BIM-Guided Semantic Fingerprinting for Vision-Based Construction Progress Monitoring of Cold-Formed Steel Framing

A thesis submitted in fulfilment of the requirements for the degree of

Master of Engineering

at

The University of Waikato

by

Muhammad Ahmad

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2026

Abstract

Construction progress monitoring remains largely dependent on manual inspection, limiting the frequency and reliability of updates on active sites. Existing automated approaches have demonstrated value but commonly rely on fixed camera infrastructure, depth sensing, or metric camera-to-BIM registration, requirements that are difficult to justify on budget-constrained residential projects, and most treat BIM as a passive post-detection reference rather than an active constraint during visual interpretation.

To address this gap, this study proposes a BIM-guided semantic fingerprinting framework for element-level progress monitoring of cold-formed steel framing using standard monocular video. A custom YOLOv8m detection model identifies wall panels, doors, and windows across video frames; visual odometry provides temporal reasoning, detecting façade transitions and suppressing duplicate detections as the camera moves along the building. Accepted detections are aggregated into semantic fingerprints and matched against BIM-derived façade representations through a reward-based scoring system, enabling façade identification and element assignment without metric registration; wall-panel coverage is assessed separately as spatial continuity along each face. A gap correction mechanism recovers missed detections using BIM spatial context, and a pyRevit extension supports end-to-end BIM extraction and in-model progress visualisation.

Evaluated on a single-storey residential cold-formed steel structure in Hamilton, New Zealand, the framework correctly identified all four building façades, matched element counts to ground truth, and reduced 2,050 raw detections across 983 frames to 20 unique element observations, a 99.02% reduction, without fixed infrastructure, depth sensing, or point-cloud reconstruction. The work is currently validated on a single proof-of-concept site, and further testing on independent projects is needed to establish broader generalisability.

Preface

The first months were spent not solving the problem, but finding it. Computer vision in construction is a well-populated field, and locating a gap worth filling, one that was both genuinely unsolved and actually tractable, took longer than I would care to admit.

The eventual answer came from a shift in perspective: away from what a camera could see, and toward what it could understand. This reframing, from pixel-level detection to semantic interpretation grounded in BIM context, became the foundation of the work that followed.

This thesis began, as many things do, with a problem that seemed straightforward. Construction sites need to be monitored. Cameras exist. Surely someone could point one at a building and figure out what was going on. Several months and a worrying amount of troubleshooting later, I can confirm that it is not that straightforward.

What I did not anticipate was how much resistance a genuinely novel approach would put up. When a method has not been tried before, there is no roadmap for the awkward middle, only the slow accumulation of evidence that the direction is right and the details are wrong. There were more dead ends, restarts, and recalibrations than the tidy methodology section that follows would suggest.

But the plan did work. I hope the reader finds the journey at least as instructive as I did, and ideally, rather quicker.

Acknowledgements

I would like to express my sincere gratitude to Prof. James Lim, my supervisor, for his continuous guidance, technical direction, and support throughout the development of this research. His supervision not only provided clarity during complex stages of the work but also encouraged me to rethink and refine my approach, making the research both more structured and a valuable learning experience.

I am also grateful to Dr Rehan Masood for his encouragement, valuable feedback, and motivational support during the course of this study. My thanks extend to Dr Arthur Fang for his insightful input and guidance, which helped shape key aspects of this work.

I would like to acknowledge the CG11 research group for providing a supportive research environment, access to resources, and opportunities for academic development throughout this project. I am also appreciative of my friends, Humaira Hameed and Kevin Ramani, for their ongoing academic discussions and assistance during my time in New Zealand and beyond.

I am grateful to my badminton group for providing balance, consistency, and a supportive space outside of academic work.

Finally, I am deeply grateful to my family for their unwavering encouragement, patience, and support throughout my academic journey.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BIM	Building Information Modelling
CFS	Cold-Formed Steel
CV	Computer Vision
DeepSORT	Deep Simple Online and Realtime Tracking
Faster R-CNN	Faster Region-based Convolutional Neural Network
FPN	Feature Pyramid Network
FPS	Frames Per Second
IFC	Industry Foundation Classes
IoT	Internet of Things
IoU	Intersection over Union
JSON	JavaScript Object Notation
LiDAR	Light Detection and Ranging
mAP	Mean Average Precision
Mask R-CNN	Mask Region-based Convolutional Neural Network
ORB	Oriented FAST and Rotated BRIEF
PPVC	Prefabricated Prefinished Volumetric Construction
RANSAC	Random Sample Consensus
ResNet	Residual Network
RFID	Radio Frequency Identification
RMS	Root Mean Square
SfM	Structure from Motion
SGD	Stochastic Gradient Descent
SLAM	Simultaneous Localisation and Mapping

SORT	Simple Online and Realtime Tracking
SSD	Single Shot MultiBox Detector
UAV	Unmanned Aerial Vehicle
VO	Visual Odometry
YOLO	You Only Look Once

Chapter 1

Introduction

1.1. Problem statement

Construction projects frequently face delays and cost overruns, with studies reporting substantial schedule slippages and budget increases across multiple international contexts [1–3]. An industry report estimates that around 98% of megaprojects incur cost overruns exceeding 30%, while 77% experience schedule delays of at least 40% [4]. Reliable progress monitoring is therefore essential for project control, as it supports timely intervention, quality assurance, and evidence-based decision-making related to cost, time, quality, and safety [5,6]. Yet, the construction industry continues to depend on labour-intensive, error-prone data collection methods that hinder timely and reliable reporting. Current manual inspection approaches struggle to deliver timely and reliable feedback on quality and progress, typically requiring multiple site visits to confirm compliance [7].

The problem is particularly relevant in residential and light-framed construction, where frequent monitoring is desirable, but project budgets may not justify specialised sensing infrastructure. As housing demand grows in New Zealand and construction projects become increasingly complex, such manual assessment methods may no longer be sufficient to ensure compliance with the New Zealand Building Code and monitoring requirements [8]. These practical constraints have driven growing interest in automated approaches for more scalable and consistent construction progress monitoring. Automated progress monitoring has therefore received increasing attention. However, existing approaches still face practical and deployment constraints that limit their adoption in ordinary construction environments.

1.2. Research gap

Automated monitoring approaches broadly divide into non-visual and vision-based methods, each with characteristic deployment constraints. Non-visual methods, including GPS, RFID and sensor-based systems, can support component tracking and identification, but they usually require dedicated hardware installations and may perform poorly in indoor or partially enclosed environments [9,10]. Vision-based methods have become increasingly attractive because site images and videos can be collected using common devices and can support activity recognition, equipment tracking, worker monitoring, and construction progress assessment [11–13]. Deep learning detectors, including two-stage methods such as Faster R-CNN [14] and single-stage methods such as YOLO [15] have made image-based recognition more practical, while segmentation methods such as Mask R-CNN [16] can provide more detailed pixel-level interpretation where sufficient annotation is available. Vision-based approaches reduce hardware dependencies, but their practical deployment remains challenging. Frequent occlusions, dynamic site conditions, and sensitivity to lighting variations, especially in partially completed or indoor environments, complicate their use [10,17]. These limitations highlight the need for lightweight, cost-effective monitoring solutions that operate reliably under visually complex conditions.

Beyond these sensing constraints, a deeper interpretive limitation persists: most existing studies treat BIM as a static reference for post-detection comparison rather than drawing on its rich semantic information during interpretation [13,18–21]. Studies integrating BIM and computer vision have demonstrated the value of combining model-based context with image-derived observations for progress monitoring, localisation, and safety assessment. However, a fundamental interpretive challenge remains unresolved across these approaches: as a camera moves along a building face, the same physical element appears across multiple frames. Tracking-by-detection methods such as SORT [22], DeepSORT [23], ByteTrack [24], and BoT-SORT [25] can maintain object identities in image space, but they do not identify which building face is being observed or assign detections to planned BIM elements. Façade identification typically requires fixed camera setups, external localisation, or explicit camera-to-BIM registration, requirements that are difficult to achieve in ordinary residential construction [20,26]. Together, these observations point to a specific and largely unaddressed gap: the semantic content of BIM, element type, quantity, order, and position is rarely used as an active constraint to interpret monocular video, identify façades, and assign detections to planned elements without metric registration.

1.3. Research aim

This study aims to develop a framework that uses BIM semantic information as an active interpretive constraint for element-level progress verification from monocular video, without metric registration or specialised sensing. Unlike Scan-vs-BIM, SfM-BIM, and SLAM-BIM frameworks that rely on metric registration or dedicated localisation hardware, the key novelty lies not in the object detector itself but in the use of BIM-derived semantic façade fingerprints to support video interpretation, façade identification, duplicate suppression, and element assignment. Each façade fingerprint encodes the planned element types, quantities, spatial order, façade location, floor level, and relative element positions. YOLO-based detections provide confidencescored observations of doors, windows, and wall panels, while visual odometry (VO) is used for sequence continuity, façade transition detection, and temporal duplicate suppression. Detected door and window openings are aggregated into observed fingerprints and matched against BIM-derived side representations to identify façades and assign elements, while wall panel detections are assessed separately as coverage.

The proposed framework was evaluated on a single-storey cold-formed steel framing structure in Hamilton, New Zealand (Site 1), where ground-truth element counts were established through direct physical inspection. Additional examples from the site are included throughout Chapter 3 to illustrate key framework components. The validation is designed as a controlled proof of concept on a representative single-storey CFS structure, demonstrating the framework's core mechanisms.

The main contributions of this study are: (1) a BIM-derived semantic fingerprint representation that encodes façade-level element type, planned quantity, order, and relative position; (2) a monocular-video processing workflow that combines YOLO detection, visual-odometry-assisted temporal reasoning, and duplicate suppression; (3) a BIM matching and visualisation process that assigns confirmed observations to planned elements for element-level presence verification.

1.4. Objectives

To achieve this aim, the study pursues the following objectives:

1. Formulate a BIM-derived semantic fingerprint representation that encodes façade-level element type, quantity, spatial order, and relative position.
2. Design a monocular video interpretation pipeline that integrates object detection, visual odometry, and temporal consistency reasoning for façade-level observation.
3. Establish a BIM-guided matching and assignment mechanism that maps visual observations to BIM elements using semantic constraints and gap correction.
4. Implement the proposed framework and evaluate its performance on a single-storey cold-formed steel structure with known ground truth.
5. Investigate the robustness and failure conditions of BIM-guided façade identification under monocular video constraints.

The rest of this thesis is organised as follows. Chapter 2 reviews the literature on automated construction monitoring and BIM-CV integration. Chapter 3 presents the framework methodology, covering the object detection pipeline, VO-based temporal reasoning, semantic fingerprinting, and BIM matching. Chapter 4 evaluates three core technical components: object detection model selection, scene segmentation and façade transition detection, and element counting accuracy. Chapter 5 discusses results, framework limitations, and directions for future work. Chapter 6 presents the conclusions.

Chapter 2

Literature review

Recent advancements in Building Information Modelling (BIM) and computer vision (CV) have enabled a wide range of automated approaches for construction progress monitoring. This literature review focuses on three areas: (1) vision-based approaches to construction progress monitoring, (2) BIM as a semantic resource and (3) existing BIM-CV integration frameworks.

2.1. Vision-based approaches to construction progress monitoring

The construction industry has undergone a steady shift toward digitisation, driven by long-standing challenges in coordination, quality control, and information management [27,28]. With the growing adoption of digital technologies in construction, numerous automated monitoring approaches have been proposed to replace or supplement manual inspection and reporting.

Early digital monitoring methods addressed these challenges by attaching physical identifiers to components and using non-vision-based technologies such as RFID, GPS, Bluetooth beacons, and QR-code tagging to track component locations, installation status, or workforce movement [9,29]. These approaches enable direct retrieval of BIM metadata during on-site inspections by scanning tagged components, improving traceability and reducing manual data-entry errors. This is effective for controlled environments such as prefabrication and modular construction, where components are accessible and systematically tagged before delivery. On live sites, however, these methods are subject to limitations in read range, signal interference, and localisation accuracy. In addition, performance can degrade in environments with physical obstructions or enclosed components that restrict signal propagation or visibility [9,30,31].

Laser scanning and LiDAR offer a more geometrically rigorous alternative to image-based methods, generating dense point clouds that can be compared directly against as-planned BIM models. Spatial deviations are detectable with high accuracy, and progress can be inferred by identifying regions where geometry has or has not materialised [32]. The main trade-off is operational rather than technical: while laser scanning provides high geometric accuracy, it increases hardware cost, field setup time, and post-processing effort [33,34]. In practice, sensing data and BIM models are commonly treated as separate inputs and only compared after processing, rather than being integrated throughout the workflow. As a result, BIM is often used as a fixed reference for checking results, instead of acting as a continuously updated model during data collection and interpretation [35,36].

More recently, image processing and deep learning techniques have been applied for object detection, semantic segmentation, activity recognition, and material classification to infer construction progress. Compared to sensor-based approaches, image-based methods are easier to deploy and can be captured using widely available devices, although their performance remains sensitive to lighting, occlusion, viewpoint, and scene complexity. Deep learning-based object detection architectures have become the dominant approach for vision-based construction monitoring.

Two-stage detectors, such as Faster R-CNN, provide high localisation accuracy through region proposal mechanisms, making them suitable for detailed component recognition tasks [14]. Single-stage detectors such as the YOLO (You Only Look Once) family and SSD enable faster inference speeds, supporting near real-time applications on mobile or edge devices [15,37]. More recent iterations, including YOLOv8, extend this with anchor-free detection heads, feature pyramid networks, and multi-scale feature fusion, improving reliability on small and partially occluded components [38].

Computer vision approaches applied to construction progress monitoring typically fall into several categories depending on the level of scene understanding required. Object detection methods identify discrete components such as structural elements, openings, or safety equipment, while semantic and instance segmentation approaches provide pixel-level classification of materials and structural regions. Beyond these, activity recognition methods have been applied to infer installation progress from worker and equipment interactions. Each level of abstraction offers a different degree of scene understanding, but all rely primarily on visual appearance features and lack inherent spatial or semantic context, a limitation that becomes acute when structural elements are visually similar, partially occluded, or encountered from varying viewpoints.

Among vision-based approaches, the most directly comparable work involves detecting or counting individual building elements for progress assessment. Hamledari et al. [39] detected interior partition components from 2D images using shape- and colour-based modules, inferring partition assembly state for comparison against a 4D BIM. Chua and Cheah [40] counted windows on prefabricated prefinished volumetric construction (PPVC) façades with a YOLOv5 detector, rectifying occluded or missed windows from the repetitive column structure of multi-storey modules before checking detected counts against as-planned column totals. Both demonstrate element-level counting under occlusion, but operate on single images of one structural typology and report aggregate state or completion percentage rather than assigning individual detections to specific planned elements; neither resolves which façade is being observed nor maintains element identity as a camera moves across a building.

Maintaining temporal consistency across frames introduces a further challenge that is less frequently addressed in the detection literature. As a camera moves along a building face, the same physical element appears repeatedly across many frames, and without cross-frame reasoning, each appearance risks being registered as a distinct detection. VO and SLAM-based methods allow relative camera tracking without external infrastructure, making them compatible with handheld or mobile capture workflows. VO approaches generally include both feature-based and direct methods, which differ in whether tracking relies on explicit keypoint matching or pixel intensity alignment. Among feature-based methods, ORB (Oriented FAST and Rotated BRIEF) is widely adopted due to its computational efficiency, rotation invariance, and suitability for deployment on standard mobile hardware without GPU acceleration. In construction monitoring research, VO and SLAM techniques are often used to spatially register captured imagery with BIM geometry or reconstructed point clouds.

However, these approaches typically focus on metric localisation accuracy, requiring sufficient scene texture and stable lighting conditions to maintain reliable feature correspondence [41–43]. Construction environments frequently contain texture-poor surfaces, repetitive structural elements, and temporary occlusions, which can introduce drift or tracking instability. Additional degradation may arise from rolling shutter distortion in smartphone cameras, where sequential row capture introduces minor geometric inconsistencies during camera motion, potentially reducing feature matching reliability [42]. In the present study, VO is used not for spatial localisation but for temporal reasoning, specifically to detect sequence continuity across frames and prevent duplicate element counting as the camera moves along the building face.

2.2. BIM as a semantic resource

Building Information Modelling (BIM) is frequently characterised as a three-dimensional geometric representation of a facility, yet a BIM model also encodes structured non-geometric information that describes what each element is and how elements relate to one another [28]. Beyond shape and dimension, object-based models hold attribute data such as element type, material, and planned quantity, together with spatial relationships including adjacency, containment, and sequence, describing a building that is simultaneously geometric, spatial, and semantic [13,28]. This layered information has underpinned a broad range of construction applications, from 4D scheduling to quality and progress assessment [17,18].

In progress monitoring specifically, this information is most often accessed through its geometric layer. As-planned geometry provides a metric reference against which captured site data can be compared, and progress is inferred where expected geometry has or has not materialised [35]. This geometric framing is effective where dense spatial data can be reconstructed and registered, but it draws on only part of the information a model contains; semantic attributes such as element type, planned count, and spatial order are typically retained in the model yet remain unused during interpretation [17]. The value of these attributes lies in their stability: while the visual appearance of an element varies with viewpoint, lighting, and stage of completion, its planned type, quantity, and position within a face are fixed properties recorded in the model in advance.

Semantic and relational attributes of this kind are exposed directly by object-based modelling, in which each element carries a type classification and a set of properties that persist independently of how the element is later observed [28]. These attributes describe a building at a level of abstraction well suited to constraining interpretation: the number of openings expected on a face, their left-to-right order, and their relative spacing are compact, view-invariant descriptors that can be evaluated without metric reconstruction. Studies that combine model-based context with image-derived observations have demonstrated the benefit of such planned information for resolving ambiguity in visually complex scenes [20,21], a benefit that is most pronounced in repetitive structural systems where elements are visually similar and appearance alone is insufficient to distinguish them.

Where BIM has been used to actively inform visual inference rather than to verify results afterwards, the constraint applied has most often been geometric. Golparvar-Fard et al. [44] used as-planned model geometry to reason about occlusion and expected appearance when reconstructing as-built point clouds, and Braun et al. [45,46] combined model-derived geometric constraints and precedence relationships with photogrammetric point clouds to infer the presence of occluded components. These approaches show that planned model information can guide interpretation, though they operate on reconstructed three-dimensional data and so retain a dependence on metric registration. The semantic and relational attributes of a model, type, quantity, order, and relative position have comparatively seldom been applied as a direct constraint on image-space observations, particularly for monocular video, where metric reconstruction is unreliable. Representing a building face by these attributes offers a route to interpreting detections without registration and motivates the semantic fingerprinting approach developed in this study.

2.3. Integration of BIM and computer vision

BIM-CV integration has been explored across a range of construction monitoring tasks. Most frameworks use BIM geometry as a reference against which visual observations are aligned or compared. The common pattern is to capture site data through vision or sensing, reconstruct a spatial representation, and then assess deviation from the BIM model. While effective under controlled conditions, this pipeline reinforces the role of BIM as a passive endpoint rather than an active participant in the interpretation process, a pattern that persists largely because metric registration and spatial reconstruction are prerequisites for geometric comparison, leaving semantic BIM attributes such as element type, quantity, and spatial order unused during inference.

Several studies have explored BIM-CV integration by using BIM as a reference model for interpreting visual data. Wei et al. [20] proposed a vision-based indoor construction monitoring framework that employed a handheld stereo camera and Mask R-CNN for area recognition, combined with SLAM-based localisation to align observations with the BIM model. While effective, the framework depends on localisation hardware, limiting its scalability and deployability. Elrifaee and Zayed [26] introduced a BIM-CV framework integrated with ROS and IoT-based depth sensors to support real-time safety monitoring. Although the system demonstrates real-time spatial awareness, its reliance on external sensors and hardware infrastructure constrains its applicability under diverse site conditions.

Beyond hardware-assisted approaches, several studies have explored BIM-CV integration through software-driven workflows that enable information exchange between the two systems. For example, Kulinan et al. [47] developed an integrated framework for automated workforce safety monitoring, where workers were detected and tracked using computer vision and spatially localised within the BIM environment. This approach enabled real-time visualisation of worker positions and safety zones directly in BIM but was limited by occlusions and obstructed camera views, which affected tracking accuracy.

A more active role of BIM in supporting computer vision is demonstrated by Golparvar-Fard et al. [44], who used as-planned BIM models to guide the reconstruction of as-built 3D point clouds from unordered site photographs. Their voxel-colouring and Bayesian inference approach leveraged BIM geometry to reason about occlusions and automatically detect progress deviations.

More recently, Hsieh et al. [48] achieved component-level construction phase recognition by integrating multi-camera object detection with BIM through explicit camera-to-model registration, enabling colour-coded progress visualisation within a virtual environment. Pal et al. [49] extended this to activity-level monitoring, combining SfM-based point cloud reconstruction with 4D BIM and semantic segmentation of orthographic views to estimate completion percentages across multiple construction activities.

While BIM provides detailed geometric, spatial, and semantic information, it typically requires manual input to reflect evolving site conditions, which is labour-intensive and prone to error [7]. Conversely, computer vision captures dynamic visual data but remains vulnerable to occlusion, lighting variations, and a lack of spatial context, particularly in visually repetitive construction environments.

Progress monitoring frameworks typically quantify construction status using metrics such as percentage completion of structural assemblies, detection of installed components relative to planned quantities, or identification of deviations from expected construction sequences. Some approaches incorporate schedule information to evaluate temporal progress, while others assess spatial completeness by comparing detected elements against BIM geometry. Despite these developments, reliable association between visual observations and corresponding BIM elements remains challenging in visually complex or partially completed environments, particularly where systems rely on manual camera registration or fixed viewpoint assumptions. Table 1 summarises representative studies on computer vision and BIM integration for construction progress and quality monitoring. Prior work varies in its reliance on spatial registration and sensing requirements, as well as in the extent to which BIM is utilised during inference. The proposed framework explores the use of BIM-based semantic constraints with monocular video input without requiring metric registration.

Table 1. Comparison of selected studies integrating vision-based monitoring with BIM for construction progress assessment.

Study	Sensor / Input	BIM dependency type	Registration required	Output	Scope
Golparvar-Fard et al. [44]	Unordered photographs	Geometric comparison against BIM geometry	Yes	Progress deviation maps	Dense image sets required; no semantic element assignment
Wei et al. [20]	Handheld stereo camera (ZED 2)	SLAM-based BIM registration for progress mapping	Yes	Area-level progress in BIM	Stereo camera and SLAM setup required

Elrifaee & Zayed [26]	IoT depth and RGB camera	BIM-linked safety zone contextualisati on	Yes	Real-time safety zone occupancy	Dedicated sensing infrastructure required
Kulinan et al. [47]	CCTV cameras	BIM 3D space is used for worker localisation	Yes	Worker risk status in BIM	Perspective calibration against BIM reference points is required
Pal et al. [49]	UAV, 360° and fisheye camera images	4D BIM as a geometric reference for point cloud registration	Yes	Activity-level completion percentages with BIM visualisation	3D reconstruction required each monitoring round; low-light and occlusion affect accuracy
Hsieh et al. [48]	Multi-angle IP surveillance cameras	BIM is used for camera-to-model perspective alignment	Yes	Component-level construction phase classification with BIM visualisation	Camera reconfiguration required when site conditions change; RC structural phases only
Chua & Cheah [40]	Smartphone and publicly sourced images	As-planned column/storey counts for completion check	No	Building-level completion percentage	Repetitive multi-storey PPVC; single window class; single images
Proposed framework	Monocular smartphone video	BIM semantic constraints for element assignment	No	Element-level presence verification in BIM	Single structural typology; symmetric façades require further validation

Collectively, previous BIM-CV progress monitoring studies demonstrate the value of combining visual sensing with planned BIM information. However, most existing frameworks use BIM primarily as a passive reference for post-detection comparison, or they depend on fixed viewpoints, hardware-assisted localisation, depth sensing, point-cloud reconstruction, or explicit 2D–3D camera pose registration. These assumptions limit deployability in live construction environments, particularly for repetitive structural systems such as CFS framing, where many elements have similar visual appearances and may be repeatedly observed across sequential frames. Where BIM has acted as an active constraint, the constraint has typically been geometric: Braun et al. [45], for instance, combined BIM-based geometric constraints and technological precedence relationships with photogrammetric point clouds to infer the existence of occluded components, but this requires metric reconstruction and registration. Limited work has examined how BIM-derived semantic relationships, such as element type, planned spacing, adjacency, sequence, and quantity, can actively constrain the interpretation of computer vision outputs. In contrast, the proposed framework requires only a standard smartphone camera and pyRevit, with no depth sensing, point-cloud reconstruction, or metric camera registration. By avoiding specialised sensing hardware and associated deployment requirements, the framework offers a lower-barrier approach to element-level presence verification in residential construction.

Chapter 3

Methodology

This chapter presents the methodology of the proposed framework. The framework takes as input a monocular video of a building exterior and a corresponding BIM model and outputs door and window observations assigned to BIM element IDs together with wall-panel coverage, visualised in Revit. It comprises four stages: (1) extracting a semantic representation of each building face from the BIM model (Section 3.1); (2) processing the video stream through object detection and confidence filtering (Section 3.2); (3) using visual odometry to associate detections across frames and segment the sequence by façade (Section 3.3); and (4) matching the accumulated observations to BIM faces and assigning detections to planned elements (Section 3.4). The overall workflow is shown in Fig. 1.

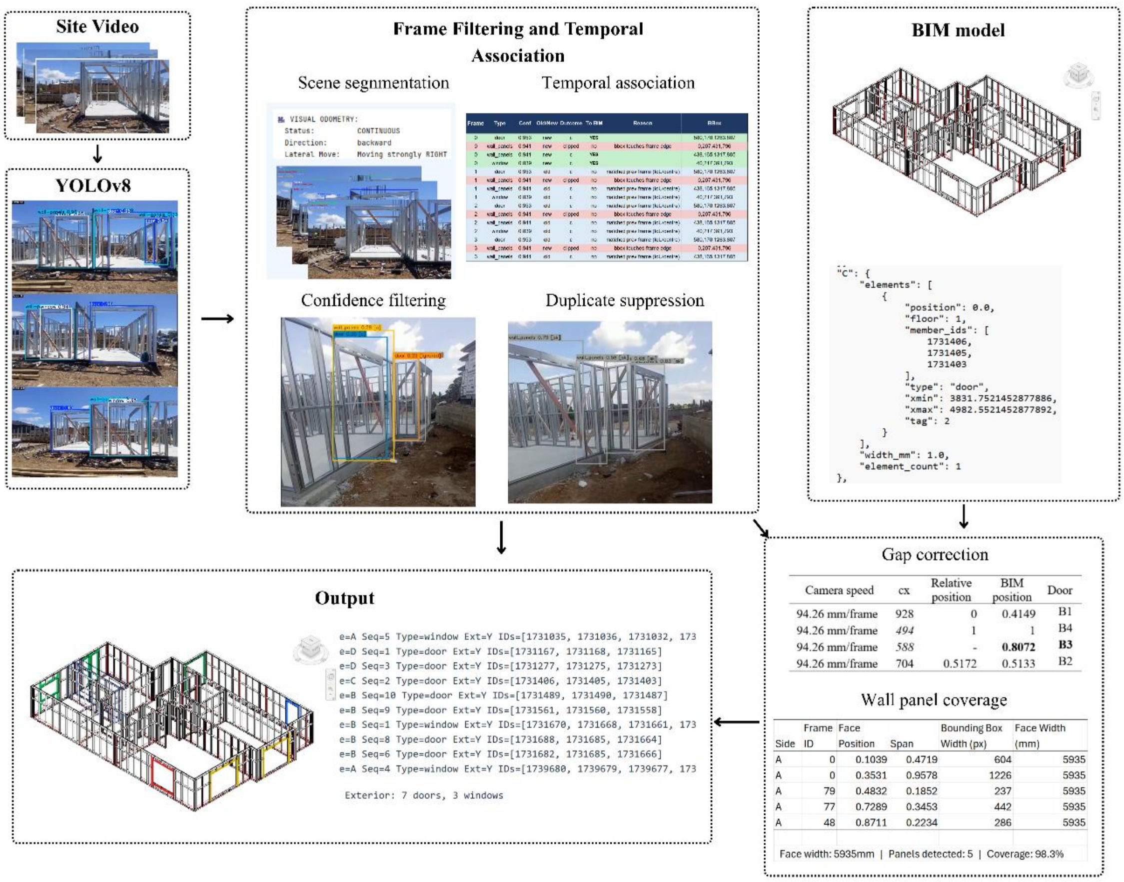


Fig. 1. Overall workflow of the proposed framework

3.1. Scene representation in BIM

BIM serves two primary functions in this framework: (1) as a data source, providing structured information about building elements and their locations to aid in interpreting site images, and (2) as a visualisation tool, which highlights detected objects from the framework.

To enable interaction between the proposed framework and the BIM environment, an extension was developed using pyRevit. pyRevit is an open-source scripting environment that allows Python-based tools to be embedded directly within Revit's interface [50].

The extension is organised into six functional modules, which are as follows: (1) collection of wall, door, and window openings; (2) extraction of their spatial positions within the model; (3) determination of building sides; (4) association of elements based on adjacency and layout; (5) structuring the extracted information into a semantic representation; and (6) exporting the data into a JSON file.

3.1.1. Semantic fingerprints

To match construction images with BIM data, a representation was needed to encode information from these images in a format compatible with BIM element descriptions. A semantic fingerprint for a given building side is a structured record containing: element types present (door, window, wall panel), their count per type, their normalised spatial positions along the face in the range [0, 1], their sequential left-to-right order, and the floor level. Normalised positions are derived from each element's geometric centre, computed as the midpoint of its xmin and xmax coordinates exported from the BIM model, then scaled relative to the total face width. For example, the fingerprint for exterior Side B of Site 1 describes 5 elements on floor 1 across a face width of 10,455 mm (window at 0.000 tag 1; doors at 0.456/0.530/0.752/0.898, tags 6/8/9/10). Each element also carries the structural member IDs of its constituent framing components, enabling direct mapping back to the BIM model once a match is confirmed. This compact representation is used to distinguish one building face from another and to assign detected elements to specific BIM components by position matching.

The use of a semantic fingerprint is further motivated by limitations inherent in monocular image capture and cold-formed steel construction. Monocular images do not provide absolute depth or scale, which makes early-stage geometric reconstruction unreliable. In CFS assemblies, this issue is compounded by the high density of visually similar members, where attempts to detect individual members can lead to ambiguous matches and increased complexity.

As a result, element selection was limited to components that (1) are commonly present in CFS assemblies, (2) exhibit sufficient visual contrast and structural features for reliable detection at image scale, and (3) exist as explicit objects within BIM models. Based on these criteria, three element classes were selected: doors, windows, and wall panels.

3.1.2. Element collection and side classification

Building elements were extracted from the Revit model using filtered element collectors scoped to the active view. For models following a structured framing convention, openings were identified by clustering framing members based on their type and role: header plates, king studs, and sill plates, where the presence of a sill plate distinguished window openings from door openings. For models imported via IFC, where element categories may differ, a separate collector identified opening elements directly by category. In both cases, each opening was assigned a sequential position index along its face rather than a Revit element tag, as tag parameters are unreliable across import workflows.

The building footprint $\mathcal{B}$ is derived from the bounding box extents of all wall panels, spanning from the minimum to maximum X and Y coordinates across all panel edges, as shown in Eq. (1).

$$\mathcal{B} = [Xmin,\ Xmax] \times [Ymin,\ Ymax] \tag{1}$$

This boundary is divided into four logical sides, labelled A, B, C, and D, corresponding to the left, bottom, right, and top faces, respectively. Wall panels serve as vertical markers, as they are constrained to a single elevation on a given floor by design. Floor classification uses the bottom Z-coordinates of all wall panels. A split threshold is computed as the median of these values, as shown in Eq. (2), where ${z_\mathrm{i}}^{\mathrm{bot}}$ denotes the bottom Z-coordinate of the i-th wall panel, and N is the total number of wall panels.

$$z_{\mathrm{split}} = median\{\, {z_\mathrm{i}}^{\mathrm{bot}} \,\}_{\mathrm{i=1}}{}^{\mathrm{N}} \tag{2}$$

Wall panels are not counted as discrete elements in the same way as doors and windows. Unlike openings, which are finite and individually identifiable and verified against known BIM element quantities, wall panels carry no corresponding BIM geometry. Instead, panel coverage is assessed by mapping confirmed panel detections onto a normalised face coordinate system, as described in Section 3.4.2.

3.1.3. Export & structuring

For each classified element, a compact set of attributes is exported, including element name, tag, side, and floor. These per-element attributes are aggregated into the side-level semantic fingerprint defined in Section 3.1.1, characterising how each building side is expected to appear within site images. The structured data is exported in JSON format.

3.2. Detection pipeline

3.2.1. Vision model

Objects captured in a construction environment are subject to partial visibility, occlusions, viewpoint variation, and fluctuating illumination. Such conditions are well-known challenges in computer vision and reduce the reliability of full geometric reconstruction or strict pixel-level correspondence [12,51].

As the primary objective is to determine the presence of assembly elements, a detection model was preferred over segmentation; while models such as Mask R-CNN can produce pixel-level masks and handle irregular shapes, they incur higher computational cost and require extensive annotation effort [52]. In the context of cold-formed steel assemblies, where wall panels, doors, and window openings are mostly rectangular and individually distinguishable at image scale, bounding-box detection provides sufficient information for both discrete element counting and coverage assessment while maintaining efficiency. YOLOv8, being widely used in construction-related computer vision tasks, was selected for its real-time detection speed and accuracy [52,53]. However, to evaluate the trade-off between inference speed and localisation accuracy, single-stage YOLO variants and the two-stage Faster R-CNN were also included in the comparative evaluation described in Section 4.1.

The dataset used to train the custom YOLOv8 model was collected from site images captured at active residential construction projects in Glenview, Hamilton, New Zealand. Images and videos were obtained during site visits using a smartphone camera. The detection model was trained on three classes: wall panels, doors, and windows. The training dataset included synthetic data and images from residential CFS construction sites, with Site 1 fully withheld and used as an independent case study in Chapter 4. Additional examples from partially completed structures are included in Chapter 3 to illustrate individual framework components. The training dataset consisted of 841 images, with an image-level split using Roboflow into training (631, 75%), validation (126, 15%), and test (84, 10%) subsets. Preprocessing steps included auto-orientation, export resizing to 800×800 pixels in Roboflow, and contrast adjustment; during training the detector input was rescaled to 960×960 pixels (Section 4.1). Data augmentation introduced variability in scale, colour, brightness, motion blur, and grayscale to simulate realistic site conditions. Augmentations expanded the training set to 3,155 images.

In practice, doors and windows are less frequent in typical CFS assemblies than wall panels, creating class imbalance in datasets. To partially offset this class imbalance, a total of 120 synthetic images were generated from BIM models using Bonsai [54], an approach shown to improve construction vision-model training when real data are scarce [55], with varied backgrounds and lighting conditions [56,57], resulting in the following counts: wall panels: 4,460 instances; windows: 1,110 instances; doors: 2,440 instances. Despite this, the model occasionally confused door openings with window openings due to their similar appearance in CFS houses, mitigated by the same-region conflict resolution step described in Section 3.2.2. The synthetic rendering workflow and example outputs are shown in Appendix B.

3.2.2. Confidence filtering and cross-class suppression

Raw YOLO detections vary considerably in reliability depending on occlusion, viewpoint, and lighting conditions. Confidence filtering addresses the question of whether a single detection is trustworthy enough to be counted, independently of whether the same object has been seen before. A high-confidence detection seen only once is accepted, while a low-confidence detection seen many times may still be rejected if it never clears the confidence threshold.

Three confidence tiers were applied. Detections with a confidence score at or above 0.70 were accepted immediately, subject to boundary conditions described below. Detections falling between 0.30 and 0.70 were classified as ambiguous and required temporal association: the same element had to be re-observed in a later frame at a consistent position, assessed by the standard association criteria (IoU above 0.40, or normalised centre distance below 0.08 in both axes), with the re-observation itself reaching at least the 0.30 confidence floor. Re-observed ambiguous detections were promoted to confirmed status and counted once. Detections below 0.30 were discarded entirely.

Threshold values were selected through empirical calibration on the training video sequences. The confidence floor of 0.30 reflects the lower bound below which detections were consistently unreliable across all site conditions observed; the high-confidence threshold of 0.70 was chosen to avoid requiring temporal validation for clearly visible elements. The IoU threshold of 0.40 for temporal association was selected to balance sensitivity to viewpoint shifts and resistance to false merging of adjacent elements. The centre-distance threshold of 0.08 (normalised) corresponds to approximately 100 pixels on a 1280-wide frame, which is sufficient to track a stationary element under mild camera shake. The two-tier design was preferred over a single threshold because video supplies temporal redundancy: a borderline detection seen once can be deferred for confirmation by a later observation rather than discarded or admitted outright. A single threshold sacrifices either recall (if set high) or precision (if set low); the two-tier scheme defers that trade-off to evidence.

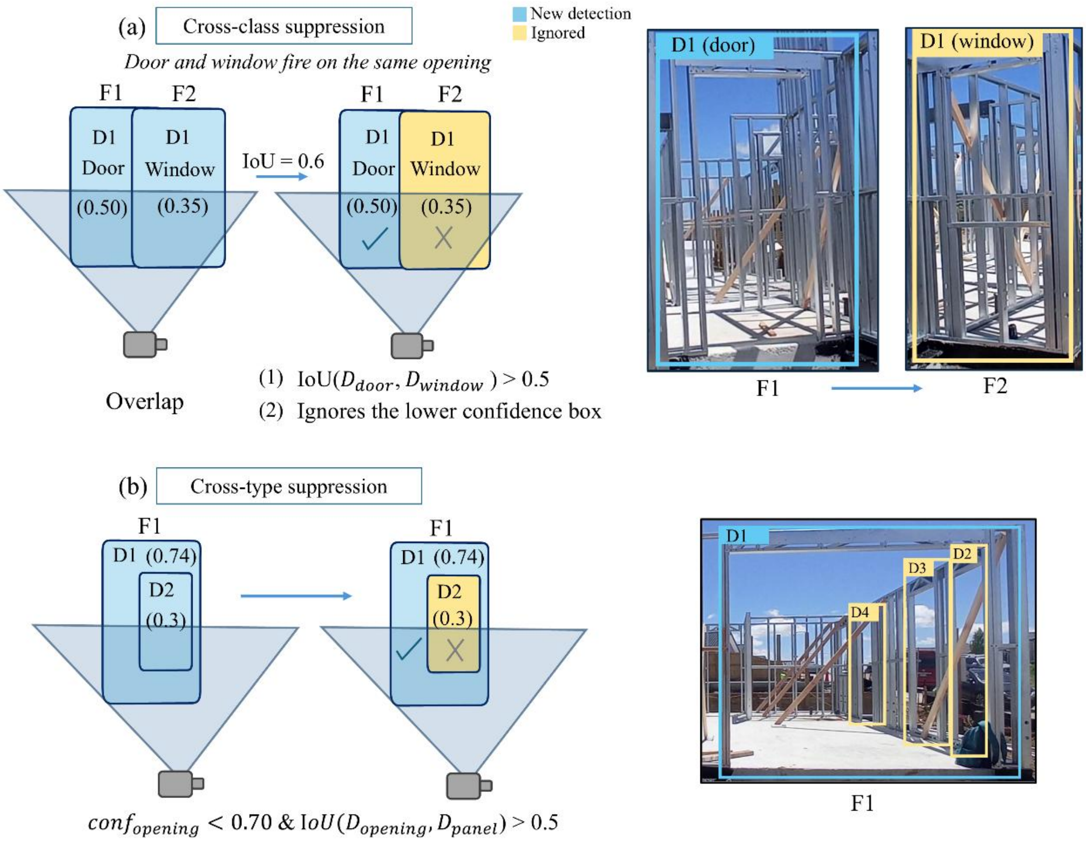


Fig. 2. Same-region conflict resolution: (a) door–window overlap on a single opening; (b) cross-type suppression of a low-confidence opening overlapping a wall panel.

Before temporal association, a same-region conflict resolution step was applied to resolve conflicting detections within the same frame. Two checks were performed. First, when door and window bounding boxes overlapped with an IoU greater than 0.50, both detections were assumed to represent the same physical opening, and the lower-confidence detection was suppressed before tracking.

Second, opening detections below the 0.70 high-confidence threshold that spatially overlapped a wall panel detection were suppressed, on the basis that the conflicting spatial evidence indicated a likely misclassification of the panel as an opening; high-confidence opening detections were exempt. In both cases, suppression occurs before temporal association and BIM matching; suppressed detections are excluded from the element count forwarded to the fingerprint matcher. These two checks are illustrated in Fig. 2(a) and (b).

Bounding box position relative to the image boundary introduced a further classification level. Detections whose bounding boxes touched or extended beyond the image boundary were classified as clipped and excluded from counting entirely, as they represent incomplete observations. As shown in Fig. 3 (c), clipped detections are suppressed before reaching the tracker.

(a) (b) (c)

Fig. 3. Filtering outcome examples from Site 1: (a) nested detections; (b) all elements skipped by duplicate suppression; (c) clipped door detection.

3.3. Visual odometry

3.3.1. Motion estimate

VO was employed to estimate relative camera motion between consecutive frames, supporting the temporal association of static construction elements without requiring absolute pose determination. A core challenge in processing sequential construction site images is that the same physical elements, such as wall panels, doors, or windows, will appear in multiple consecutive frames as the camera moves along the side. Without a mechanism to track continuity between frames, each appearance would be counted as a new detection, producing inflated element counts. The camera traversal is illustrated in Fig. 4.

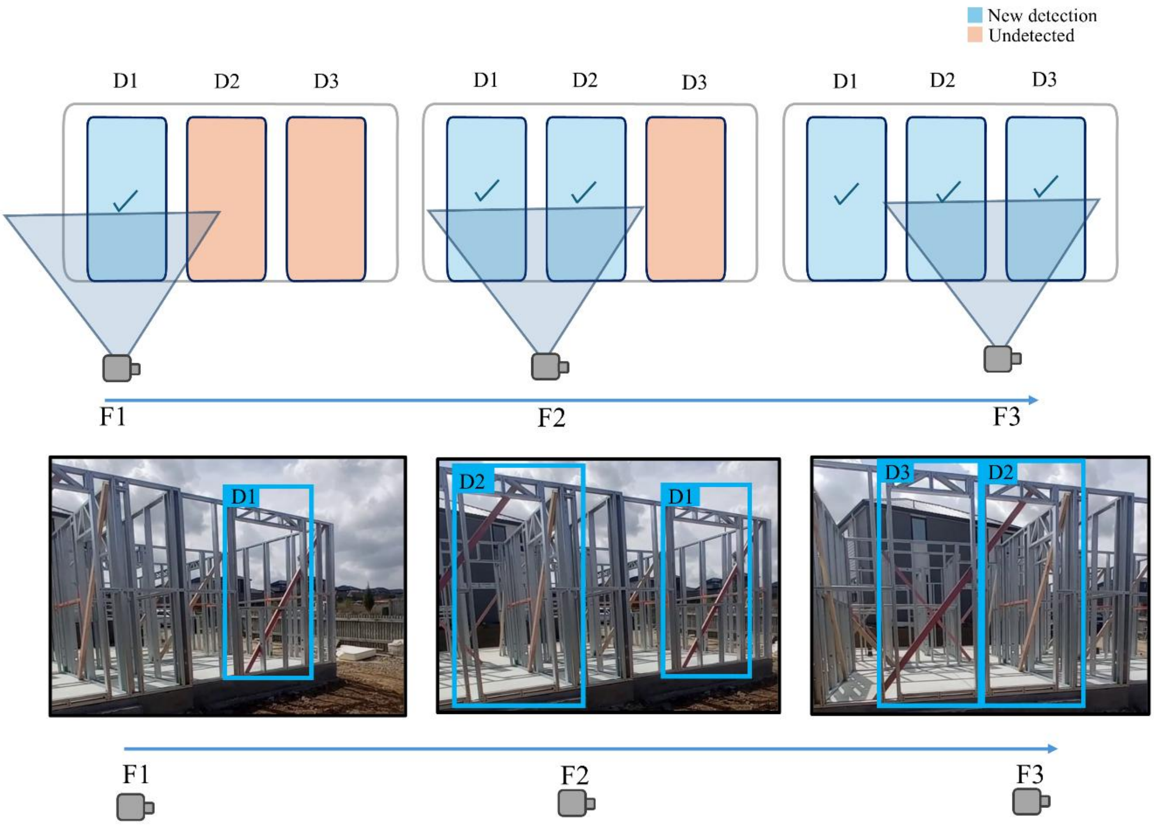


Fig. 4. Camera capture across frames: top, detection footprints per frame, bottom, corresponding site images.

Oriented FAST and Rotated BRIEF (ORB) features were used as the basis for motion estimation. ORB is a computationally efficient feature detector and descriptor that identifies distinctive keypoints in an image and encodes their local appearance in a compact binary form. ORB features were extracted from each grayscale frame and matched across consecutive images using a brute-force Hamming-distance approach. The Lowe's ratio test was then applied to filter ambiguous matches, retaining only correspondences in which the best match was closer than the second-best, $d_1 < 0.75 * d_2$.

Camera intrinsic parameters were estimated using Zhang's method [58] with an 8×11 checkerboard, yielding an RMS reprojection error of 1.16 px. Camera calibration is a standard one-time procedure for a given device and does not need to be repeated across site visits [41]. The resulting calibration accuracy is sufficient for relative pose estimation, as VO is used solely for motion continuity detection and duplicate suppression rather than metric localisation. The camera matrix K is shown in Eq. (3).

$$K = \begin{bmatrix} 3093.03 & 0 & 2031.77 \\ 0 & 3088.70 & 1493.09 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

From the set of good matches, the essential matrix was computed using RANSAC-based estimation, which encodes the geometric relationship between corresponding points in two views. Camera pose was then decomposed from the essential matrix to obtain relative rotation and translation. The inlier ratio, defined as the proportion of matches geometrically consistent with the estimated pose, was used to assess the reliability of each frame's pose estimate. The minimum match count of 20 and the inlier ratio threshold of 0.30 for VO quality assessment follow established practice for ORB-based essential matrix estimation under moderate motion. VO scene segmentation examples are shown in Table 2.

Table 2. Visual-odometry motion assessment examples, illustrating how the framework classifies continuous lateral camera movement and detects sequence breaks under partial-completion conditions.

Frame description	Original frames	VO estimated motion	VO output frame
Initial frame		Initial frame	
Camera translates to the right		Continuous lateral: strongly right	
Camera translates to the right with tilt		Continuous lateral: strongly right	
Scene change		Sequence break Not enough matches	

The lateral component of the translation vector was used to classify camera movement direction and magnitude. The full pipeline is summarised in Fig. 5. Sequence breaks are triggered by two conditions: a pixel-difference spike exceeding a threshold of 30 (Step 1) and a feature match collapsing below 20 good matches (Step 2), before pose estimation. Step 4 addresses a situation where even when sufficient matches exist, they may be geometrically inconsistent. The inlier ratio quantifies the proportion of geometrically consistent matches relative to all retained matches, and a low ρ indicates that RANSAC could not find a coherent essential matrix despite sufficient matches. Step 5 records the lateral translation magnitude, while Step 6 determines the alignment grade from ρ and the number of good matches.

```
Input:   I_t (current frame),  I_{t-1} (previous frame)
Output:  status ∈ {INIT, CONTINUOUS, SEQUENCE_BREAK},
         lateral magnitude,  inlier ratio ρ,  alignment grade
BEGIN
Step 0: First-frame initialisation
    if prev_des == null then
        return INIT
    end if
Step 1: Pixel difference (scene change detection)
    Compute pixel_diff between I_t and I_{t-1}  (downscaled 160 × 90)
    if pixel_diff > 30 then
        // abrupt scene change detected
        return SEQUENCE_BREAK
    end if
Step 2: Feature extraction and matching
    Extract ORB features from I_t and I_{t-1}
    Match descriptors using brute-force Hamming distance
    Apply Lowe's ratio test  (d_1 < 0.75 × d_2)  →  M_good
    if |M_good| < 20 then
        // insufficient geometric overlap
        return SEQUENCE_BREAK
    end if
Step 3: Pose estimation
    Estimate essential matrix E from M_good using RANSAC
    Recover rotation R and translation vector t from E
Step 4: Geometric consistency check
    Compute inlier ratio:
        ρ = |M_inlier| / |M_good|
    if ρ < 0.30 then
        // unreliable geometric structure
        return SEQUENCE_BREAK
    end if
Step 5: Motion characterisation
    lateral_magnitude ← |t[x]|
Step 6: Alignment quality grading
    if ρ > 0.70  and  |M_good| > 50 then
        grade ← EXCELLENT
    else if ρ > 0.50  and  |M_good| > 30 then
        grade ← GOOD
    else if ρ > 0.30 then
        grade ← FAIR
    else
        grade ← POOR
    end if
Step 7: Return status
    return CONTINUOUS,  lateral_magnitude,  ρ,  grade
END
```

Fig. 5. Pseudocode for the visual-odometry-based sequence-break detection algorithm using pixel-difference spikes and feature-match collapse, with geometric-consistency checks used for pose-quality grading.

3.3.2. Temporal association and duplicate suppression

Where confidence filtering concerns detection quality, temporal association concerns object identity, determining whether a detection in the current frame represents a physical element already recorded in a previous frame, or one that is genuinely new. This distinction is what prevents the same wall panel, seen across ten consecutive frames, from being counted ten times.

Each frame's accepted detections were compared against a pool of previously tracked objects. Two criteria were applied in order. First, bounding box overlap: if the Intersection over Union (IoU) between a current detection and a tracked object of the same class exceeded 0.4, the detection was considered a re-observation of that object. Second, if IoU was insufficient, for example, due to viewpoint shift or partial occlusion, a centre distance fallback was applied, accepting a match if the normalised centre coordinates differed by less than 0.08 in both axes. The two criteria are deliberately disjunctive rather than conjunctive: IoU captures the common case of similar viewpoints in adjacent frames, while centre distance handles viewpoint shifts and nested detections (e.g. an interior door visible through an exterior opening) where bounding boxes share little area but nearly coincide in normalised position. An example of a detection matched across two consecutive frames is shown in Fig. 6.

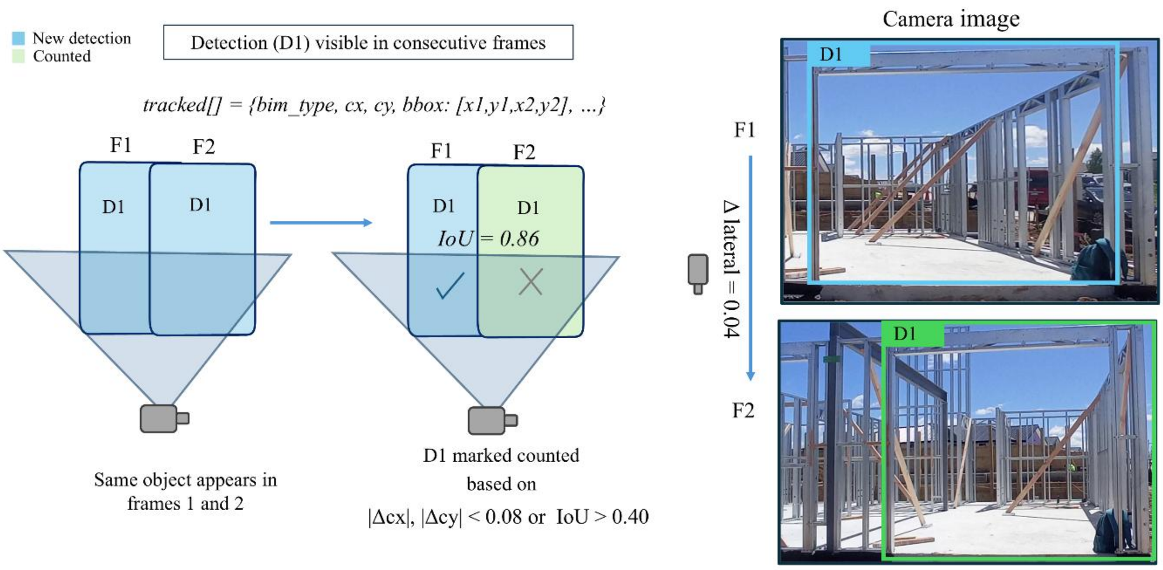


Fig. 6. Temporal association example showing element tracking across two frames.

Detections satisfying either criterion were marked as repeated observations and withheld from the BIM matcher, unless the matched object was previously ambiguous, in which case the re-observation promoted it to confirmed status. Detections matching neither criterion were registered as new elements and forwarded to the BIM matcher. Nested detections, where a smaller bounding box falls within a larger one, are also resolved at this stage. As shown in Fig. 3 (a), interior door opening visible through an exterior door opening can produce overlapping detections, which the association step treats as re-observations of the same element rather than distinct new detections; because a nested box shares little area with the enclosing one, these cases are typically resolved by the centre-distance fallback rather than the IoU criterion.

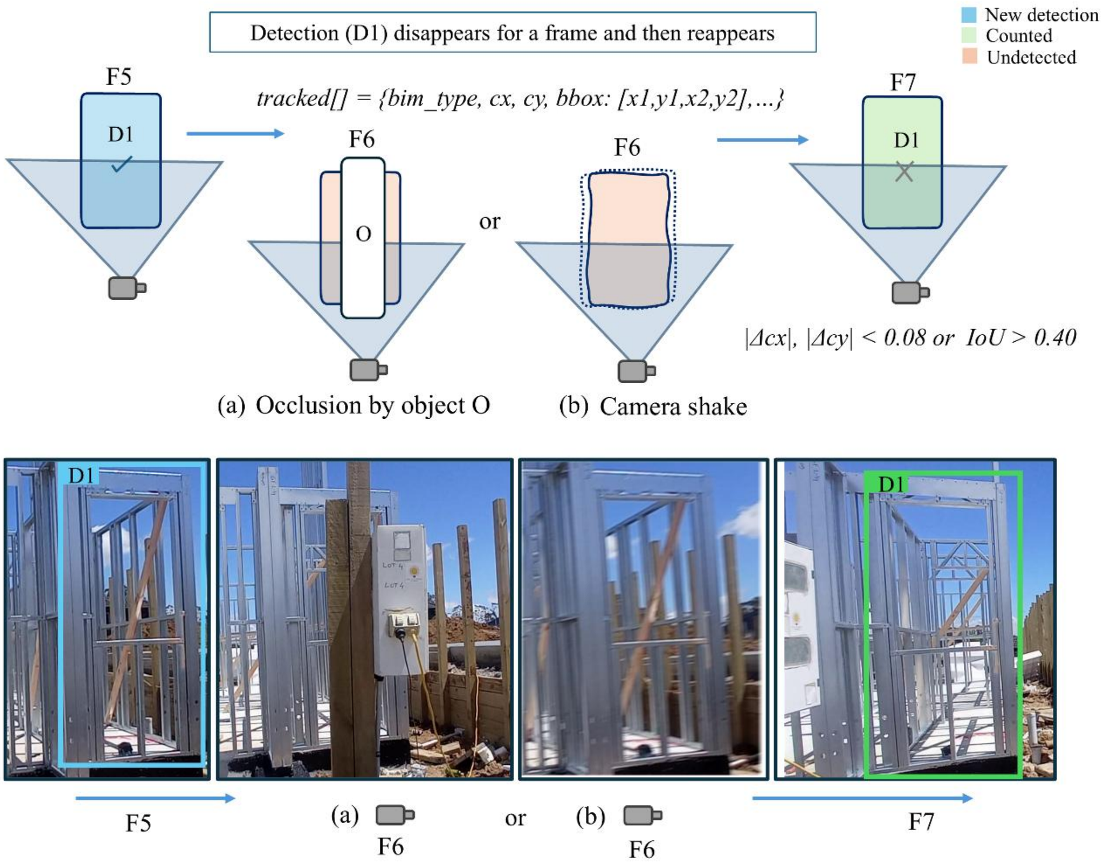


Fig. 7. Temporal tracking example showing a temporary detection loss due to occlusion or camera shake, resolved through temporal association upon reappearance.

Temporal association handles frame-to-frame continuity when the camera moves slowly, but a separate mechanism was needed to prevent duplicate counting when the camera is effectively stationary or makes only minor lateral movements. In such cases, the same element may be detected repeatedly across frames without satisfying the IoU or centre-distance thresholds, for example, when a slight camera shake shifts the bounding boxes just enough to break association, as illustrated in Fig. 7.

When the camera revisits a previously recorded façade, confirmed elements may be re-detected and counted again. Duplicate suppression prevents this by suppressing detections that match an already-confirmed element's position and class. As illustrated in Fig. 3 (b), all elements in a frame can be suppressed when the framework identifies the scene as a spatial duplicate of a previously processed segment. To prevent duplicate counting during stationary or near-stationary camera movement, cumulative lateral camera displacement was tracked within each sequence using the absolute lateral component of the translation vector from visual odometry. Each counted detection was assigned a position key defined by its class and normalised centre coordinates. While cumulative lateral displacement remained below 0.40, new detections whose class and centre coordinates matched an existing position key were suppressed as sequence duplicates rather than registered as new elements.

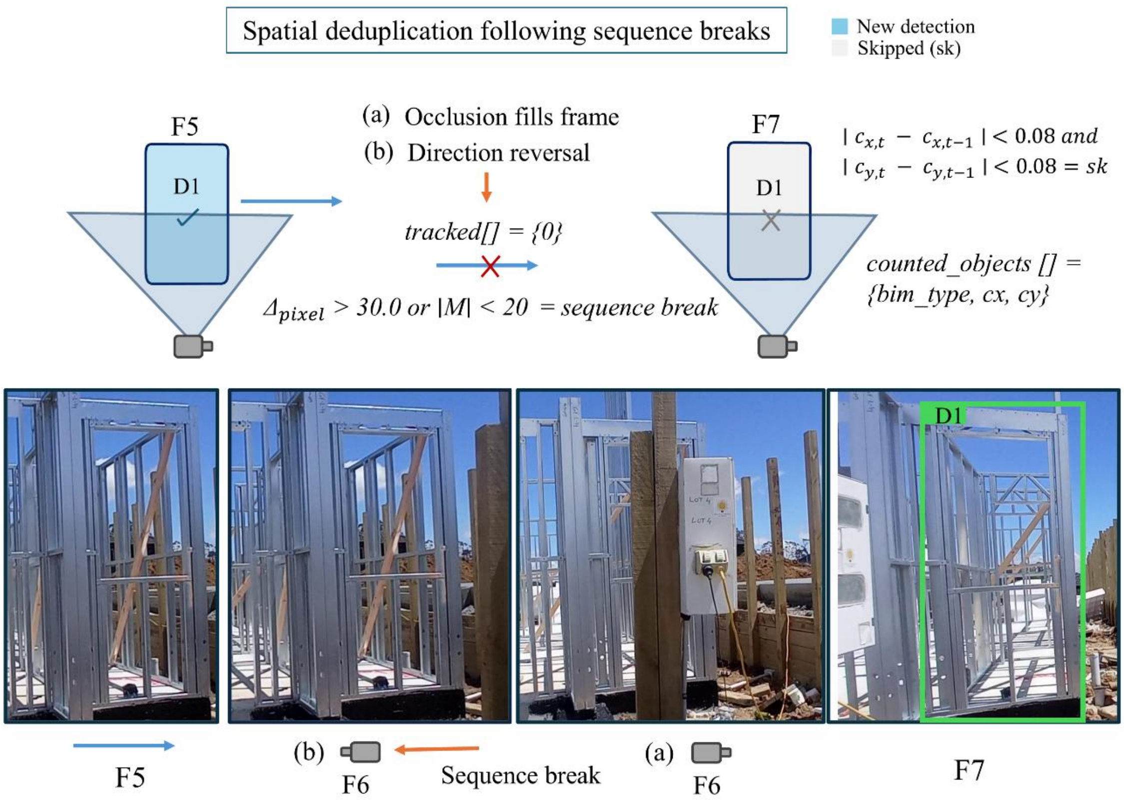


Fig. 8. Example of spatial deduplication after a sequence break caused by tracking loss.

Table 3 shows how detections are filtered across the pipeline, grouped into frame ranges with similar behaviour. Each stage reports counts with status labels: (n) new object, (c) counted to BIM, (a) ambiguous, (o) previously counted, (pr) promoted from ambiguous. Across the sequence, 23 raw detections are reduced to 6 unique elements after filtering, and temporal tracking removes duplicates. Duplicate suppression therefore operates at two ranges: the sequence-duplicate check is short-range and is cleared at every sequence break, whereas the registry of confirmed element positions persists for the entire video, suppressing re-detections of an already-counted element (same class, normalised centre within 0.08 in both axes) even after tracking has been reset at a façade transition. Fig. 8 illustrates an example of registry-based suppression. The raw YOLO and confidence-filtering counts are not additive across frames because each frame re-detects objects independently, resulting in repeated sightings of the same objects across all frames rather than a true cumulative count.

Table 3. Progressive filtering of detection across a frame sequence, showing how confidence filtering, temporal association, and duplicate suppression reduce repeated observations before BIM matching.

Frames	Raw Detections	Confidence Filtering	Temporal Association	Duplicate Suppression
0-53	Door: 1 Window: 1 Wall Panel: 2	Door: 1 Window: 1 Wall Panel: 2	Door: 1 (n) Window: 1 (n) Wall Panel: 2 (n, c)	Door: 1 (n) Window: 1 (n) Wall Panel: 2 (n, c)
54-201	Door: 1 Window: 1 Wall Panel: 1	Door: 1 (a) Window: 1 Wall Panel: 1	Door: 1 (a) Window: 1 (o) Wall Panel: 1 (o)	Door: 1 (a) Window: 1 (o) Wall Panel: 1 (o)
202	Door: 1 Window: 1 Wall Panel: 1	Door: 1 Window: 1 Wall Panel: 1	Door: 1 (pr) Window: 1 (o) Wall Panel: 1 (o)	Door: 1 (pr) Window: 1 (o) Wall Panel: 1 (o)
203-209	Window: 1 Wall Panel: 1	Window: 1 Wall Panel: 1	Window: 1 (o) Wall panel: 1 (o)	Window: 1 (o) Wall panel: 1 (o)
210-217	Door: 2 Window: 1 Wall Panel: 2	Door: 2 Window: 1 Wall Panel: 1 Wall Panel: 1 (a)	Door: 2 (o) Window: 1 (o) Wall Panel: 2 (o, a)	Door: 2 (o) Window: 1 (o) Wall Panel: 2 (o, a)
218-307	Door: 1 Window: 2 Wall Panel: 3	Door: 1 Window: 2 Wall Panel: 3	Door: 1 (o) Window: 2 (o, c) Wall Panel: 3 (o, pr)	Door: 1 (o) Window: 2 (o, c) Wall Panel: 3 (o, pr)
Ground Truth: Door: 2 Window: 1			*Count: 6* Door: 2 Window: 1 Wall Panels: 3	*Count: 6* Door: 2 Window: 1 Wall Panels: 3

Note: Wall panel ground truth is not reported as a discrete count; coverage is assessed from inter-detection gaps against total face width.

3.3.3. Scene segmentation

Scene segmentation partitions the image sequence into contiguous façade observations by identifying transition points between building sides. Two conditions triggered scene segmentation. The primary condition was a pixel-difference spike: the mean absolute pixel difference was computed between consecutive frames, downscaled to 160 × 90 pixels. During normal lateral motion, values remained stable (5–20), whereas corner transitions produced sharp increases. The pixel-difference break threshold of 30 was determined by inspecting the distribution of inter-frame differences across all corner transitions in the calibration footage, where values during normal lateral motion remained below 20, and corner transitions consistently exceeded 35.

A secondary condition was inherited from the visual odometry pipeline: a match-count failure, in which fewer than 20 good feature matches were found between consecutive frames, indicating the camera had moved too far for reliable pose estimation. A low inlier ratio (below 0.30) additionally downgraded the pose-quality grade to POOR but did not trigger a break. Both break conditions were treated identically: a break was triggered, all tracked objects were discarded, accumulated lateral displacement was reset to zero, the sequence signature dictionary was cleared, and detection restarted for the new segment. The complete per-frame filtering pipeline is summarised in Fig. 9.

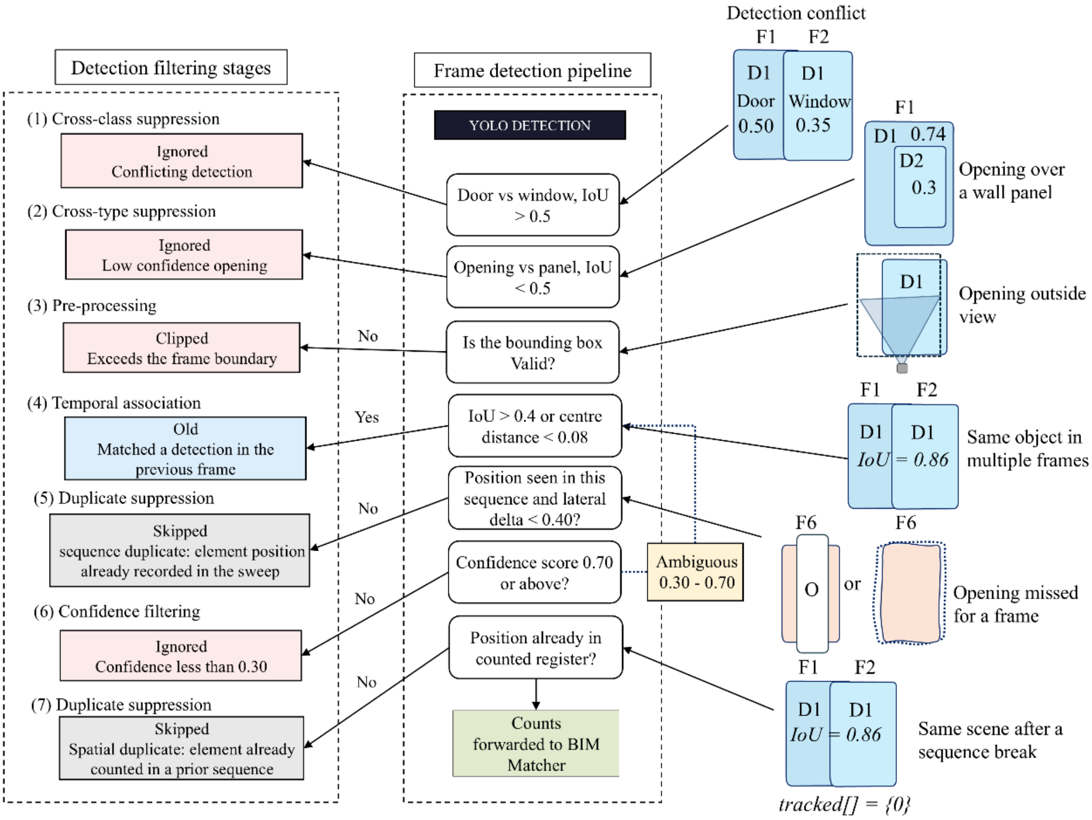


Fig. 9. Per-frame processing pipeline showing detection, filtering, and observation refinement following scene segmentation.

3.4. BIM matching

3.4.1. Side identification

Once a segment had been finalised, its accumulated detection counts were matched to one of the known building faces. The accumulated detections formed the observed element fingerprint for the façade, as described in Section 3.1.1. Once a BIM side was assigned, it was removed from further consideration to prevent multiple segments from being matched to the same BIM side while others remained unmatched. The framework treats geometrically irregular façades, including recessed garage sections or non-rectilinear wall layouts, as part of the same semantic building side when they belong to a continuous exterior face.

Candidate façade sides were ranked lexicographically using the quantities shown in Fig. 10. Priority was given to maximising overlap, followed by minimising excess and then minimising missing. The final match confidence was reported as the overlap normalised by the total number of BIM door and window elements on the candidate side. The full side-fingerprint matching procedure is given in Appendix A (Algorithm 1).

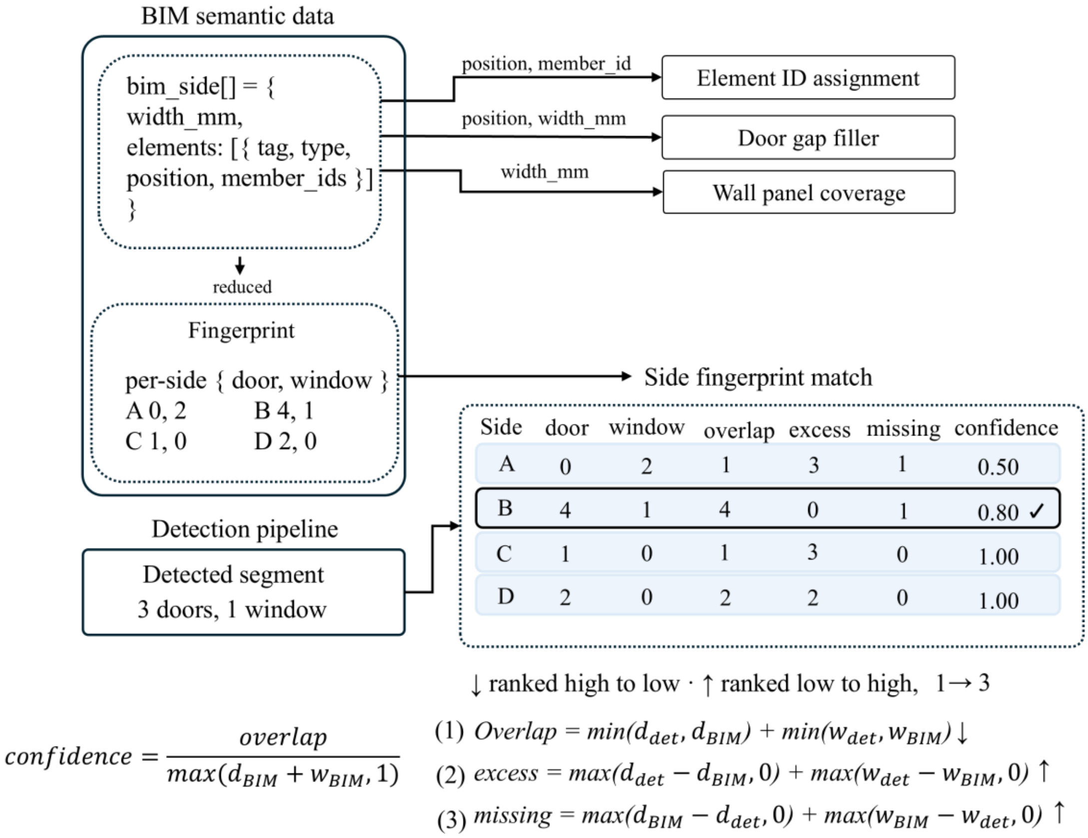


Fig. 10. BIM semantic fingerprint pipeline: BIM semantic data is used across framework modules and is reduced to class-count fingerprints for matching against detections.

3.4.2. Element assignment

Once a segment had been matched to a BIM side, confirmed detections were assigned specific BIM element IDs. Each BIM element in the JSON export contained a normalised position in the range [0, 1], representing its location along the face derived from its geometric centre. Detected elements were converted to normalised positions using pixel centre coordinates and the frame width, with direction flipped for right-to-left movement to maintain consistent ordering.

Detections and BIM elements of the same type were each sorted by normalised face position and paired in order, with each BIM element assigned at most once. A failure mode occurred when detections were missed, causing ordinal misalignment (e.g., the third detection being incorrectly mapped to the third BIM element instead of the fourth). To address this, a gap correction step was applied when the number of detections was lower than the number of BIM elements of the same type: confirmed detections were greedily matched to the nearest unmatched BIM element by position, the unmatched BIM element was identified, and its expected appearance was predicted from the relative positions of the flanking detections, or, where the element lay outside the detected span, from the camera traversal speed estimated from BIM inter-element distances. This recovery process is illustrated in Fig. 11. The gap-filling procedure is provided in Appendix A (Algorithm 2).

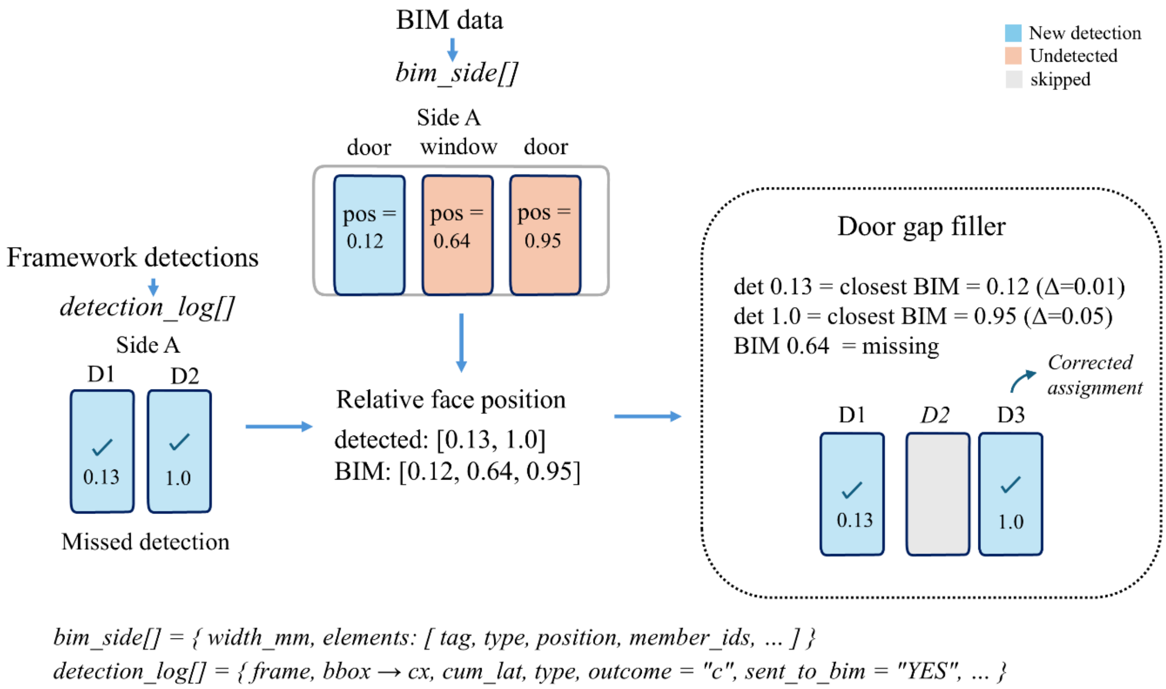


Fig. 11. BIM-guided gap correction example: recovery of a missed door opening by matching against BIM.

Wall panel coverage was assessed by mapping confirmed panel detections onto a normalised face coordinate system (0–1), converting bounding box edges to face positions accounting for camera direction. Overlapping or proximate spans were merged to represent unique physical panels, and coverage gaps were identified only between merged spans, excluding regions before the first and after the last detection. The conversion from per-frame detections to merged spans and gap identification is shown in Fig. 12.

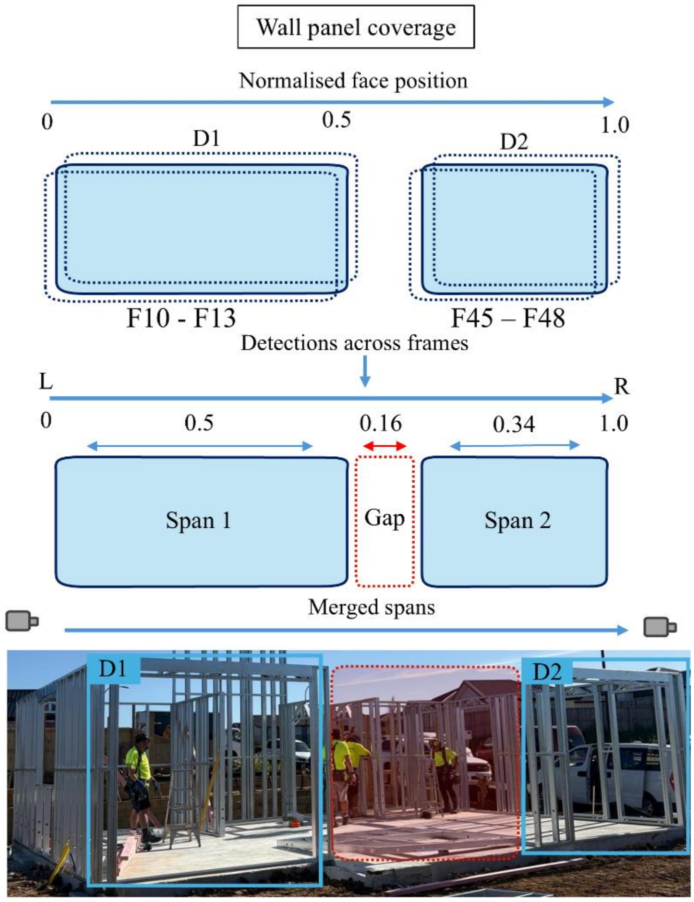


Fig. 12. Wall panel coverage: (top) per-frame detections; (middle) span conversion and gap estimation from relative positions; (bottom) merged-span example showing detected gap.

3.4.3. Visualisation

Detection results were brought into the BIM environment via the JSON output from the framework pipeline, which mapped each confirmed detection to its BIM member IDs, building face, and confidence score. A pyRevit script traversed all structural framing elements in the active Revit view, HP plates, SP plates, and studs, clustered them into door and window openings, and assigned each to its corresponding building face using the side classifications established in Section 3.1.2. Each opening was then colour-coded directly in the model view according to its assigned face: Red for Side A, Green for Side B, Blue for Side C, and Yellow for Side D, with interior openings rendered in darker tones of the same palette. The script applies graphic overrides directly to each element's surface pattern in the active Revit view using the OverrideGraphicSettings API, requiring no manual selection. The entire highlighting process runs within Revit via the pyRevit extension. The extension prints a confirmation of matched openings to the console and applies colour overrides to the model view, giving an immediately readable record of verified, missing, and BIM-corrected elements with their element IDs across each face. Fig. 13 shows the BIM model with and without progress results from Site 1. The exterior of Site 1 is shown in Fig. 14.

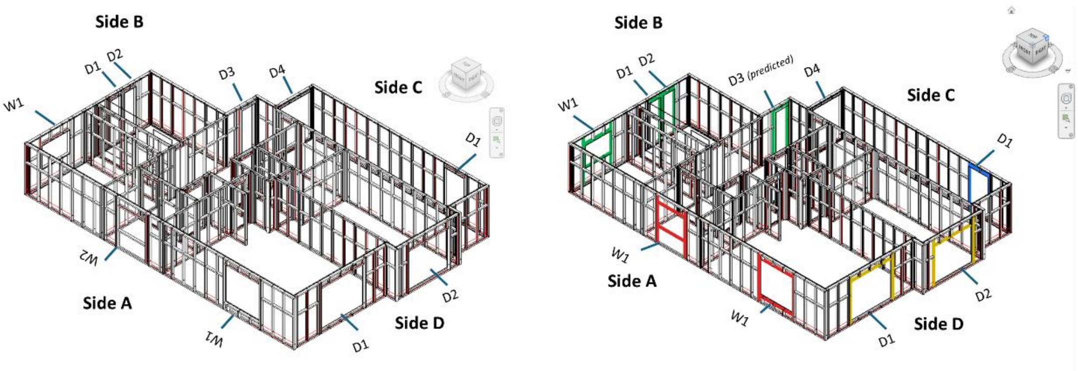


Fig. 13. BIM-based visualisation of matched construction elements, showing the raw BIM model (left) and highlighted framework-verified elements (right).

Fig. 14. Exterior faces of the Site 1 structure, labelled A–D, with detected door openings (D) and window openings (W) annotated by instance.

This chapter has presented the proposed framework methodology. Section 3.1 introduced the semantic BIM representation, addressing Objective 1; Sections 3.2 and 3.3 presented the object-detection and visual-odometry components, addressing Objective 2; and Section 3.4 described the BIM matching, element assignment, and gap-correction procedures, addressing Objective 3.

Together, these components establish the active use of BIM semantics during interpretation and element-level verification. Chapter 4 evaluates the framework on a single-storey cold-formed steel structure with known ground-truth observations.

Chapter 4

Evaluation

The framework is evaluated on three components: detection model selection, scene segmentation, and element-counting accuracy. Evaluation was conducted on a single-storey residential cold-formed steel structure in New Zealand, selected as a controlled proof-of-concept to demonstrate the framework's core mechanisms under known ground truth conditions. Ground truth element counts were established by direct physical inspection before video capture. Each component is assessed independently to isolate the contribution of detection, temporal reasoning, and BIM matching to the overall pipeline performance.

4.1. Detection model selection

To identify the most suitable object detection architecture for construction element recognition, four models were evaluated: YOLOv5m, YOLOv8m, YOLO11m, and Faster R-CNN with a ResNet-50 FPN backbone. All YOLO variants were trained at the medium model scale (m), as this configuration provides a practical balance between representational capacity and generalisation for relatively small datasets.

All YOLO models were trained on a batch size of 16 across dual NVIDIA Tesla T4 GPUs via distributed data-parallel training within the Kaggle cloud computing environment, using an input resolution of 960 by 960 pixels, a maximum of 150 epochs with early stopping (patience = 50 epochs), stochastic gradient descent (SGD) optimisation with an initial learning rate of 0.01 decayed via cosine annealing, and COCO-pretrained weights for initialisation.

Faster R-CNN was trained for 128 epochs under identical input resolution and hardware conditions, using SGD with an initial learning rate of 0.005 decayed via StepLR (step size 30, γ = 0.1) and ImageNet-pretrained backbone weights. Total training loss decreased by 96.7% over the course of training, with the point of sharpest deceleration (elbow) occurring at epoch 47. Of the total measurable loss reduction, 86.7% was achieved within the first 50 epochs, with only 13.3% of incremental improvement across the remaining 78 epochs. These results confirm that the model reached effective convergence well within the training budget. Despite achieving the highest mAP@0.5 among evaluated models (0.894), Faster R-CNN was not selected due to its lower mAP@0.5:0.95 (0.719 versus 0.745 for YOLOv8m) and substantially higher inference latency of 119.4 ms per frame (8.4 FPS), compared to 20.5 FPS for YOLOv8m.

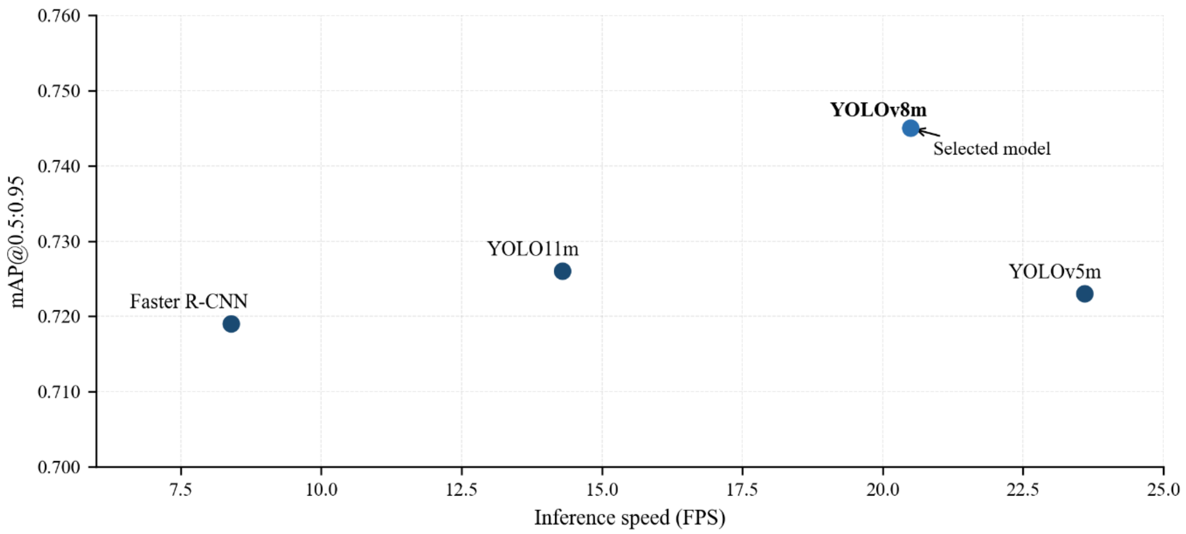


Fig. 15. Inference speed versus mAP@0.5:0.95 for evaluated detection models.

All models were evaluated on the same held-out test set of 84 images using precision, recall, mAP@0.5, and mAP@0.5:0.95. As shown in Table 4, YOLOv8m achieved the best balance of detection performance and inference speed, with an mAP@0.5 of 0.885 and an mAP@0.5:0.95 of 0.745 at 20.5 FPS. YOLOv5m demonstrated the fastest inference speed at 23.6 FPS, though with lower recall (0.788). Faster R-CNN achieved the highest mAP@0.5 (0.894) and strong recall, but at 8.4 FPS is unsuitable for real-time deployment. Based on these results, YOLOv8m was selected as the detection architecture for the proposed framework (Fig. 15). Training and validation loss curves for YOLOv8m are shown in Fig. 16. All three loss components converged stably by approximately epoch 50, indicating consistent generalisation without overfitting.

Table 4. Object-detection architecture evaluation results on the held-out test set, including model size, inference speed, and detection metrics (precision, recall, mAP@0.5, mAP@0.5:0.95).

Model	Backbone	Params (M)	FPS	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv5m	CSP (C3)	25	23.6	0.899	0.788	0.875	0.723
YOLOv8m	CSP (C2f)	25.8	20.5	0.850	0.838	0.885	0.745
YOLO11m	CSP (C3k2, C2PSA)	20	14.3	0.845	0.844	0.862	0.726
Faster R-CNN	ResNet-50 FPN	43.3	8.4	0.848	0.970	0.894	0.719

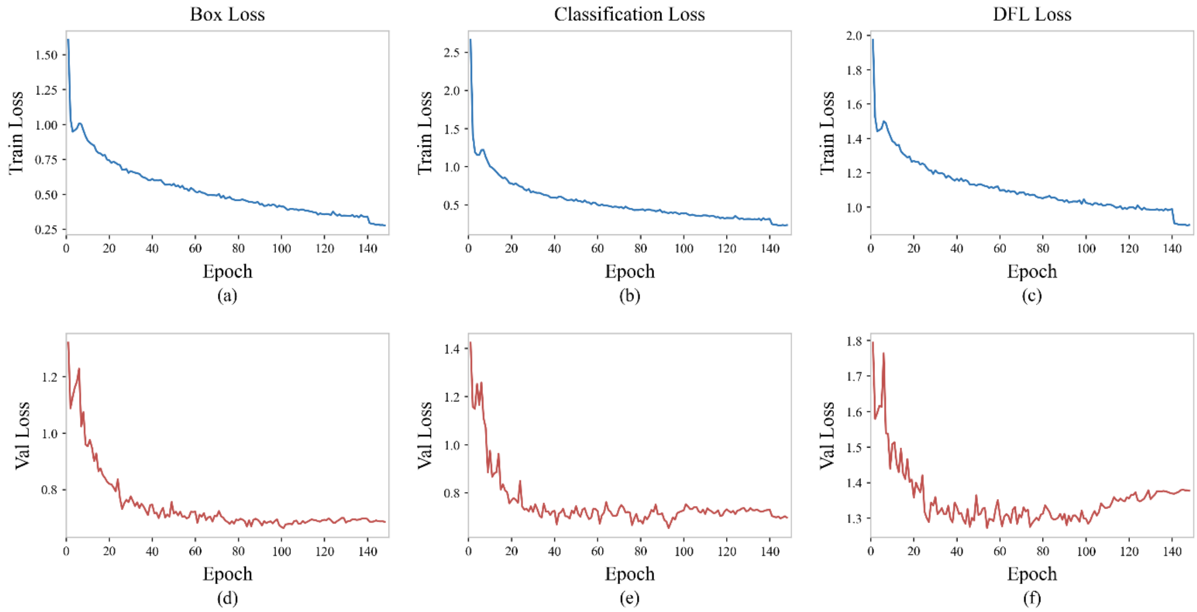


Fig. 16. YOLOv8m training and validation loss curves across 150 epochs: (a) training box loss, (b) training classification loss, (c) training DFL loss, (d) validation box loss, (e) validation classification loss, (f) validation DFL loss.

4.2. Scene segmentation

As shown in Table 5, the segmentation module correctly identified all three transitions, producing four segments with matched BIM sides that correspond to the four building faces. The per-frame pixel-difference trace and the three detected transitions are shown in Fig. 17. Two of the three breaks were triggered by the pixel difference spike condition (values of 95.8 and 36.6, both above the threshold of 30.0). The remaining break was triggered by the feature match collapse condition (15 good matches, below the threshold of 20), corresponding to a more abrupt corner transition where the camera swing was fast enough to disrupt feature tracking entirely.

Table 5. Scene segmentation and side-matching results for Site 1, showing detected break conditions and matched BIM sides for each video segment before gap correction.

Segment	Frames	Break	Framework			Ground Truth		Sides
			Door	*Window*	*Wall Panel*	*Door*	*Window*	
1	0-97	pixel diff: 95.8	0	2	5	0	2	Side A
2	98-238	pixel diff: 36.6	3	1	3	4	1	Side B
3	239-464	ORB matches: 15	1	0	2	1	0	Side C
4	465-982	end	2	0	1	2	0	Side D

Under a penalty-only system, Side B (one missing door, penalty 1) and Side D (one excess door and one excess window, penalty 2) differ by a single point, and the matched window does not affect the ranking, so the separation is fragile. The reward-based system makes the margin explicit: Side B achieves an overlap of 4 (three shared doors and one shared window) against Side D's overlap of 2, ranking Side B first.

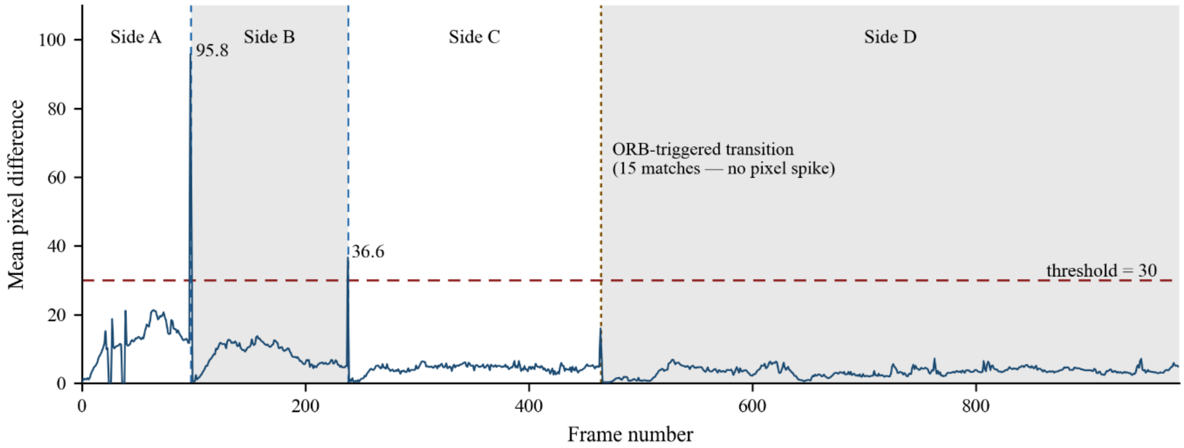


Fig. 17. Scene segmentation based on per-frame mean pixel difference during camera traversal. Pixel-difference peaks indicate façade transitions, with ORB-based matching handling transitions without significant pixel variation.

4.3. Element count

The framework's ability to correctly count unique physical elements as opposed to raw per-frame detection counts was evaluated against ground truth for doors and windows on each face. Because wall panels vary in size, coverage is assessed using the spatial extent of detected bounding box edges, with gaps identified between consecutive panel spans. Across all four building faces, 5 panels on Side A, 3 on Side B, 2 on Side C, and 1 on Side D formed continuous spans, with no gaps detected within the observed detection range.

Table 6. BIM element assignment results for Site 1, showing matched façade side, detected element type, BIM tag, and associated structural member IDs for each verified opening.

No.	Side	BIM Side	Type	Tag	Status	Member ID
1	A	A	window	5	detected	1731035, 1731036, 1731032, 1731034
2	A	A	window	4	detected	1739680, 1739677, 1739678, 1739679
3	B	B	window	1	detected	1731670, 1731668, 1731661, 1731663
4	B	B	door	6	detected	1731682, 1731685, 1731666
5	B	B	door	8	detected	1731688, 1731685, 1731664
6	B	B	door	9	predicted	1731561, 1731560, 1731558
7	B	B	door	10	assigned ID	1731489, 1731490, 1731487
8	C	C	door	2	detected	1731406, 1731405, 1731403
9	D	D	door	1	detected	1731167, 1731168, 1731165
10	D	D	door	3	detected	1731277, 1731275, 1731273

The confirmed detections for each façade, colour-coded by side alongside the corresponding BIM view, are shown in Fig. 18. Table 6 presents the detected elements and their corresponding member IDs. Each tag represents a group of structural members forming a single element, with windows comprising four members and doors comprising three members. On Side B, YOLO misclassified door B3 as a wall panel due to partial occlusion, producing only three confirmed detections where BIM expected four. As shown in Table 7, the BIM-constrained gap correction module estimated a camera traversal speed of 94.26 mm/frame from those three detections and used relative position ordering to correctly assign B1, B2, and B4 before identifying B3 as unmatched. With no door detected within the predicted appearance window at cx 601 and only a low-confidence wall panel present, the system inferred a missing element and inserted B3 with BIM-derived member IDs at the predicted position. Without this correction, B3's ID would have been assigned to the third confirmed detection, leaving B4 unmatched. Two of the three confirmed Side B doors (B1 at 0.441 and B4 at 0.309) fell below the high-confidence threshold and were recovered via the promotion mechanism, which confirms ambiguous detections upon re-sighting.

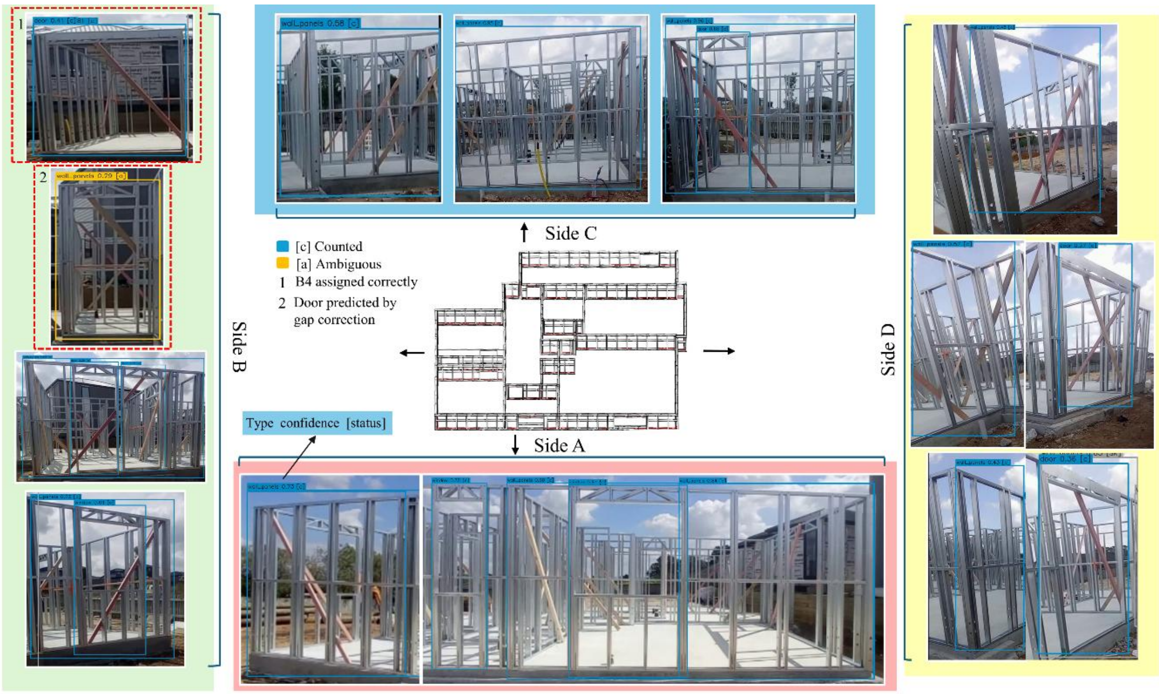


Fig. 18. Façade-wise building element detection results, grouped by façade and colour-coded by side, alongside a BIM view showing the corresponding façades.

Table 7. Gap correction results for a missing door on Side B, Site 1. Frame order reflects detection timing during a right-to-left traverse, with higher face positions encountered first and consistent relative cx ordering.

Frame	Camera speed	cx	Relative position	BIM position	Door	Tag	Member IDs
108	94.26 mm/frame	928	0	0.4558	B1	6	1731682, 1731685, 1731666
157	94.26 mm/frame	704	0.5172	0.5301	B2	8	1731688, 1731685, 1731664
141	94.26 mm/frame	601	-	**0.7520**	**B3**	**9**	**1731561, 1731560, 1731558**
125	94.26 mm/frame	494	1	0.8976	B4	10	1731489, 1731490, 1731487

The raw YOLO pass, which applies no temporal filtering or deduplication, produced detection counts ranging from 11 to 38 per element type, per frame across the sequence, illustrating the scale of inflation that temporal association and spatial deduplication address. All relevant elements were visible in the video sequence; however, one door was not correctly classified by the detector and was recovered through BIM-guided gap correction. At door B3's location the detector produced only a low-confidence wall panel observation, which never reached confirmed status and was therefore not counted. Because the confirmed door count fell one short of the BIM-expected total, the gap correction module predicted the missing door's position and inserted it with BIM-derived member IDs. The net effect is that a location the detector had treated as an ambiguous wall panel is represented as a door in the final BIM output, achieved through suppression and BIM-guided insertion rather than a direct relabelling of the detection. Fig. 19 illustrates detector misclassification at Site 1. Panel (a) shows door B3, initially detected as a wall panel and recovered by the gap correction module. Panel (b) shows a wall panel on Side D misclassified as a door alongside an adjacent ambiguous wall panel.

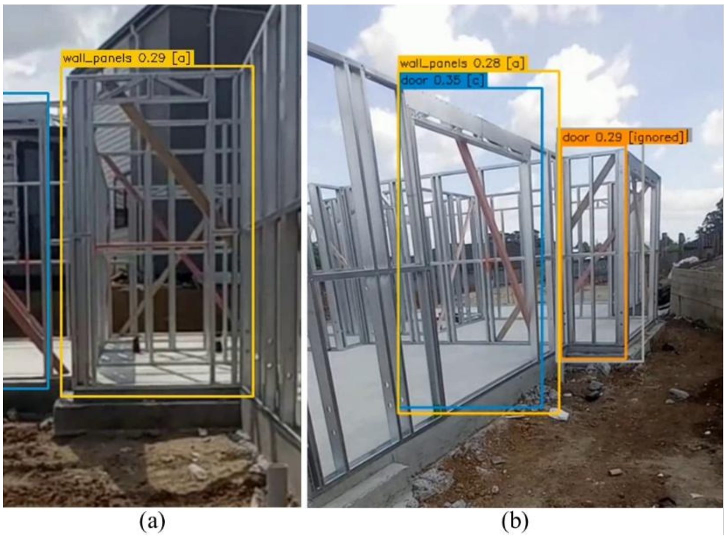


Fig. 19. Detector misclassification examples at Site 1: (a) door B3 initially misclassified as a wall panel; (b) wall panel misclassified as a door.

Total end-to-end processing time for the 983-frame sequence was 462.9 s (2.12 FPS), with YOLO inference accounting for 80.6% of compute (373.1 s). ORB feature extraction and visual odometry accounted for the remainder (89.7 s); temporal association and BIM matching contributed negligible overhead (0.03 s). Timing was obtained on an Intel Core i7-10700 CPU and an NVIDIA Quadro P2200 GPU, and overall throughput is therefore primarily dependent on detection-model inference speed and hardware configuration.

Chapter 5

Discussion

5.1. Baseline comparison and pipeline progression

Table 8 compares the unique element counts produced by the proposed temporal association mechanism with two widely adopted multi-object trackers available natively in the YOLOv8 framework, ByteTrack, which associates every detection box across frames, and BoT-SORT, which combines motion and appearance cues for re-identification. ByteTrack substantially overcounts across all classes, and BoT-SORT undercounts doors and windows; the framework sits closest to ground truth on both countable classes. ByteTrack and BoT-SORT are designed for continuous in-frame multi-object tracking rather than cross-façade element association, and neither incorporates sequence continuity nor scene segmentation; the proposed framework addresses this through VO-driven sequence-break detection. The tracker-comparison and sensitivity procedures are detailed in Appendix C.

Table 8. Comparison of unique element counts produced by ByteTrack, BoT-SORT, and the proposed temporal association and duplicate suppression mechanism before gap correction, evaluated on Site 1.

Metric	Raw YOLO	ByteTrack	BoT-SORT	Framework tracking	Ground truth
Doors	296	12	5	6	7
Windows	184	4	2	3	3
Wall panels	1570	24	14	11	-

* The door undercount (6 of 7) corresponds to door B3; the error originates in the detection stage and is recovered by gap correction.

Table 9 summarises the progressive reduction of raw YOLO detections to unique element counts across all 983 frames of Site 1. The raw detection count of 2,050 reflects the cumulative per-frame output of the detection model before any filtering, where the same physical element is re-detected independently in every frame it appears. Confidence filtering reduced this to 2,035 accepted detections by discarding 15 low-confidence detections below the 30% floor and flagging 24 ambiguous detections for temporal promotion. Temporal association then reduced the accepted pool to 20 unique elements by suppressing 1,002 repeated observations and 39 boundary-clipped detections (clipping is evaluated at the start of the association loop, before tracking); a further door was subsequently recovered by gap correction (Table 9, Stage 5). This 99.02% reduction from raw detections to unique element counts demonstrates that temporal association is the dominant filtering mechanism and that the framework's multi-stage design successfully distinguishes repeated observations from unique physical elements across frames.

Duplicate suppression caught a further 950 detections, 48 doors and 902 wall panels, without changing the final unique element count. These detections passed confidence filtering and appeared new to the temporal tracker, typically after a sequence break, but matched the spatial position of an already-confirmed element. Because the spatial duplicate check runs against the counted objects, any detection it catches was never going to increment the unique count; the element was already recorded. Duplicate suppression therefore acts as a safety net, preventing previously counted elements from being counted again as the camera moves around the building. This behaviour is evident in Table 10, where the 950 detections removed during duplicate suppression correspond to redundant re-observations of previously confirmed objects.

Table 9. Pipeline progression analysis for Site 1, showing the reduction from raw per-frame YOLO detections to unique element observations after confidence filtering, temporal association, and duplicate suppression.

Stage		Doors	Windows	Wall Panels	Count
Stage 1	Raw detections	296	184	1,570	2,050
Stage 2	Confidence filtering	290	184	1,561	2,035
Stage 3	Temporal association	6	3	11	20
Stage 4	Duplicate suppression	6	3	11	20
Stage 5	Gap correction	7	3	11	21
Ground truth	—	7	3	—	—

Each stage of the pipeline addresses a distinct failure mode. Confidence filtering removes unreliable single-frame detections, preventing low-confidence and ambiguous observations from reaching the tracker unchecked. Temporal association is the dominant reduction mechanism, suppressing repeated observations of the same physical elements across frames. Without it, every frame-level re-detection would be counted as a new element, as demonstrated by the raw YOLO and ByteTrack counts in Table 8. VO-based scene segmentation enables façade-level separation. Without it, the tracker has no reset point at building corners, causing elements from different faces to be merged into a single segment, as shown by ByteTrack and BoT-SORT overcounting across all classes. BIM matching converts anonymous detections into assigned element IDs and resolves façade identity without camera registration. Gap correction recovers elements missed by the detector. In Site 1, door B3 was misclassified as a wall panel and would have caused an ordinal assignment error without BIM-guided position recovery.

Table 10 details the detections removed at each stage by class and reason. Doors showed the highest proportion of low-confidence discards relative to their ground truth count, reflecting the visual similarity between doors and wall panels in CFS assemblies and the partial occlusion conditions encountered on Side B. The 15 temporally promoted detections (initially flagged as ambiguous but confirmed after re-sighting) demonstrate that the temporal promotion mechanism successfully recovered detections that single-frame confidence filtering alone would have discarded. The remaining five detections, excluded from the table, were high-confidence observations that passed through all filtering stages without requiring suppression or promotion.

Table 10. Classification of detection outcomes at each filtering stage for Site 1, including low-confidence, ambiguous, repeated, boundary-clipped, and spatial-duplicate detections.

Stage	Reason	Doors	Windows	Wall panels	Total
Confidence filtering	<30% confidence	6	0	9	15
Confidence filtering	Ambiguous	7	2	15	24
Confidence filtering	Ambiguous, promoted after re-sighting	6	2	7	15
Temporal association	Repeated observations (old)	221	179	602	1,002
Temporal association	Clipped at image boundary	8	0	31	39
Duplicate suppression	Spatial signature skip	48	0	902	950

These results demonstrate that the multi-stage pipeline functions as intended across a continuous capture sequence, reducing raw per-frame detections to unique element counts consistent with ground truth.

5.2. Threshold calibration and sensitivity

The framework's thresholds were calibrated using footage from two construction sites and supplementary site imagery. The two calibration sites and supplementary imagery were drawn from the training pool and did not include Site 1, which was reserved entirely for evaluation and sensitivity reporting. The confidence thresholds (0.30–0.70) had the strongest impact on behaviour, governing how 24 ambiguous detections at Site 1 were handled. Detection confidence varies with occlusion, lighting conditions, motion blur, and background clutter; therefore, a single acceptance threshold would either discard recoverable detections or admit unreliable ones. The two-tier design addresses this by using temporal redundancy in video data. Since each object appears across multiple frames, borderline detections can be deferred to subsequent observations, while redundant detections can be safely discarded without affecting final counts. This improves framework performance under occlusion and variable imaging conditions, as demonstrated by the 15 ambiguous detections that were successfully promoted after re-sighting, as shown in Table 10.

Temporal association parameters (IoU = 0.40, centre distance = 0.08) were used to suppress repeated observations, correctly filtering 1,002 duplicate detections. IoU measures the ratio of the overlapping area between two bounding boxes to their combined area; an IoU of 0.40 requires that the overlap constitute at least 40% of the union of the two boxes. The centre-distance criterion provides a fallback when viewpoint shift reduces box overlap: if the normalised frame-centre coordinates of two detections of the same class differ by less than 0.08 in both axes, they are still matched. A tighter threshold can cause an element to be counted more than once; a lower threshold risks merging adjacent elements. A sensitivity analysis was conducted, shown in Table 11. No configuration produced false positives.

Table 11. Sensitivity analysis results for confidence, IoU, and centre-distance threshold perturbations. Precision = 1.000 across all trials and is omitted. IoU: Intersection over Union; ctr_dist: centre-distance fallback; FN: false negatives; F1: harmonic mean of precision and recall.

Trial	High confidence	Low confidence	IoU	Centre_distance	Note	FN	Recall	F1
1	0.7	0.3	0.4	0.08	baseline	1	0.9	0.947
2	0.6	0.3	0.4	0.08	conf_high -0.1	1	0.9	0.947
3	0.8	0.3	0.4	0.08	conf_high +0.1	1	0.9	0.947
4	0.7	0.2	0.4	0.08	conf_low -0.1	2	0.8	0.889
5	0.7	0.4	0.4	0.08	conf_low +0.1	1	0.9	0.947
6	0.7	0.3	0.3	0.08	iou -0.1	1	0.9	0.947
7	0.7	0.3	0.5	0.08	iou +0.1	1	0.9	0.947
8	0.7	0.3	0.4	0.05	ctr_dist -0.03	1	0.9	0.947

9	0.7	0.3	0.4	0.11	ctr_dist +0.03	1	0.9	0.947

The centre-distance fallback threshold was evaluated at three values around the baseline of 0.08, as shown in Table 11. Across all three trials, per-side element counts were identical, producing no change in precision, recall, or F1. The single false negative in all trials corresponds to door B3, misclassified by the detector as a wall panel; this error originates in the detection stage rather than the tracking logic. These results indicate that the centre-distance threshold is not a binding constraint under the capture conditions of Site 1. Under faster panning, stronger camera shake, or significant vertical drift, the fallback would fire more frequently, and the threshold choice would matter more. The nearest-neighbour spacing between adjacent doors on Side B is approximately 0.10 in normalised coordinates, meaning a threshold above this value would theoretically risk merging adjacent detections; the tested upper value of 0.11 produced no merge.

The pixel-difference threshold of 30 drives scene segmentation and is therefore directly responsible for correct face transition detection. Across calibration footage, values during normal lateral motion remained consistently below 20, while corner transitions exceeded 35, providing a stable operating margin around the chosen value. A sensitivity analysis sweeping the threshold across five values (20–40) confirmed this empirically: values of 25 and above produced identical segmentation into four segments (F1 = 0.947), while a value of 20 over-segmented the sequence into nine segments, collapsing BIM side matching (F1 = 0.261). The 30-frame tolerance window defines the interval within which a missed detection can be recovered using predicted appearance frames derived from camera speed and BIM element spacing: it determines whether an existing unmatched detection near the predicted frame is treated as the missing element, and when no detection falls within the window, the element is inserted at the BIM-predicted position. An element is recorded as a true miss only when no appearance prediction is possible. The gap correction window is a design choice tied to capture frame rate; 30 frames correspond to approximately one second at 30 fps, rather than a tunable threshold, and affect only one element across the validation set.

The misclassification of door B3 highlights a limitation of the detector rather than the framework. When door B3 was partially occluded, the detector consistently misclassified it as a low-confidence wall panel, producing only three confirmed door detections where BIM expected four. As described in Section 3.4.2, the gap correction module ran post-hoc after comparing the confirmed door count against BIM: it identified the unmatched BIM door, predicted its appearance from the relative positions of the flanking detections, and, with no unmatched door detection within the 30-frame window around the prediction, inserted B3 with BIM-derived member IDs at the predicted location. The coincident low-confidence wall panel detection at that position is consistent with the detector misclassifying the occluded door, although the correction itself is driven by the BIM count and position constraints rather than by inspection of the panel detection. Door B3 is illustrated in Fig. 19(a). Another case of detector misclassification is shown in Fig. 19(b), where a wall panel was detected as a door and was ignored by the confidence filter; no reclassification was triggered as the detector identified it correctly in the following frames. Gap correction is demonstrated on a single case; its reliability under consecutive misses or irregular camera motion remains to be established.

5.3. Framework scope and positioning

The proposed framework has a different scope from many existing BIM-CV integration approaches. Previous studies such as [46] and [44] have used BIM geometry to guide or validate photogrammetric reconstruction, and generate spatially registered as-built records. Other studies have established spatial correspondence using dedicated sensing systems. For example, Wei et al. [20] and Elrifaee and Zayed [26] achieve spatial correspondence through dedicated hardware, stereo cameras with SLAM and IoT depth sensors, respectively, enabling precise localisation and safety monitoring.

In contrast, the present framework does not aim to produce a metric as-built reconstruction. Its main purpose is to demonstrate that BIM semantic information, including element type, planned quantity, façade location, order, and relative position, can serve as an active constraint for interpreting video-based detections. This allows the framework to reduce repeated counting and support element-level presence verification without relying on full visual completeness, global scene understanding, or camera-to-BIM registration.

5.4. Limitations

The framework was evaluated on a single-storey CFS residential structure under known ground-truth conditions, and the timber bracing present at Site 1 partially occludes structural openings in some frames, reducing detector confidence. These conditions represent realistic early-stage construction but constitute a proof-of-concept validation rather than a controlled test of occlusion robustness. Generalisation across multiple sites, building configurations, camera operators, and construction stages remains to be established.

Two methodological limitations bound what the framework can claim per element class. Wall-panel coverage is assessed as continuity within the observed range rather than against planned BIM quantities, so the absence of detected gaps indicates panel continuity but not count-level validation. The framework targets element-level presence verification — confirming that planned openings exist and assigning detections to BIM IDs — and does not perform dimensional or geometric verification against design intent.

Two structural limitations bound where the matching mechanism remains well-posed. Façades that present identical element-type counts to the matching function cannot be discriminated by element counts alone; this affects geometrically symmetric configurations where opposite faces share the same opening signature, and scene segmentation correctly identifies the transitions but side assignment is ambiguous. Gap correction was demonstrated on a single recovery (door B3); its reliability under consecutive misses or boundary misses, where position interpolation loses its anchors, remains to be established.

5.5. Future work

Broader validation across multiple sites, multi-storey buildings, and varied camera operators is the most immediate priority and would test the BIM matching and façade identification components under conditions outside the current validation envelope. Resolving the symmetric-façade case requires reintroducing information the fingerprint discards, a known starting face, absolute camera orientation, or appearance cues that distinguish otherwise count-equivalent faces, which would extend the operating range to building configurations currently outside the scope. Strengthening gap correction through a richer recovery model that handles consecutive and boundary misses, rather than the single-miss case demonstrated here, would broaden the conditions under which detector failures can be recovered. Looking further out, extending the framework toward geometric-level inspection, verifying dimensional compliance or detecting deviations from design intent, and integration with project scheduling data for automated progress alerting represent the most significant directions for expanding its scope beyond element-level presence verification.

Chapter 6

Conclusion

6.1. Overview

Construction progress monitoring remains heavily dependent on manual inspection, which is labour-intensive, intermittent, and difficult to scale to the frequent checks that residential and light-framed construction would benefit from. Automated alternatives have been pursued for this reason, but they carry their own constraints: sensor- and tag-based systems require dedicated hardware, while vision-based methods, though deployable with ordinary devices, struggle in the visually repetitive and partially completed conditions typical of active sites. A particular difficulty is that appearance alone cannot resolve which building face is being observed or assign a detection to a specific planned element, and the building information models that could supply this context have largely been used as a passive reference, consulted only after detection to score deviation rather than to guide interpretation.

This thesis set out to address that gap by testing whether the semantic content of a BIM model, the type, quantity, order, and relative position of its elements, could instead act as an active constraint on how monocular construction video is interpreted. It proposed a BIM-guided semantic fingerprinting framework for element-level presence verification, in which each building face is represented in advance by a compact semantic record of its planned openings, and accumulated video detections are matched against that record to identify the façade, assign detections to specific BIM elements, and recover elements the detector misses.

The work demonstrates that this is achievable. By matching observed detections against BIM-derived fingerprints, the framework resolved which façade was being observed, assigned detections to planned elements, and recovered a missed opening, all without metric reconstruction, fixed camera infrastructure, or camera-to-model registration. The significance of this result lies less in individual detection accuracy than in what it shows about the role BIM can play: a planned model carries enough structured information to disambiguate visually repetitive scenes that defeat appearance-based methods alone. In doing so, the work shifts BIM from a passive endpoint of the monitoring pipeline to an active participant in interpretation. It shows that this shift is achievable with ordinary site video.

6.2. Achievement of objectives

These outcomes were reached through five objectives, each addressed in turn and summarised in Table 12.

Table 12. Summary of research objectives and their implementation within the proposed framework.

Objective	How it was met	Evidence
O1	Developed a BIM-derived semantic fingerprint representation encoding element type, planned quantity, left-to-right order, and relative position per façade	Section 3.1; Eq. 1–2; Fig. 10
O2	Developed a monocular video interpretation pipeline using object detection, visual odometry, confidence filtering, and temporal consistency reasoning	Sections 3.2–3.3; Tables 2–3; Figs 5–9
O3	Developed a BIM-guided matching and assignment mechanism for façade identification, element assignment, and gap correction	Section 3.4; Fig. 11
O4	Implemented and evaluated the complete framework on a single-storey CFS structure	Section 3.4.3; Chapter 4; Tables 4–7; Figs 13–14
O5	Evaluated robustness and failure conditions through sensitivity and theoretical analysis	Chapter 5; Table 11; Sections 5.2, 5.4

Objective 1 (formulate a BIM-derived semantic fingerprint representation) was met by the BIM extraction and fingerprinting workflow of Section 3.1, which encodes each façade as a structured record of element type, planned quantity, spatial order, and relative position.

Objective 2 (design a monocular video interpretation pipeline) was met by the detection, confidence-filtering, visual-odometry, and temporal-association components of Sections 3.2–3.3, which convert raw per-frame detections into stable façade-level observations.

Objective 3 (establish a BIM-guided matching and assignment mechanism) was met by the fingerprint matching, element assignment, and gap-correction procedures of Section 3.4, which identify each façade and assign confirmed detections to specific BIM element IDs without metric registration.

Objective 4 (implement and evaluate the framework on a single-storey CFS structure) was met in Chapter 4: the 983-frame Site 1 sequence was reduced from 2,050 raw detections to 20 unique element observations, the four exterior faces were correctly segmented and matched, and door and window counts matched ground truth after BIM-guided recovery of one misclassified door.

Objective 5 (investigate robustness and failure conditions) was addressed in Chapter 5 through threshold sensitivity analysis (Table 11) and analysis of the matching function's well-posedness, which characterised the symmetric-façade configuration as the boundary at which side assignment becomes ambiguous.

6.3. Contributions

The study demonstrates three main technical contributions: a visual-odometry-assisted temporal association method for duplicate suppression without metric localisation; a semantic fingerprinting strategy for façade identification and BIM element assignment; and a BIM-guided gap correction mechanism, demonstrated on a representative single-miss case, for recovering elements missed by the detector. The workflow was implemented through a pyRevit extension that connects BIM extraction, element assignment, and in-model progress visualisation without specialist software or hardware, providing a low-cost, deployable route to element-level presence verification for residential construction.

6.4. Limitations and future work

The framework's limitations and the directions opening from them are stated in Chapter 5. The evaluation was conducted on a single residential case study under controlled conditions, supporting an assessment of mechanism correctness rather than statistical generalisation across diverse project settings. Within this context, additional limitations include the count-level (rather than dimension-level) claim of the verification output, the matching ambiguity under symmetric façades, and the demonstration of gap correction on a single recovery. The most immediate future work is broader validation across configurations, building heights, and operators; resolution of the symmetric-façade case through additional information channels; and extension of the verification scope toward geometric-level inspection and integration with project schedules. The pyRevit extension and supporting code developed in this study are available at: https://github.com/MuhammadAhmad-442.

Appendix A

Algorithm 1: Side Fingerprint Matching

Input: *BIM export (per-side element lists); detected count vector for a segment.*
Output: *best-matching BIM side, confidence, match type.*
Constant: *MATCH_ELEMENTS = {door, window}   // wall panels excluded*

```
procedure BUILD_FINGERPRINTS(bim_export)
  for each side S in bim_export.exterior.sides do
    fingerprint[S] <- {door: 0, window: 0, wall_panels: 0}
    for each element e in S.elements do
      fingerprint[S][e.type] <- fingerprint[S][e.type] + 1
    end for
  end for
  return fingerprint
end procedure
```

```
procedure SCORE(detected, bim_counts)
  overlap <- 0; excess <- 0; missing <- 0
  for each class c in MATCH_ELEMENTS do
    overlap <- overlap + min(detected[c], bim_counts[c])
    excess  <- excess  + max(detected[c] - bim_counts[c], 0)
    missing <- missing + max(bim_counts[c] - detected[c], 0)
  end for
  total_bim  <- sum(bim_counts[c] for c in MATCH_ELEMENTS)
  confidence <- overlap / max(total_bim, 1)
  if excess = 0 and missing = 0 then
    match_type <- "exact"
  else if excess = 0 and missing > 0 then
    match_type <- "subset"
  else
    match_type <- "fuzzy"
  end if
  return (overlap, excess, missing, confidence, match_type)
end procedure
```

```
procedure MATCH(detected, claimed_sides)
  candidates <- empty list
  for each side S not in claimed_sides do
    candidates.append( (S, SCORE(detected, fingerprint[S])) )
  end for
  sort candidates by overlap descending, then excess ascending, then missing ascending
  if candidates is not empty then
    return candidates[0]            // best match
  else
    return NO_MATCH
  end if
end procedure
```

Algorithm 2: BIM-constrained door gap filling

Input: *BIM side data (elements with position, member_ids; face width_mm); confirmed detection log for the side; camera direction.*
Output: *element list with detected and predicted entries.*
Constant: *WINDOW_FRAMES = 30   // search radius around a predicted frame*

```
procedure FILL_GAPS(bim_side, detections, direction)
  bim_doors <- BIM elements of type door, sorted by position ascending
  confirmed <- detections with outcome = "c" and sent_to_bim = "YES", sorted by frame
  n_bim <- |bim_doors|;  n_det <- |confirmed|
  if n_det = n_bim then
    return confirmed                  // counts match, nothing to fill
  end if
  if n_det = 0 then
    return empty list                 // cannot estimate speed
  end if
  span_mm <- (bim_doors.last.position - bim_doors.first.position) x width_mm
  speed   <- span_mm / (confirmed.last.frame - confirmed.first.frame)
  det_pos <- NORMALISE_CX(confirmed, direction)
  used_bim <- empty set
  for each detection i in det_pos do
    j* <- argmin over j not in used_bim of |det_pos[i] - bim_doors[j].position|
    used_bim <- used_bim + {j*}
    record pairing (i -> j*)
  end for
  predicted <- empty list
  for each unmatched BIM door b (b not in used_bim) do
    (pred_frame, pred_cx) <- PREDICT(b, speed, confirmed, det_pos)
    if an unmatched detection lies within WINDOW_FRAMES of pred_frame then
      continue                          // real detection already covers it
    else
      predicted.append({ frame: pred_frame, cx: pred_cx, tag: b.tag,
                         member_ids: b.member_ids, source: "predicted" })
    end if
  end for
  return sort(confirmed + predicted by frame)
end procedure

procedure NORMALISE_CX(confirmed, direction)
  cx <- [ d.cx for d in confirmed ]
  if direction = "RL" then
    cx <- [ max(cx) - v for v in cx ]      // flip for right-to-left travel
  end if
  return [ (v - min(cx)) / max(max(cx) - min(cx), 1) for v in cx ]
end procedure

procedure PREDICT(bim_door, speed, confirmed, det_pos)
  before <- detections with position <= bim_door.position
  after  <- detections with position >  bim_door.position
  if before is not empty and after is not empty then     // interpolate between neighbours
```

```
    t        <-  (bim_door.position  -  before.last.pos)  /  (after.first.pos  -
    before.last.pos)
    frame <- before.last.frame + t x (after.first.frame - before.last.frame)
    cx    <- before.last.cx    + t x (after.first.cx    - before.last.cx)
  else if before is not empty then                        // extrapolate forward
    frame <- before.last.frame + (bim_door.position - before.last.pos) x width_mm
    / speed
    cx    <- before.last.cx
  else                                                   // extrapolate backward
    frame <- after.first.frame - (after.first.pos - bim_door.position) x width_mm
    / speed
    cx    <- after.first.cx
  end if
  return (frame, cx)
end procedure
```

Algorithm 3: Wall panel coverage reporting

Input: *confirmed wall-panel detections; face width_mm; BIM elements (landmark labels only); camera direction.*
Output: *coverage percentage, merged spans, uncovered gaps.*
Constants: *MIN_GAP_FRACTION = 0.05 (gaps below this ignored as noise); MERGE_THRESHOLD = 0.02 (spans this close are the same panel)*

```
procedure BUILD_REPORT(detections, width_mm, bim_elements, direction)
  panels <- empty list
  for each detection d in detections with outcome = "c" and sent_to_bim = "YES"
  do
    if direction = "RL" then
      face_left  <- 1 - (d.x2 / frame_width)
      face_right <- 1 - (d.x1 / frame_width)
    else
      face_left  <- d.x1 / frame_width
      face_right <- d.x2 / frame_width
    end if
    clamp face_left, face_right to [0, 1]
    panels.append({face_left, face_right, frame: d.frame})
  end for
  if panels is empty then
    return zero-coverage report
  end if
  sort panels by face_left ascending
  spans <- empty list
  for each panel p in panels do
    if spans is empty or p.face_left > spans.last.right + MERGE_THRESHOLD then
      spans.append( new span [p.face_left, p.face_right] )
    else
      spans.last.right <- max(spans.last.right, p.face_right)
    end if
  end for
  detected_range <- spans.last.right - spans.first.left
```

```
  covered        <- sum(span.right - span.left for span in spans)
  coverage_pct   <- min(covered / detected_range x 100, 100)
  gaps <- empty list
  for i from 1 to |spans| - 1 do
    gap <- spans[i].left - spans[i-1].right
    if gap > MIN_GAP_FRACTION then
      gaps.append({ size_fraction: gap, size_mm: gap x width_mm,
                    nearest_bim: BIM element closest to gap centre })
    end if
  end for
  return { coverage_pct, spans, gaps }
end procedure
```

Appendix B

Figure B.1: FrameCAD Structural Plan and Blender 3D Model Render.

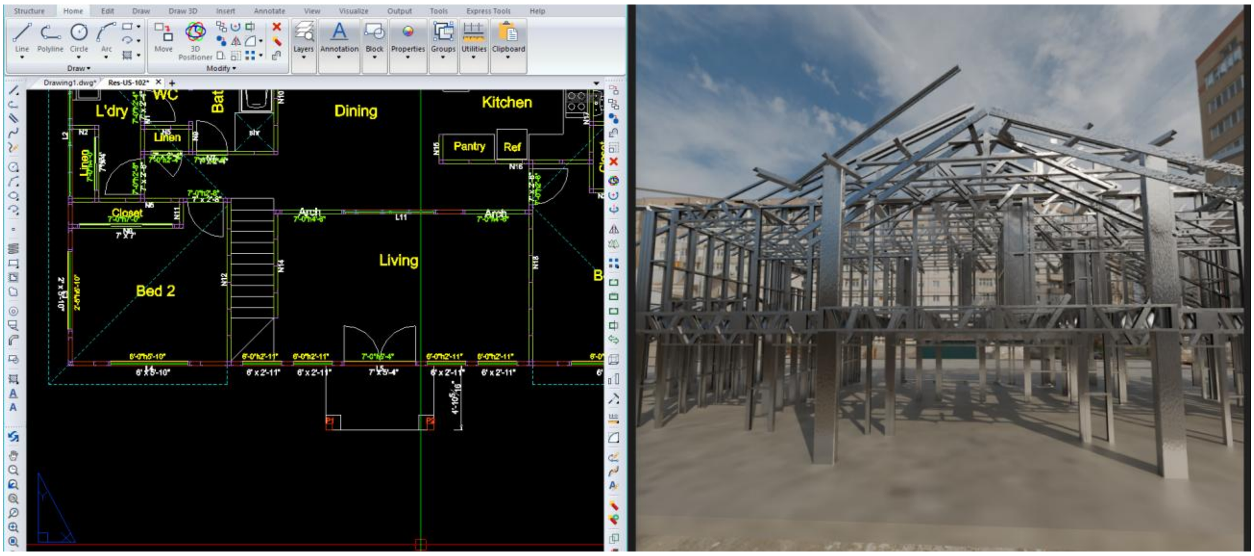


Figure B.2: FrameCAD 3D model visualisation and corresponding annotated image in Roboflow.

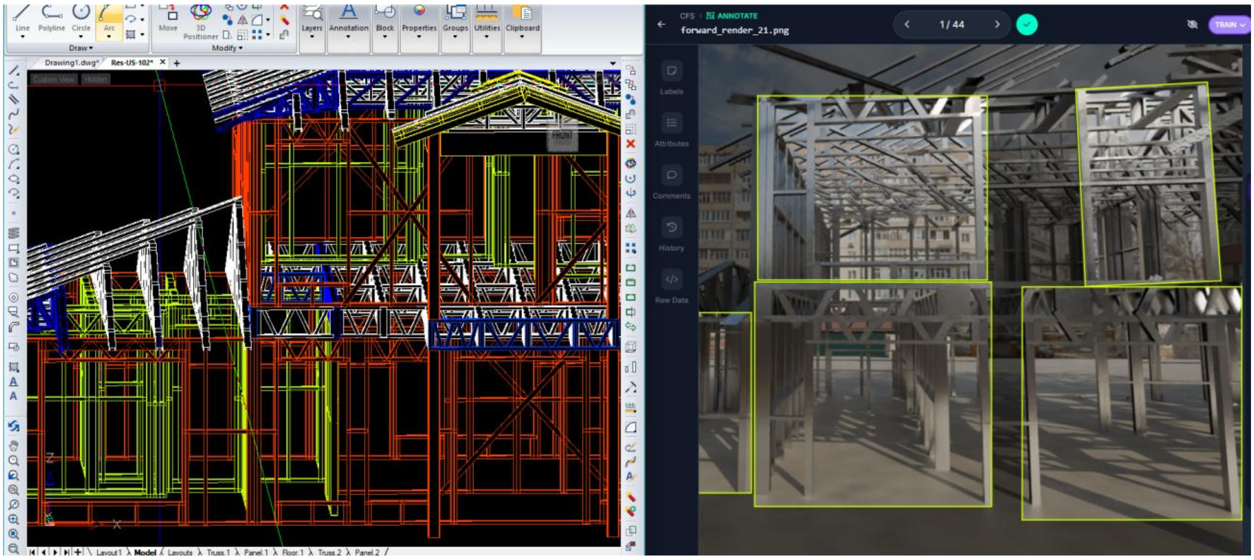


Figure B.3: Automated rendering workflow in Blender.

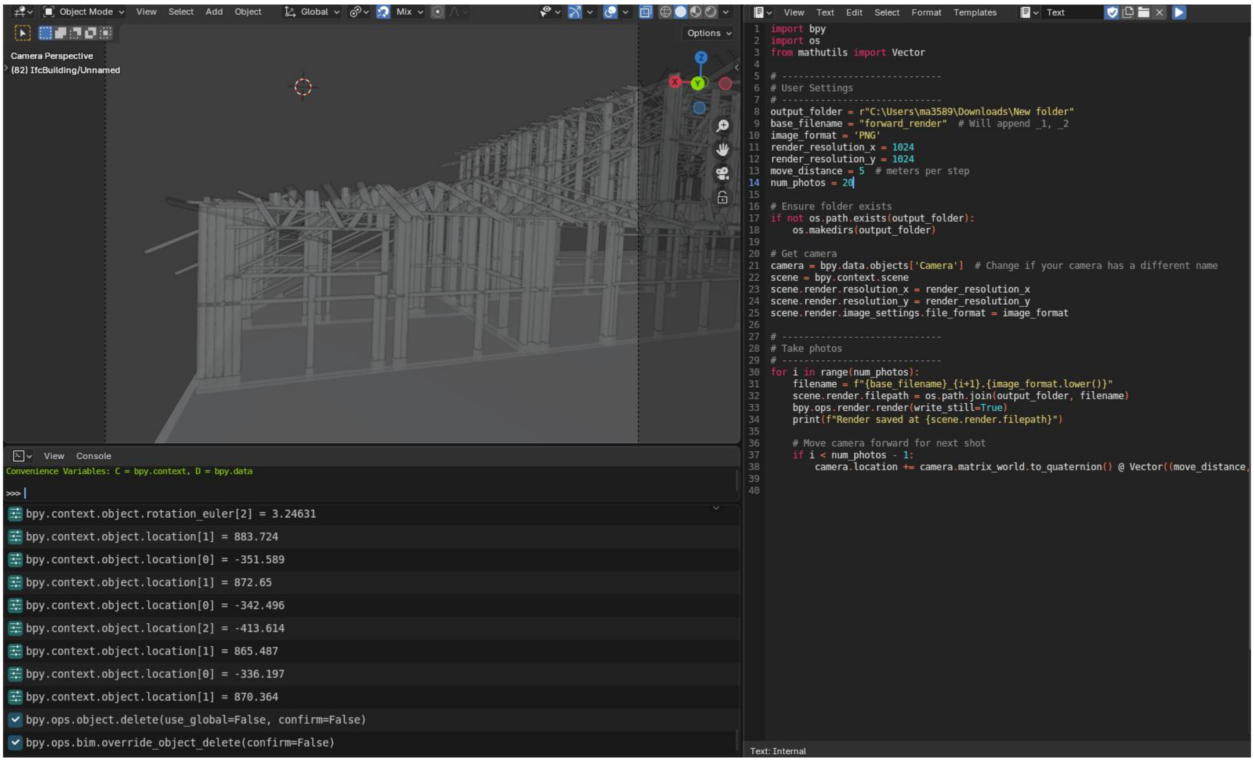
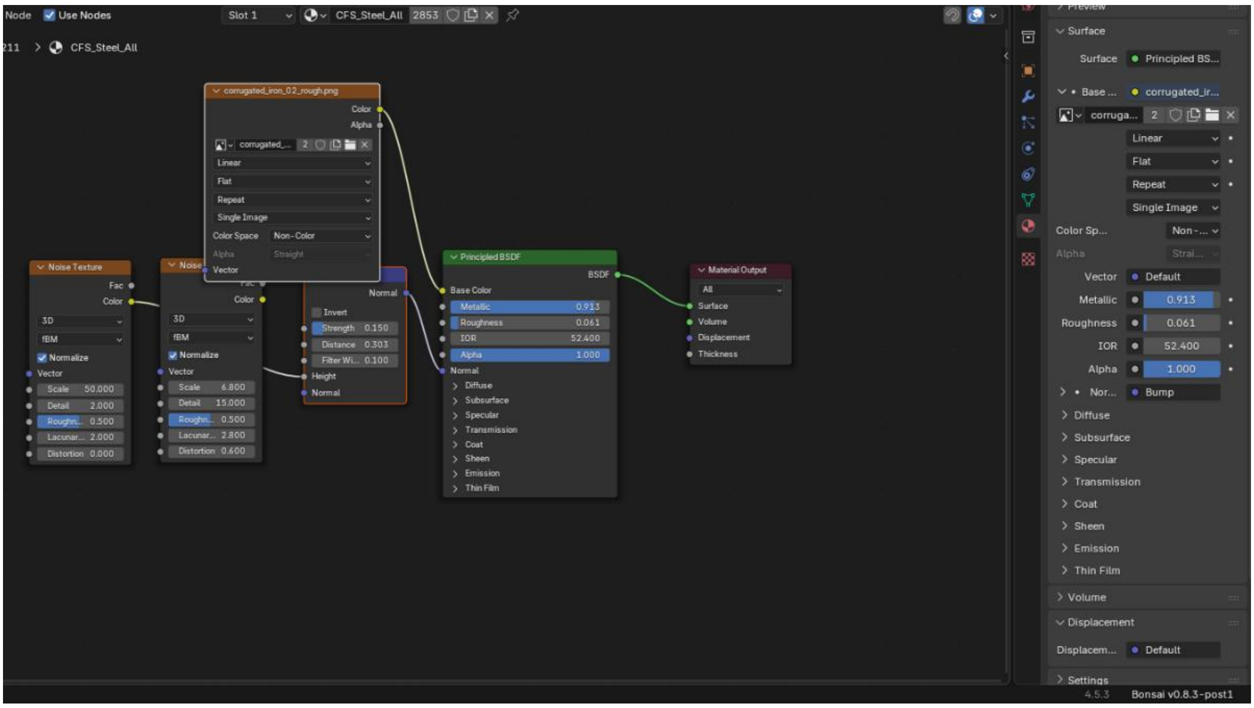


Figure B.4: Selected steel colour profile applied in Blender for synthetic CFS rendering.



Appendix C

Algorithm 1: Tracker comparison

Purpose: *count unique element instances using built-in multi-object trackers, to benchmark against the proposed framework.*
Input: *video frames; YOLO model; tracker config (bytetrack.yaml or botsort.yaml).*
Output: *unique track-ID count per element class.*

```
procedure RUN_BUILTIN_TRACKER(frames, model, tracker_name)
  track_ids <- { door: empty set, window: empty set, wall_panels: empty set }
  reset model tracker state          // model.predictor <- None
  for each frame f in frames do
    result <- model.track(f, tracker = tracker_name, persist = True)
    if result has no track IDs then
      continue
    end if
    for each (box, id) in result do
      c <- normalise_class(box.class)
      if c is a target type then
        track_ids[c].add(id)
      end if
    end for
  end for
  return { class: |track_ids[class]| for each class }
end procedure
```

```
procedure COMPARE_ALL(frames, model)
  proposed  <- run proposed framework (segment, match, gap-fill)
  bytetrack <- RUN_BUILTIN_TRACKER(frames, model, "bytetrack.yaml")
  reset model tracker state
  botsort   <- RUN_BUILTIN_TRACKER(frames, model, "botsort.yaml")
  for each method in {proposed, bytetrack, botsort} do
    for each class:  count <- unique instances
    compare count against ground truth -> TP, FP, FN, precision, recall, F1
    record capability flags:
      facade_assigned, bim_ids_assigned,
        missed_element_recovery, cross_sequence_continuity
  end for
  return comparison table
end procedure
```

Algorithm 2: Sensitivity analysis

Purpose: *quantify how robust the framework is to its key thresholds.*
Input: *frames; cached YOLO detections (inference run once, never repeated); ground-truth element counts per side.*
Output: *per-trial precision, recall, F1; stability verdict.*

```
procedure SENSITIVITY_SWEEP(grid, parameter_set)
```

```
  baseline <- current threshold values
  trials   <- empty list
  for each configuration in grid do
    patch the module-level threshold(s) to this configuration
    reset any stateful component if required     // e.g. VO state for pixel-diff
    result <- RUN_FRAMEWORK(frames, cached_detections)     // no YOLO re-inference
    (TP, FP, FN) <- SCORE(result, ground_truth)
    precision <- TP / (TP + FP)
    recall    <- TP / (TP + FN)
    F1        <- 2 x precision x recall / (precision + recall)
    record trial { config, TP, FP, FN, precision, recall, F1 }
  end for
  restore baseline thresholds
  base_F1   <- F1 of the baseline trial
  max_delta <- max over trials of |F1 - base_F1|
  stable    <- (max_delta < 0.05)        // < 5 percentage-point swing
  return trials, stable
end procedure
```

Three sweeps are run, one parameter family at a time:

Sweep 1 — confidence & IoU (one parameter shifted per trial):

```
  CONF_HIGH  in  {0.60, 0.70, 0.80}
  CONF_LOW   in  {0.20, 0.30, 0.40}
  IOU        in  {0.30, 0.40, 0.50}
  primary outcome: precision / recall / F1
```

Sweep 2 — scene-cut threshold:

```
  PIXEL_DIFF_BREAK  in  {20, 25, 30, 35, 40}
  reset VO state between trials
  primary outcome: number of SEQUENCE_BREAKs -> segment count, then F1
```

Sweep 3 — centre-distance threshold:

```
  CENTER_THRESHOLD  in  {0.05, 0.08, 0.11}
  primary outcome: F1, with worst trial saved as annotated video
    (tight -> re-counts on shake; loose -> merges neighbours)
```

```
procedure SCORE(result, ground_truth)
  for each side do
    for each class do
      TP <- TP + min(detected, ground_truth)
      FP <- FP + max(detected - ground_truth, 0)
      FN <- FN + max(ground_truth - detected, 0)
    end for
  end for
  aggregate across sides
end procedure
```

References

[1] Omar H, Mahdjoubi L, Kheder G. Towards an automated photogrammetry-based approach for monitoring and controlling construction site activities. Computers in Industry 2018;98:172–82. https://doi.org/10.1016/j.compind.2018.03.012.

[2] Shah FH, Bhatti OS, Ahmed S. A review of the effects of project management practices on cost overrun in construction projects. Engineering Proceedings 2023;44:1. https://doi.org/10.3390/engproc2023044001.

[3] Gómez-Cabrera A, Gutierrez-Bucheli L, Muñoz S. Causes of time and cost overruns in construction projects: a scoping review. International Journal of Construction Management 2024;24:1107–25. https://doi.org/10.1080/15623599.2023.2252288.

[4] Sriram Changali, Azam Mohammad, Mark van Nieuwland. The construction productivity imperative 2015. https://www.mckinsey.com/capabilities/operations/our-insights/the-construction-productivity-imperative (accessed October 13, 2025).

[5] Latif M, Saleem M, Cheema SM. Analyzing the key factors contributing to project delays in the construction industry: a comprehensive study. Journal of Development and Social Sciences 2023;4:936–44. https://doi.org/10.47205/jdss.2023(4-III)85.

[6] Abdelalim AM, Salem M, Salem M, Al-Adwani M, Tantawy M. An Analysis of Factors Contributing to Cost Overruns in the Global Construction Industry. Buildings 2025;15:18. https://doi.org/10.3390/buildings15010018.

[7] Navon R, Sacks R. Assessing research issues in Automated Project Performance Control (APPC). Automation in Construction 2007;16:474–84. https://doi.org/10.1016/j.autcon.2006.08.001.

[8] Housing in Aotearoa New Zealand: 2025. Stats NZ 2025. https://www.stats.govt.nz/reports/housing-in-aotearoa-new-zealand-2025/ (accessed May 11, 2026).

[9] Valero E, Adán A. Integration of RFID with other technologies in construction. Measurement 2016;94:614–20. https://doi.org/10.1016/j.measurement.2016.08.037.

[10] Ekanayake B, Wong JK-W, Fini AAF, Smith P. Computer vision-based interior construction progress monitoring: A literature review and future research directions. Automation in Construction 2021;127:103705. https://doi.org/10.1016/j.autcon.2021.103705.

[11] Gong J, Caldas CH. Computer Vision-Based Video Interpretation Model for Automated Productivity Analysis of Construction Operations. Journal of Computing in Civil Engineering 2010;24:252–63. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000027.

[12] Xu S, Wang J, Shou W, Ngo T, Sadick A-M, Wang X. Computer vision techniques in construction: a critical review. Arch Computat Methods Eng 2021;28:3383–97. https://doi.org/10.1007/s11831-020-09504-3.

[13] Begić H, Galić M. A Systematic Review of Construction 4.0 in the Context of the BIM 4.0 Premise. Buildings 2021;11:337. https://doi.org/10.3390/buildings11080337.

[14] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017;39:1137–49. https://doi.org/10.1109/TPAMI.2016.2577031.

[15] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, p. 779–88. https://doi.org/10.1109/CVPR.2016.91.

[16] He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, p. 2980–8. https://doi.org/10.1109/ICCV.2017.322.

[17] ElQasaby AR, Alqahtani FK, Alheyf M. State of the Art of BIM Integration with Sensing Technologies in Construction Progress Monitoring. Sensors 2022;22:3497. https://doi.org/10.3390/s22093497.

[18] Rehman IU, Mazher KM, Wuni IY. Systematic review of 4D BIM benefits in construction projects. Results in Engineering 2025;28:107091. https://doi.org/10.1016/j.rineng.2025.107091.

[19] Sami Ur Rehman M, Shafiq MT, Ullah F. Automated Computer Vision-Based Construction Progress Monitoring: A Systematic Review. Buildings 2022;12:1037. https://doi.org/10.3390/buildings12071037.

[20] Wei W, Lu Y, Zhong T, Li P, Liu B. Integrated vision-based automated progress monitoring of indoor construction using mask region-based convolutional neural networks and BIM. Automation in Construction 2022;140:104327. https://doi.org/10.1016/j.autcon.2022.104327.

[21] Shamsollahi D, Moselhi O, Khorasani K. Data integration using deep learning and real-time locating system (RTLS) for automated construction progress monitoring and reporting. Automation in Construction 2024;168:105778. https://doi.org/10.1016/j.autcon.2024.105778.

[22] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple Online and Realtime Tracking. 2016 IEEE International Conference on Image Processing (ICIP), 2016, p. 3464–8. https://doi.org/10.1109/ICIP.2016.7533003.

[23] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China: IEEE Press; 2017, p. 3645–9. https://doi.org/10.1109/ICIP.2017.8296962.

[24] Zhang Y, Sun P, Jiang Y, Yu D, Weng F, Yuan Z, et al. ByteTrack: Multi-object Tracking by Associating Every Detection Box. Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII, Berlin, Heidelberg: Springer-Verlag; 2022, p. 1–21. https://doi.org/10.1007/978-3-031-20047-2_1.

[25] Aharon N, Orfaig R, Bobrovsky B-Z. BoT-SORT: Robust Associations Multi-Pedestrian Tracking 2022. https://doi.org/10.48550/arXiv.2206.14651.

[26] Elrifaee M, Zayed T. Smart IoT-BIM framework with modified zonal safety analysis (ZSA) for real-time safety monitoring in construction. Automation in Construction 2025;178:106431. https://doi.org/10.1016/j.autcon.2025.106431.

[27] Paneru S, Jeelani I. Computer vision applications in construction: Current state, opportunities & challenges. Automation in Construction 2021;132:103940. https://doi.org/10.1016/j.autcon.2021.103940.

[28] Succar B. Building information modelling framework: A research and delivery foundation for industry stakeholders. Automation in Construction 2009;18:357–75. https://doi.org/10.1016/j.autcon.2008.10.003.

[29] Soltanmohammadlou N, Sadeghi S, Hon CKH, Mokhtarpour-Khanghah F. Real-time locating systems and safety in construction sites: A literature review. Safety Science 2019;117:229–42. https://doi.org/10.1016/j.ssci.2019.04.025.

[30] Valero E, Adán A, Cerrada C. Evolution of RFID Applications in Construction: A Literature Review. Sensors 2015;15:15988–6008. https://doi.org/10.3390/s150715988.

[31] Heuer C, Jung V, Brell-Cokcan S. A hardware-based RFID identification and tracking system for components in digitalised construction logistics. Developments in the Built Environment 2025;23:100726. https://doi.org/10.1016/j.dibe.2025.100726.

[32] Bosché F, Ahmed M, Turkan Y, Haas CT, Haas R. The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. Automation in Construction 2015;49:201–13. https://doi.org/10.1016/j.autcon.2014.05.014.

[33] Aryan A, Bosché F, Tang P. Planning for terrestrial laser scanning in construction: A review. Automation in Construction 2021;125:103551. https://doi.org/10.1016/j.autcon.2021.103551.

[34] Zhang C, Arditi D. Automated progress control using laser scanning technology. Automation in Construction 2013;36:108–16. https://doi.org/10.1016/j.autcon.2013.08.012.

[35] Valero E, Bosché F, Bueno M. Laser scanning for BIM. ITcon 2022;27:486–95. https://doi.org/10.36680/j.itcon.2022.023.

[36] Turkan Y, Bosche F, Haas CT, Haas R. Automated progress tracking using 4D schedule and 3D sensing technologies. Automation in Construction 2012;22:414–21. https://doi.org/10.1016/j.autcon.2011.10.003.

[37] Wu X, Sahoo D, Hoi SCH. Recent advances in deep learning for object detection. Neurocomputing 2020;396:39–64. https://doi.org/10.1016/j.neucom.2020.01.085.

[38] Jocher G, Qiu J, Chaurasia A. Ultralytics YOLO 2023.

[39] Hamledari H, McCabe B, Davari S. Automated computer vision-based detection of components of under-construction indoor partitions. Automation in Construction 2017;74:78–94. https://doi.org/10.1016/j.autcon.2016.11.009.

[40] Chua WP, Cheah CC. Deep-Learning-Based Automated Building Construction Progress Monitoring for Prefabricated Prefinished Volumetric Construction. Sensors 2024;24:7074. https://doi.org/10.3390/s24217074.
[41] Aqel MOA, Marhaban MH, Saripan MI, Ismail NBt. Review of visual odometry: types, approaches, challenges, and applications. SpringerPlus 2016;5:1897. https://doi.org/10.1186/s40064-016-3573-7.
[42] Yang N, Wang R, Gao X, Cremers D. Challenges in monocular visual odometry: photometric calibration, motion bias, and rolling shutter effect. IEEE Robot Autom Lett 2018;3:2878–85. https://doi.org/10.1109/LRA.2018.2846813.
[43] Lu T, Tervola S, Lü X, Kibert CJ, Zhang Q, Li T, et al. A novel methodology for the path alignment of visual SLAM in indoor construction inspection. Automation in Construction 2021;127:103723. https://doi.org/10.1016/j.autcon.2021.103723.
[44] Golparvar-Fard M, Peña-Mora F, Savarese S. Automated progress monitoring using unordered daily construction photographs and IFC-based building information models. Journal of Computing in Civil Engineering 2015;29:04014025. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000205.
[45] Braun A, Tuttas S, Borrmann A, Stilla U. A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds. Journal of Information Technology in Construction (ITcon) 2015;20:68–79.
[46] Braun A, Tuttas S, Borrmann A, Stilla U. Improving progress monitoring by fusing point clouds, semantic data and computer vision. Automation in Construction 2020;116:103210. https://doi.org/10.1016/j.autcon.2020.103210.
[47] Kulinan AS, Park M, Aung PPW, Cha G, Park S. Advancing construction site workforce safety monitoring through BIM and computer vision integration. Automation in Construction 2024;158:105227. https://doi.org/10.1016/j.autcon.2023.105227.
[48] Hsieh C-C, Chen H-M, Chen W-Y, Wu T-Y. Automated construction progress monitoring and control through AI-based image recognition and BIM integration. Automation in Construction 2026;183:106783. https://doi.org/10.1016/j.autcon.2026.106783.
[49] Pal A, Lin JJ, Hsieh S-H, Golparvar-Fard M. Activity-level construction progress monitoring through semantic segmentation of 3D-informed orthographic images. Automation in Construction 2024;157:105157. https://doi.org/10.1016/j.autcon.2023.105157.
[50] Iran-Nejad, Ehsan. pyRevit: Rapid Application Development (RAD) Environment for Autodesk Revit n.d.
[51] Adan A, Huber D. 3D Reconstruction of Interior Wall Surfaces under Occlusion and Clutter. Visualization and Transmission 2011 International Conference on 3D Imaging, Modeling, Processing, 2011, p. 275–81. https://doi.org/10.1109/3DIMPVT.2011.42.
[52] Usamentiaga R, Lema DG, Pedrayes OD, Garcia DF. Automated Surface Defect Detection in Metals: A Comparative Review of Object Detection and Semantic Segmentation Using Deep Learning. IEEE Transactions on Industry Applications 2022;58:4203–13. https://doi.org/10.1109/TIA.2022.3151560.
[53] Tran SV-T, Pham HC, Le QT, Lee U-K. Digital technologies for monitoring hazardous area entry in construction sites. Automation in Construction 2025;177:106357. https://doi.org/10.1016/j.autcon.2025.106357.
[54] Bonsai: OpenBIM Blender Add-on n.d. https://bonsaibim.org/ (accessed December 30, 2025).
[55] Hong Y, Park S, Kim H, Kim H. Synthetic data generation using building information models. Automation in Construction 2021;130:103871. https://doi.org/10.1016/j.autcon.2021.103871.
[56] Han S, Park W, Jeong K, Hong T, Koo C. Utilizing synthetic images to enhance the automated recognition of small-sized construction tools. Automation in Construction 2024;163:105415. https://doi.org/10.1016/j.autcon.2024.105415.
[57] Kim J, Wang I, Yu J. Experimental Study on Using Synthetic Images as a Portion of Training Dataset for Object Recognition in Construction Site. Buildings 2024;14:1454. https://doi.org/10.3390/buildings14051454.
[58] Zhang Z. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000;22:1330–4. https://doi.org/10.1109/34.888718.