.002.

Efraimidis, P.S. and Spirakis, P.G. (2006). Weighted random sampling with a reservoir, Information Processing Letters, vol. 97, no. 5, pp. 181–185, https://doi.org/10.1016/j.ipl.2005.11.003.

Ester, M., Kriegel, H.P., Sander, J. and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, in KDD'96. Portland, Oregon: AAAI Press, pp. 226–231, https://dl.acm.org/doi/10.5555/3001460.3001507.

Fontana, S., Cattaneo, D., Ballardini, A.L., Vaghi, M. and Sorrenti, D.G. (2021). A benchmark for point clouds registration algorithms," Robotics and Autonomous Systems, vol. 140, p. 103734, https://doi.org/10.1016/j.robot.2021.103734.

Francis, V., Sawhney, A., Manu, P., Shang, G. and Silva Bartolo, P.J. (2023). Handbook of Construction Safety, Health and Well-being in the Industry 4.0 Era, 1st ed. London: Routledge, https://doi.org/10.1201/9781003213796?urlappend=%3Futm_source%3Dresearchgate.

Ge, X. (2017). Automatic markerless registration of point clouds with semantic-keypoint-based 4-points congruent sets, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 130, pp. 344–357, https://doi.org/10.1016/j.isprsjprs.2017.06.011.

HÅNDBOGEN (2025). Branchearbejdsmiljørådet for Bygge & Anlæg, 2020th ed. in Håndbogen - Arbejdsmiljø i bygge og anlæg. https://bfa-ba.dk/haandbogen-arbejdsmiljoe-i-bygge-og-anlaeg/.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N. and Markham, A. (2019). RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds, arXiv, https://arxiv.org/abs/1911.11236.

Husemann, J., Kunz, M., Suiker, M. and Berns, K. (2025). From BIM to Autonomous Navigation: Using BIM Models to Enable Autonomous Navigation and Localization in Construction Environments, 42nd International Symposium on Automation and Robotics in Construction, Montreal, Canada, http://www.iaarc.org/publications/2025_proceedings_of_the_42nd_isarc_montreal_canada/from_bim_to_autonomous_navigation_using_bim_models_to_enable_autonomous_navigation_and_localization_in_construction_environments.html.

Ifcclouds (2023). https://github.com/D4ve-R/ifcclouds.

Jin, Z., Gambatese, J., Liu, D. and Dharmapalan, V. (2019). Using 4D BIM to assess construction risks during the design phase," ECAM, vol. 26, no. 11, pp. 2637–2654, https://doi.org/10.1108/ECAM-09-2018-0379?urlappend=%3Futm_source%3Dresearchgate.

Johansen, K.W., De Figueiredo, R.P., Golovina, O. and Teizer, J. (2021). Autonomous Safety Barrier Inspection in Construction: An Approach Using Unmanned Aerial Vehicles and Safe BIM, 38th International Symposium on Automation and Robotics in Construction, Dubai, UAE, http://www.iaarc.org/publications/2021_proceedings_of_the_38th_isarc/autonomous_safety_barrier_inspection_in_construction-an_approach_using_unmanned_aerial_vehicles_and_safe_bim.html.

Johansen, K.W., Schultz, C. and Teizer, J.(2022) BIM-based Fall Hazard Ontology and Benchmark Model for Comparison of Automated Prevention through Design Approaches in Construction Safety, Proceedings of the 29th EG-ICE International Workshop on Intelligent Computing in Engineering, EG-ICE, pp. 408–417, https://doi.org/10.7146/aul.455.c231.

Johansen, K.W., Schultz, C. and Teizer, J. (2023). Hazard ontology and 4D benchmark model for facilitation of automated construction safety requirement analysis, Computer aided Civil Eng, vol. 38, no. 15, pp. 2128–2144, https://doi.org/10.1111/mice.12988.

John, S.T., Roy, B.K., Sarkar, P. and Davis, R. (2020). IoT Enabled Real-Time Monitoring System for Early-Age Compressive Strength of Concrete, J. Constr. Eng. Manage., vol. 146, no. 2, p. 05019020, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001754.

Kanan, R., Elhassan, O. and Bensalem, R. (2018). An IoT-based autonomous system for workers' safety in construction sites with real-time alarming, monitoring, and positioning strategies, Automation in Construction, vol. 88, pp. 73–86, https://doi.org/10.1016/j.autcon.2017.12.033.

Khan, N., Saleem, M.R., Lee, D., Park, M.W. and Park, C. (2021). Utilizing safety rule correlation for mobile scaffolds monitoring leveraging deep convolution neural networks, Computers in Industry, vol. 129, p. 103448, https://doi.org/10.1016/j.compind.2021.103448.

Kim, J., Paik, S., Lian, Y., Kim, J. and Kim, H. (2023). Transformation of Point Clouds to Images for Safety Analysis of Scaffold Joints, 40th International Symposium on Automation and Robotics in Construction, Chennai, India, http://www.iaarc.org/publications/2023_proceedings_of_the_40th_isarc_chennai_india/transformation_of_point_clouds_to_images_for_safety_analysis_of_scaffold_joints.html.

Kim, K., Cho, Y. and Zhang, S. (2016). Integrating work sequences and temporary structures into safety planning: Automated scaffolding-related safety hazard identification and prevention in BIM, Automation in Construction, vol. 70, pp. 128–142, https://doi.org/10.1016/j.autcon.2016.06.012.

Kolar, Z., Chen, H. and Luo, X. (2018). Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images, Automation in Construction, vol. 89, pp. 58–70, https://doi.org/10.1016/j.autcon.2018.01.003.

Li, B., Fitzgerald, J. and Schultz, C. (2022). Modelling the impacts of crowds on occupants in the built environment—A static, rule-based approach to human perception and movement, Advanced Engineering Informatics, vol. 51, p. 101452, https://doi.org/10.1016/j.aei.2021.101452.

Li, Y., Esmaeili, B., Gheisari, M., Kosecka, J. and Rashidi, A. (2022). Using Unmanned Aerial Systems (UAS) for Assessing and Monitoring Fall Hazard Prevention Systems in High-rise Building Projects, arXiv, https://arxiv.org/abs/2209.13137.

Liu, E. and Atinfinity. (2021). EdwardLiuyc/StaticMapping: Release for DOI, Zenodo, https://zenodo.org/record/4740488.

Liu, W., Sun, W., Wang, S. and Liu, Y. (2021). Coarse registration of point clouds with low overlap rate on feature regions," Signal Processing: Image Communication, vol. 98, p. 116428, https://doi.org/10.1016/j.image.2021.116428.

Masood, M.K., Aikala, A., Seppänen, O. and Singh, V. (2020). Multi-Building Extraction and Alignment for As-Built Point Clouds: A Case Study With Crane Cameras, Front. Built Environ., vol. 6, p. 581295, https://doi.org/10.3389/fbuil.2020.581295.

Mavridis, P., Andreadis, A. and Papaioannou, G. (2015). Efficient Sparse ICP, Computer Aided Geometric Design, vol. 35–36, pp. 16–26, https://doi.org/10.1016/j.cagd.2015.03.022.

Mehranfar, M., Braun, A. and Borrmann, A. (2024). From dense point clouds to semantic digital models: End-to-end AI-based automation procedure for Manhattan-world structures, Automation in Construction, vol. 162, p. 105392, https://doi.org/10.1016/j.autcon.2024.105392.

Mihić, M., Sigmund, Z., Završki, I. and Butković, L.L. (2023). An Analysis of Potential Uses, Limitations and Barriers to Implementation of 3D Scan Data for Construction Management-Related Use—Are the Industry and the Technical Solutions Mature Enough for Adoption?, Buildings, vol. 13, no. 5, p. 1184, https://doi.org/10.3390/buildings13051184.

Namian, M., Khalid, M., Wang, G. and Turkan, Y. (2021). Revealing Safety Risks of Unmanned Aerial Vehicles in Construction, Transportation Research Record: Journal of the Transportation Research Board, vol. 2675, no. 11, pp. 334–347, https://doi.org/10.1177/03611981211017134.

OSHA. (2019).Occupational Safety and Health Administration, fall protection systems and falling object protection-criteria and practices, 2019th ed. in Standard 1910.29. OSHA, https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.29.

Pinto, A., Nunes, I.L. and Ribeiro,R.A. (2011). Occupational risk assessment in construction industry – Overview and reflection, Safety Science, vol. 49, no. 5, pp. 616–624, https://doi.org/10.1016/j.ssci.2011.01.003.

Qi, C.R., Yi, L., Su, J. and Guibas, L.J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, arXiv, https://arxiv.org/abs/1706.02413.

Qi, J., Issa, R.R.A., Hinze, J. and Olbina, S. (2011). Integration of Safety in Design through the Use of Building Information Modeling, Computing in Civil Engineering (2011), Miami, Florida, United States: American Society of Civil Engineers, pp. 698–705, http://ascelibrary.org/doi/10.1061/41182%28416%2986.

Qi, J., Issa, R.R.A., Olbina, S. and Hinze, J. (2014). Use of Building Information Modeling in Design to Prevent Construction Worker Falls, J. Comput. Civ. Eng., vol. 28, no. 5, p. A4014008, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000365.

Schneider, P.J. and Eberly, D.H. (2003). Geometric tools for computer graphics, Chapter 3. in The Morgan Kaufmann series in computer graphics and geometric modeling. Amsterdam: Boston Morgan Kaufmann Publishers.

Siebert, S. and Teizer, J. (2014). Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system, Automation in Construction, vol. 41, pp. 1–14, https://doi.org/10.1016/j.autcon.2014.01.004.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv, https://arxiv.org/abs/1409.1556.

Soygaonkar, A.R., and Bhangale, P.P. (2014). Study of Job Layout for Construction Project, IJERT, vol. Volume 03, Issue 11, https://www.ijert.org/study-of-job-layout-for-construction-project.

Sun, Q. and Turkan, Y. (2020). A BIM-based simulation framework for fire safety management and investigation of the critical factors affecting human evacuation performance," Advanced Engineering Informatics, vol. 44, p. 101093, https://doi.org/10.1016/j.aei.2020.101093.

Teizer, J., Johansen, K.W. and Schultz, C. (2022). The Concept of Digital Twin for Construction Safety, Construction Research Congress 2022, Arlington, Virginia: American Society of Civil Engineers, pp. 1156–1165, https://ascelibrary.org/doi/10.1061/9780784483961.121.

Teizer, J., Johansen, K.W., Schultz, C.L., Speiser, K., Hong, K. and Golovina, O. (2024). A Digital Twin Model for Advancing Construction Safety, Construction Logistics, Equipment, and Robotics, J. Fottner, K. Nübel, and D. Matt, Eds., in Lecture Notes in Civil Engineering, vol. 390. Cham: Springer Nature Switzerland, pp. 201–212, https://link.springer.com/10.1007/978-3-031-44021-2_22.

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F. and Guibas, L.J. (2019). KPConv: Flexible and Deformable Convolution for Point Clouds, arXiv, https://arxiv.org/abs/1904.08889.

UnitX. (2025). The role of point cloud data in modern machine vision, UnitX Labs, https://www.unitxlabs.com/resources/point-cloud-machine-vision-system-3d-perception-automation/.

Xu, J., Cheung, C., Manu, P., Ejohwomu, O. and Too, J. (2023). Implementing safety leading indicators in construction: Toward a proactive approach to safety management, Safety Science, vol. 157, p. 105929, https://doi.org/10.1016/j.ssci.2022.105929.

Yin, C., Wang, B., Gan, V.J.L., Wang, M. and Cheng, J.C.P. (2021). Automated semantic segmentation of industrial point clouds using ResPointNet++, Automation in Construction, vol. 130, p. 103874, https://doi.org/10.1016/j.autcon.2021.103874.

Zhang, S., Teizer, J., Lee, J.K., Eastman, C.M. and Venugopal, M. (2013). Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules, Automation in Construction, vol. 29, pp. 183–195, https://doi.org/10.1016/j.autcon.2012.05.006.

Zhang, J., Yao, Y. and Deng, B. (2021). Fast and Robust Iterative Closest Point, IEEE Trans. Pattern Anal. Mach. Intell., pp. 1–1, https://doi.org/10.48550/arXiv.2007.07627.

Zhao, H., Jiang, L., Jia, J., Torr, P. and Koltun, V. (2020).Point Transformer, arXiv, https://arxiv.org/abs/2012.09164.

APPENDIX

All code used in this work has been uploaded to the GitHub repository: https://github.com/vimavb45/Guardrail-Safety-Inspection-using-BIM-and-Pointcloud-Segmentation.

Deviation between the as-planned and the as-built algorithm

```
AnalyzeGuardrailDeviations(PointCloudFile, IFCFile): String;
 // Input: PointCloudFile (coordinates and labels), IFCFile (SafeBIM model)
 // Output: Path to modified IFC file

 GuardrailPoints := EXTRACT_POINTS(PointCloudFile, Labels = {"guardrail_post", "guardrail_bar"});
 IF GuardrailPoints.size = 0 THEN
  PRINT "No guardrail points found in point cloud";
  RETURN "";
 END IF;

 Clusters := DENSITY_BASED_CLUSTERING(GuardrailPoints, MaxDist, MinPts);
 FOR i FROM 1 TO Clusters.size DO
  Cluster := Clusters.get(i);
  BoundingBox := COMPUTE_ORIENTED_BOUNDING_BOX(Cluster);
  ClusterType := DETERMINE_TYPE(Cluster, Majority = {"guardrail_post", "guardrail_bar"});
  ClusterInfo.append(ClusterType, BoundingBox, CENTROID(BoundingBox));
 END FOR;

 StackedGroups := EMPTY_LIST;
 FOR i FROM 1 TO Clusters.size DO
  Cluster1 := Clusters.get(i);
  FOR j FROM i+1 TO Clusters.size DO
   Cluster2 := Clusters.get(j);
   IF VERTICAL_ALIGNMENT(Cluster1, Cluster2, HorzTol, VertTol) THEN
    StackedGroups.append(GROUP(Cluster1, Cluster2));
   END IF;
  END FOR;
 END FOR;

 SafetyClassification := EMPTY_DICT;
 FOR i FROM 1 TO StackedGroups.size DO
  Group := StackedGroups.get(i);
  InstanceCount := COUNT_INSTANCES(Group);
  PointsBelow := GET_POINTS_BELOW(Group, PointCloudFile, Exclude = {"floor", "guardrail_post",
"guardrail_bar"});

  IF InstanceCount = 3 THEN
   SafetyClassification.set(Group, "safe", Reason = "three instances");
  ELSE IF InstanceCount = 2 AND PointsBelow.size > 0 THEN
   SafetyClassification.set(Group, "safe", Reason = "two instances with points below");
  ELSE
   SafetyClassification.set(Group, "unsafe", Reason = "insufficient instances or support");
  END IF;
 END FOR;

 IFCGuardrails := LOAD_IFC_GUARDRAILS(IFCFile, Type = "IfcRailing", Keyword = "guardrail");
 ComparisonResults := EMPTY_DICT;
```

```
IF StackedGroups.size > 0 THEN
  FOR i FROM 1 TO IFCGuardrails.size DO
    IFCGuardrail := IFCGuardrails.get(i);
    IFCBBox := COMPUTE_BOUNDING_BOX(IFCGuardrail.geometry);
    IFCCentroid := CENTROID(IFCBBox);
    MatchFound := False;
    ClosestGroup := NULL;
    MinDistance := INFINITY;

    FOR j FROM 1 TO StackedGroups.size DO
      Group := StackedGroups.get(j);
      GroupCentroid := CENTROID(Group.bounding_boxes);
      Distance := COMPUTE_DISTANCE(IFCCentroid, GroupCentroid);
      Overlap := CHECK_BBOX_OVERLAP(IFCBBox, Group.bounding_boxes);

      IF Distance < DistThreshold OR Overlap THEN
        MatchFound := True;
        IF Distance < MinDistance THEN
          MinDistance := Distance;
          ClosestGroup := Group;
        END IF;
      END IF;
    END FOR;

    IF MatchFound THEN
      ComparisonResults.set(IFCGuardrail.id, Present = True, MatchedGroup = ClosestGroup,
                    Distance = MinDistance, Safety = SafetyClassification.get(ClosestGroup).status,
                    Reason = SafetyClassification.get(ClosestGroup).reason);
    ELSE
      ComparisonResults.set(IFCGuardrail.id, Present = False, MatchedGroup = NULL,
                    Distance = NULL, Safety = "unsafe", Reason = "not present in point cloud");
    END IF;
  END FOR;
ELSE
  FOR i FROM 1 TO IFCGuardrails.size DO
    IFCGuardrail := IFCGuardrails.get(i);
    ComparisonResults.set(IFCGuardrail.id, Present = False, MatchedGroup = NULL,
                  Distance = NULL, Safety = "unsafe", Reason = "no guardrails detected in point cloud");
  END FOR;
END IF;

ModifiedIFC := LOAD_IFC(IFCFile);
RedColor := DEFINE_COLOR(RGB = (1.0, 0.0, 0.0));
UnsafeStyle := CREATE_STYLE(RedColor);

FOR i FROM 1 TO IFCGuardrails.size DO
  IFCGuardrail := IFCGuardrails.get(i);
  IF ComparisonResults.get(IFCGuardrail.id).Safety = "unsafe" THEN
    Geometry := GET_GEOMETRY(IFCGuardrail);
    IF Geometry.has_style THEN
      UPDATE_STYLE(Geometry, UnsafeStyle);
    ELSE
      ADD_STYLE(Geometry, UnsafeStyle, Name = "unsafe_guardrail");
    END IF;
```

```
    END IF;
  END FOR;

  OutputFile := GENERATE_NEW_FILENAME(IFCFile, Suffix = "_modified");
  SAVE_MODIFIED_IFC(ModifiedIFC, OutputFile);
  PRINT "Modified IFC saved at:", OutputFile;

  RETURN OutputFile;
END;
```