

# BIM Informed Visual SLAM for Construction Monitoring

Asier Bikandi-Noya<sup>1</sup>, Miguel Fernandez-Cortizas<sup>1</sup>, Muhammad Shaheer<sup>1</sup>,  
Ali Tourani<sup>1</sup>, Holger Voos<sup>1</sup>, and Jose Luis Sanchez-Lopez<sup>1</sup>

**Abstract**—Simultaneous Localization and Mapping (SLAM) is a key tool for monitoring construction sites, where aligning the evolving *as-built* state with the *as-planned* design enables early error detection and reduces costly rework. LiDAR-based SLAM achieves high geometric precision, but its sensors are typically large and power-demanding, limiting their use on portable platforms. Visual SLAM offers a practical alternative with lightweight cameras already embedded in most mobile devices. However, visually mapping construction environments remains challenging: repetitive layouts, occlusions, and incomplete or low-texture structures often cause drift in the trajectory map. To mitigate this, we propose an RGB-D SLAM system that incorporates the Building Information Model (BIM) as structural prior knowledge. Instead of relying solely on visual cues, our system continuously establishes correspondences between detected wall and their BIM counterparts, which are then introduced as constraints in the back-end optimization. The proposed method operates in real time and has been validated on real construction sites, reducing trajectory error by an average of 23.71% and map RMSE by 7.14% compared to visual SLAM baselines. These results demonstrate that BIM constraints enable reliable alignment of the digital plan with the *as-built* scene, even under partially constructed conditions.

## I. INTRODUCTION

Monitoring construction progress requires comparing the *as built* state of a site map with the *as planned* design map [1], allowing visualization and continuous tracking even when structures are incomplete or partially built.

To enable this comparison, the two maps must be aligned in real time to support practical applications such as Augmented Reality (AR) monitoring on portable devices [2], making online processing a key requirement for reliable progress monitoring in different construction phases. A common approach to obtain the *as-built* state is by mapping the environment using Simultaneous Localization and Mapping (SLAM) [3] techniques. In construction environments, LiDAR-based SLAM is often preferred for its high geometric precision [4], providing accurate 3D reconstructions of the

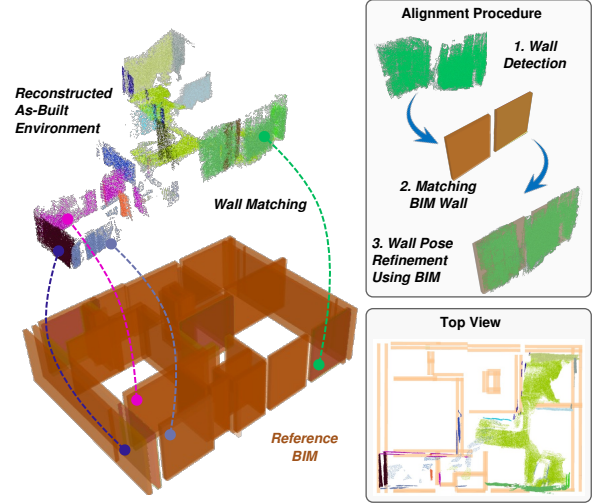


Fig. 1: Visualization of the proposed BIM-informed RGB-D SLAM system. The figure shows the *as-planned* BIM and the reconstructed *as-built* map, with detected walls matched to their BIM counterparts. A 2D projection illustrates the correct alignment between the SLAM map and the BIM layout.

built environment. However, LiDAR sensors are large, expensive, and require high power, making them unsuitable for portable construction monitoring applications. Visual SLAM offers a compelling alternative by using only camera sensors, which are lightweight, low-cost, and already integrated into most mobile devices. Moreover, cameras provide rich visual information, offering the potential to facilitate the understanding of structural elements.

However, visual SLAM in construction environments faces significant challenges, such as occlusions from equipment and materials, repetitive structural layouts, and low-texture surfaces [5], which can often cause tracking drift and mapping errors. As these errors accumulate over time, the resulting maps lose accuracy and become unreliable for construction monitoring [6]. To address these limitations, Building Information Models (BIM) offer valuable geometric priors that can constrain and guide the SLAM process [7]. BIM provides the complete *as-planned* architectural layout, including precise locations and dimensions of structural elements such as walls, columns, and floors. Incorporating these constraints into SLAM systems can help to maintain better tracking accuracy and reduce drift, leading to more reliable long-term construction monitoring.

<sup>1</sup>Authors are with the Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg. Holger Voos is also associated with the Faculty of Science, Technology, and Medicine, University of Luxembourg, Luxembourg. {asier.bikandi, muhammad.shaheer, miguel.fernandez, ali.tourani, holger.voos, joseluis.sanchezlopez}@uni.lu

\* This work was partially funded by the Fonds National de la Recherche of Luxembourg (FNR) under the project C22/IS/17387634/DEUS, by a partnership between the SnT-UL and Gamma Tech.

\* We would like to thank CLK S.Ä.R.L for granting us access to test this work in their construction sites.

\* For the purpose of Open Access, and in fulfillment of the obligations arising from the grant agreement, the authors have applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

Most existing approaches have leveraged BIM either for global localization [8], [9] or with LiDAR-based SLAM [10]. While BIM integration with LiDAR systems is well-explored, combining BIM with visual approaches remains largely underexplored. Some works integrate BIM priors with visual global localization [11], but usually require extensive coverage of the environment for reliable alignment. However, in online construction monitoring, alignment is desirable at the beginning of monitoring [12].

To address these challenges, as illustrated in Fig. 1, this work proposes a BIM-informed RGB-D SLAM system that operates online to enhance map consistency by minimizing trajectory drift. Our approach extends a visual SLAM backbone by incorporating back-end constraints derived from wall associations between the *as-built* and *as-planned* maps. Walls are chosen as the key structural elements because they are common, stable, and informative in construction environments, providing reliable anchors for aligning the SLAM map with the BIM model.

The main contributions of this work are:

- A BIM-informed visual SLAM pipeline that reduces drift by enforcing structural consistency between the *as-built* map and the *as-planned* BIM.
- An initialization and association strategy based on walls that requires minimal prior information and allows monitoring from the earliest stages of monitoring.
- An online system that operates robustly in different real-world construction environments, maintaining accurate mapping even under partially built conditions.

## II. RELATED WORKS

**Visual SLAM in structured environments.** In construction scenarios, visual SLAM must remain robust under long-term operation, partial observations, and repetitive layouts. *ORB-SLAM3* [13] is a strong candidate as it supports multi-session mapping and re-localization through the ATLAS manager, while *vS-Graphs* [14] further extends this line by modeling structural planes and integrating them into a graph-based back-end, making it particularly relevant for construction environments. An alternative approach is *BAD-SLAM* [15], which introduced a direct bundle adjustment formulation for dense RGB-D data, jointly optimizing poses and structures; however, real-time performance over long sequences is not guaranteed, as the system lacks dedicated optimization strategies to maintain efficiency over extended trajectories. Other advances have focused on efficiency and robustness of the front-end, such as GPU-accelerated visual odometry (*cuVSLAM*) [16] and self-supervised feature detection (*SuperPoint*) [17].

However, in construction environments, large textureless walls, repetitive layouts, and partial or evolving structures exacerbate the challenges of pose estimation. Such conditions increase drift and cause ambiguities in registration, leading to deviated maps that limit their use for reliable monitoring [5]. To overcome these limitations, the use of prior knowledge from architectural plans reflected in the BIM [18] can be used to guide the mapping.

**Global localization with priors.** A common strategy to integrate prior knowledge in SLAM is to estimate the robot’s pose directly within a pre-existing map using particle filters such as Monte Carlo Localization [19]. With LiDAR input, this approach has been extended to leverage 3D architectural priors for building-scale localization [8] and to demonstrate real-world BIM-based pipelines [20]. In parallel, vision-based methods such as [11] build an object-centric map by aligning monocular RGB detections to a 2D floor plan and perform global localization via Monte Carlo localization. Other recent approaches construct semantic scene graphs from images conditioned on a prebuilt 3D LiDAR map, improving robustness with learning-based inpainting and space partitioning while keeping computation low [21].

While effective for pose estimation, these pipelines rely solely on the prior map for localization and do not continuously incorporate new sensor observations to refine an evolving *as-built* map. Additionally, they typically require substantial exploration or overlap with the prior before achieving a reliable match, delaying initialization and limiting their applicability for monitoring small or partially constructed areas.

**BIM integration in SLAM systems.** To address these limitations, recent works examine the possibility of coupling BIM with SLAM so that architectural priors directly regularize the trajectory and keep the evolving map consistent with the design. *BIM-SLAM* [7] integrates BIM into multi-session 3D LiDAR SLAM by seeding a pose-graph from the model, anchoring subsequent sessions to the BIM, and reconstructing new elements not present in the plans. It reduces long-term drift without prior robot pose and generates a BIM-aligned lifelong map for indoor environments.

Tightly coupled *as-build* and *as-planned* formulations further exploit architectural structure. *iS-Graphs* first derive an Architectural Graph (A-Graph) from BIM and merge it with the online *S-Graph* [22] to form an informed graph, enabling global robot localization with room semantics [1]. Follow-up work optimizes this joint graph and leverages architectural constraints to measure discrepancies between the plan and observations [10]. These approaches, while benefiting from LiDAR-based high accuracy and 360° coverage for room detection, rely on room-level semantics for global localization. This makes them vulnerable in symmetric environments and prevents monitoring of smaller areas, such as a single room, without extensive exploration.

A few works have extended these ideas to vision-based SLAM. For example, [23] integrates visual SLAM with detailed digital twins to suppress drift; however, this approach relies on an accurate digital twin that includes detailed visual information, which may not be readily available in evolving indoor construction environments. To the best of our knowledge, no other visual SLAM method leverages prior BIM to maintain a precise, drift-bounded *as-built* map suitable for construction monitoring.

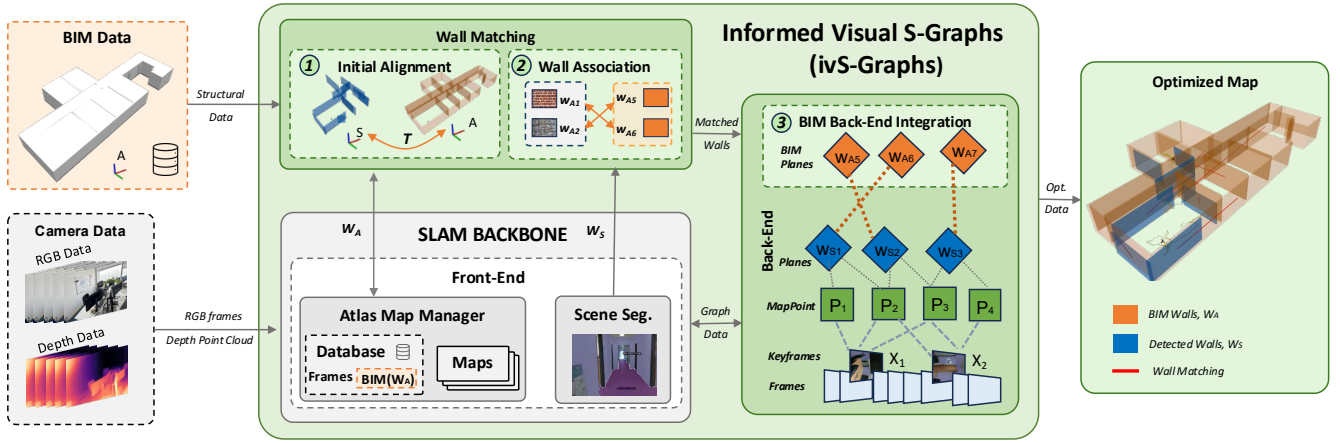


Fig. 2: System architecture of *ivS-Graphs*. The pipeline takes BIM and RGB-D camera data as inputs. The SLAM backbone front-end processes visual data into keyframes, map points, and wall segments. Our contributions are highlighted in green: (1) initial alignment followed by a (2) continuous wall association and (3) the integration of BIM in the back-end of the system. These establish BIM-to-SLAM correspondences ( $W_A \leftrightarrow W_S$ ) that are introduced as constraints into the back-end graph. The resulting optimized map (right) aligns the evolving as-built structure with the as-planned BIM.

### III. METHODOLOGY

#### A. System Overview

We build upon *vS-Graphs* [14], a visual SLAM system that extends *ORB-SLAM3* [13] by introducing structural concepts into the pipeline. This choice is motivated by its ability to represent planar elements such as walls, which is particularly useful since they are common in structured indoor construction environments. An overview of the proposed system architecture is shown in Fig. 2.

**Inputs.** The pipeline receives two main inputs: (i) RGB-D data, and (ii) a set of architectural walls extracted from the BIM with the specification of two BIM wall IDs expected to be observed first. Each wall, either detected or BIM-extracted wall, is represented as a structured object:

$$w_i = \begin{cases} \text{id}_i, & \text{unique wall identifier} \\ \pi_i = (\mathbf{n}_i, d_i), & \text{supporting plane: normal and distance} \\ \mathbf{c}_i, & \text{centroid position} \\ l_i, t_i, & \text{length and thickness of the wall} \end{cases} \quad (1)$$

Here,  $\mathbf{n}_i \in \mathbb{R}^3$  and  $d_i \in \mathbb{R}$  define the planar component  $\pi_i$  of the wall.  $\mathbf{c}_i \in \mathbb{R}^3$  is the centroid of the wall, and  $l_i, t_i$  are its length and thickness.

We denote  $\mathbf{W}$  as a set of walls and use  $\pi$  to refer specifically to the planar component of a wall in the back-end optimization.

The architecture is divided into three main modules:

**1) SLAM backbone (Section III-B).** *vS-Graphs* [14] is used as our backbone, providing capabilities for feature tracking, local mapping, and structural element extraction. Its graph-based scene representation serves as the foundation for integrating BIM priors in the back-end.

**2) Wall Matching (Section III-C).** The wall matching algorithm establishes associations between BIM walls  $\mathbf{W}_A$  and detected walls  $\mathbf{W}_S$  throughout the mapping process.

It enables an initial alignment with the BIM layout from an early stages of operation and continues to refine wall associations as new walls are observed.

**3) BIM-Constrained Back-End (Section III-D).** The matched pairs of walls are incorporated into a factor graph that combines visual constraints with architectural priors. By introducing BIM walls as fixed nodes and linking them to detected walls, this integration propagates architectural consistency through graph optimization.

#### B. SLAM Backbone

The *vS-Graphs* system [14] provides a feature-based front-end and a graph-based back-end for structural mapping. On the one hand, it processes RGB-D frames through its front-end. Of particular relevance are the *Scene Segmentor*, which detects planar surfaces from the depth data and extracts  $\mathbf{W}_S$ , and the *ATLAS* map manager, which handles remapping and stores the BIM data alongside the evolving map.

On the other hand, the back-end is a factor graph where entities encode both geometric and structural information of the scene:

**Keyframes:** Camera poses represented as  $\text{SE}(3)$  nodes, connected through visual odometry constraints.

**Map points:** 3D landmarks constrained by reprojection factors linking them to keyframes.

**Walls:** Extracted from planar surfaces detected in the *Scene Segmentor*, they are connected to the map points that observe them via pose-wall constraints.

This graph-based representation forms the foundation for our BIM integration in the back-end.

#### C. Wall Matching

The wall matching module establishes correspondences between BIM walls  $\mathbf{W}_A$  and detected walls  $\mathbf{W}_S$ , provided respectively by the BIM input and the SLAM backbone front-end. This step is essential for enabling BIM-constrained

---

**Algorithm 1** Wall Matching Procedure

---

```
1: Input:  $\mathbf{W}_A$ ,  $\mathbf{W}_S$   $\triangleright$  BIM and detected walls
2: Output:  $\mathcal{M}$   $\triangleright$  Pair of matched walls

3: Step 1: Initial Alignment
4: repeat
5:   Find  $w_{S_i}, w_{S_j}$  that  $|\mathbf{n}_{S_i} \perp \mathbf{n}_{S_j}| < \tau_\perp$ 
6:   Match  $w_{S_i}, w_{S_j} \rightarrow w_{A_1}, w_{A_2}$ 
7:   Estimate  ${}^S\hat{T}_A$  via Least Square Method
8: until residual  $< \tau_{init}$ 

9: Step 2: Continuous Matching
10: for each  $w_S \in \mathbf{W}_S$  do
11:   For each BIM wall  $w_A \in \mathbf{W}_A$  satisfying:
      $d_{\text{plane}}(\pi_S, \pi_A) \leq \tau_p$  and  $d_{\text{centroid}}^\perp(w_S, w_A) \leq \tau_c \|\mathbf{c}_S - \mathbf{c}_A\|$ 
12:   Compute combined score:  $s(w_S, w_A)$ 
13:   Best result:  $w_A^* = \arg \min s(w_S, w_A)$ 
14:    $\mathcal{M} \leftarrow (w_S, w_A^*)$ 
15: end for
```

---

SLAM, as it both initializes the relative transformation  ${}^S\hat{T}_A$  between the BIM frame  $A$  and the SLAM frame  $S$ , and continuously maintains valid associations as the map evolves. The overall procedure is explained in Algorithm 1, which is organized in two stages: (1) initial alignment and (2) continuous matching.

**Initial Alignment:** Before any BIM correction is applied, the two coordinate frames must be aligned. The only prior required is the specification of two BIM wall IDs expected to be observed first,  $id_{A_1}$  and  $id_{A_2}$ . The system then waits until it observes two walls  $(w_{S_i}, w_{S_j})$  whose supporting planes are nearly perpendicular, i.e.,  $\pi_{S_i} \perp \pi_{S_j}$ . This ensures the plane equations system is linearly independent, which is necessary to compute a valid rigid transformation and helps select the intended reference walls for initialization.

Once these two pairs are found, the rigid transformation  ${}^S\hat{T}_A \in \text{SE}(3)$  that aligns the BIM frame  $A$  to the SLAM map frame  $S$  is estimated by minimizing the residuals between corresponding planes using a least-squares method:

$${}^S\hat{T}_A = \arg \min_{T \in \text{SE}(3)} \sum_{k=1}^2 \|f(T, \pi_{A_k}) - \pi_{S_k}\|^2 \quad (2)$$

where  $f(T, \pi_A)$  applies the rigid transform  $T = (R, \mathbf{t})$  to the BIM plane  $\pi_A$ . Once  ${}^S\hat{T}_A$  is estimated, it is applied to all  $\mathbf{W}_A$  as explained in other systems [22], so that they are re-expressed in the SLAM frame as  $\mathbf{W}_A^S$ , and stored in the Atlas Manager for subsequent continuous association with detected walls  $\mathbf{W}_S^S$ . This initialization assumes that the first observed walls approximately match their BIM counterparts. Small deviations in the matched  $w_S$  are tolerated due to the least-squares estimation of  ${}^S\hat{T}_A$ . This design allows for robust initialization from almost the start of exploration, without requiring additional landmarks or prior manual alignment.

**Continuous Matching:** After initialization, the system updates wall associations as new walls are detected, refining

the alignment between the SLAM map and the BIM. For each detected wall  $w_{S_i}$ , two complementary measures are evaluated against each BIM wall  $w_{A_j}$ :

**1) Plane-parameter distance (PPD).** The distance between a detected plane  $\pi_{S_i}$  and a BIM plane  $\pi_{A_j}$  is computed using their minimal parameter vectors as:

$$d_{\text{plane}}(\pi_{S_i}, \pi_{A_j}) = \sqrt{(\pi_{S_i} \ominus \pi_{A_j})^\top (\pi_{S_i} \ominus \pi_{A_j})} \quad (3)$$

where  $\ominus$  denotes the difference between plane parameters.

**2) Projected centroid distance (PCD).** To account for the finite extent of walls, we also compare the centroids of  $w_{S_i}$  and  $w_{A_j}$  along the wall normal:

$$d_{\text{centroid}}^\perp(w_{S_i}, w_{A_j}) = |(\mathbf{c}_{S_i} - \mathbf{c}_{A_j})^\top \mathbf{n}_{A_j}| \quad (4)$$

This penalizes matches to BIM walls that are far from the detected wall segment, even if their infinite planes overlap.

**3) Combined score.** The plane-parameter distance (PPD) and the projected centroid distance (PCD) are combined into a normalized weighted score:

$$s(w_{S_i}, w_{A_j}) = \alpha \frac{d_{\text{plane}}(\pi_{S_i}, \pi_{A_j})}{\tau_p} + (1-\alpha) \frac{d_{\text{centroid}}^\perp(w_{S_i}, w_{A_j})}{\tau_c} \quad (5)$$

where  $\tau_p$  and  $\tau_c$  normalize the two distances to make them comparable, and  $\alpha \in [0, 1]$  controls the relative importance of the plane parameters versus the centroid proximity. In our experiments, we set  $\alpha$  to 0.7. The candidate BIM wall  $w_{A_j}$  with the lowest score  $s(w_{S_i}, w_{A_j})$  is selected as the association for the detected wall  $w_{S_i}$ . All such associations are collected into the set  $\mathcal{M} = \{(w_{S_i}, w_{A_j})\}$ .

#### D. BIM-Constrained Back-End Integration

This module integrates the matched walls into the SLAM optimization back-end by extending the factor graph with additional wall nodes. With the creation of edges between the matched pairs, it creates constraints to the evolving map to the *as-planned* layout.

**Graph Construction.** The back-end factor graph is built integrating components from the backbone: keyframes  $X$ , map points  $P$ , and detected structural walls  $\mathbf{W}_S$ . After the initial alignment (Sec. III-C), BIM walls  $\mathbf{W}_A$  are introduced as fixed wall nodes in the graph, while detected walls  $\mathbf{W}_S$  remain optimizable.

Wall-to-wall factors are created from the associations  $\mathcal{M} = \{(w_{S_i}, w_{A_j})\}$  computed by the wall matching module. The cost function for the wall-to-wall factor is defined as:

$$c_{\text{wall-wall}}(w_s, w_a) = \|w_{S_i} \ominus w_{A_j}\|_{\Lambda_{w_{ij}}}^2 \quad (6)$$

where  $\Lambda_{w_{ij}}$  is the covariance matrix associated with the uncertainty of the plane association. This uncertainty is derived from the wall matching score  $s(w_{S_i}, w_{A_j})$  defined in Sec. III-C. Pairs with higher confidence in the matching are assigned smaller covariances, enforcing stronger alignment, while less confident matches receive larger covariances to reduce their influence.  $\Lambda_{w_{ij}}$  is computed from the matching score as:

$$\Lambda_{w_{ij}} = \mathbf{I} \cdot \epsilon \cdot s(w_{S_i}, w_{A_j}) \quad (7)$$

where  $\epsilon$  is a small constant used to scale the weight for numerical stability and  $I$  is the identity matrix of appropriate dimension. This ensures that the factor is scaled according to the confidence of the wall association.

**Graph Optimization.** The defined factor graph is periodically optimized during local mapping to maintain global consistency and incorporate architectural restrictions when new wall associations are available.

The total cost function of the system is composed of three types of residuals: visual reprojection errors and pose-to-wall constraints, as explained in [14], and BIM wall-to-wall factors (our contribution):

$$c_{\text{total}} = \sum_{(x,p) \in (X,P)} c_{\text{reproj}}(x,p) + \sum_{(x,w_s) \in (X,W_S)} c_{\text{pose-wall}}(x,w_s) + \sum_{(w_s,w_a) \in \mathcal{M}} c_{\text{wall-wall}}(w_s,w_a) \quad (8)$$

The optimizer adjusts the variables  $\{X,P,W_S\}$  to minimize  $c_{\text{total}}$ . The gradient of each residual propagates corrections to the connected keyframes and map points, ensuring that matched BIM walls influence the SLAM map according to their confidence. After optimization, keyframe poses and map points are updated in the Atlas map, while BIM walls remain fixed. Corrections also propagate across sessions through the existing map merging and loop closing mechanism, as in the baseline [14].

Associations with large residuals are automatically down-weighted using a robust Huber kernel [24], [25], which reduces their influence on the optimization. This is particularly important in construction environments, where partial structures or occlusions can cause incorrect wall detections and, consequently, erroneous wall associations.

#### IV. EXPERIMENTAL RESULTS

##### A. Validation Methodology

**Datasets.** Public indoor datasets (e.g., [26]) typically provide RGB-D data, but to the best of our knowledge, no public dataset includes both RGB-D and the corresponding BIM, which limits the evaluation of our system. To address this, we collected real-world sequences that combine RGB-D streams with provided BIM data, along with LiDAR point clouds for ground-truth generation. RGB-D data is captured with an *Intel RealSense D435i*, and point clouds with a *Velodyne VLP-16 LiDAR*, all integrated within the *SMapper* device [27].

The sequences span multiple indoor layouts as summarized in Table I. On the one hand, we have offices with different trajectories and spaces, and on the other hand, we have construction sites at different stages of completion. Specifically, in the offices, we have two buildings, each with its respective BIM model. Sequence of offices include different sequences of trajectories that correspond to various

TABLE I: Summary of real-world sequences used for evaluation. “Spaces” counts distinct navigable spaces (rooms, corridors, and the kitchen). “Area (m<sup>2</sup>)” is the approximate floor area covered by the sequence, and “Duration (s)” is the recording time.

Env.	Sequence	#Walls	#Spaces	Area (m <sup>2</sup> )	Duration (s)
Offices	office1-1	10	4	100	670
	office1-2	12	3	109	265
	office1-3	12	4	113	524
	office1-4	6	2	54	169
	office2-1	10	3	168	268
	office2-2	10	3	168	370
	office2-3	10	3	70	172
	office2-4	4	4	40	260
Constr. Sites	CS1	16	5	48	377
	CS2-MidFurn	17	5	68	215
	CS3-Furn	13	4	95	180
Total		120	40	1033	3470

navigable spaces, such as rooms and corridors. In the construction sites, we have three different locations, each with its own BIM model. These sites represent different stages of construction: *CS1* is in the early construction phase, with visible electrical cables and pipes; *CS2* is in a mid-phase, where main elements are constructed but some work is still ongoing; and *CS3* is in a nearly finished phase, where some furniture are already in place.

**Baselines.** To assess the impact of BIM integration on visual SLAM, we evaluate against established RGB-D SLAM systems such as *ORB-SLAM3* [13] and *BAD-SLAM* [15]. These are chosen as baselines because of their suitability, reported results, and availability of their code. In addition, since our approach builds on *vS-Graphs* [14], the comparison with it will allow us to isolate the effect of adding BIM wall constraints.

**Implementation Details.** All algorithms run on a workstation with an *Intel® Core™ i9-11950H* (2.60 GHz), an *NVIDIA T600 Mobile GPU* (4 GB), and 32 GB RAM.

##### B. Evaluation Experiments

**1) Trajectory Estimation Performance:** We evaluate trajectory accuracy using the Absolute Trajectory Error (ATE), which measures the deviation of the estimated trajectory from the reference trajectories obtained by processing LiDAR scans with *S-Graphs* [22]. This metric is important because errors in the trajectory accumulate over time, leading to degradation in the reconstructed map. Table II summarizes results for each dataset and baseline.

With the aim of providing qualitative context to the evaluation, we include a top-down overlay in Fig. 3 in one of the university datasets. For visualization purposes, only the initial semantic alignment is used to display the *as-planned* data in the *vS-Graphs* baseline on the right. This allows us to visually assess the implication of BIM constraints in the trajectory estimation.



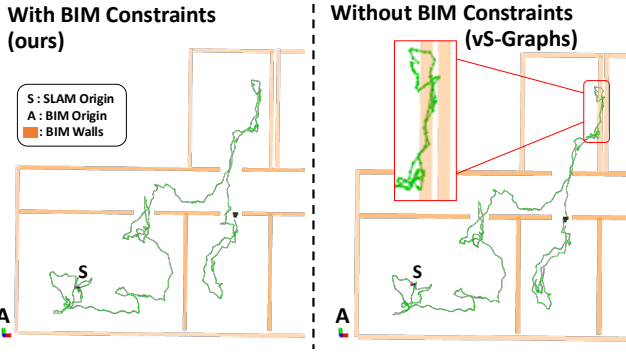


Fig. 3: Qualitative comparison of estimated trajectories on the *office1-1* sequence. Left: proposed method with BIM constraints; right: baseline without BIM constraints, just initial alignment integrated.

2) **Mapping Performance:** We evaluate the quality of the reconstructed maps by computing the Root Mean Square Error (RMSE) between the reconstructed point cloud and the reference BIM point cloud, which serves as ground truth. This metric directly reflects the geometric consistency of the map with respect to the as-planned design. In Table III We compare the RMSE of *ivS-Graphs* against baseline methods.

3) **Robustness to Partially Built Environments:** To evaluate the robustness of the system to partially constructed environments, since no such dataset is readily available (see Table I), we conduct an experiment where we add BIM walls at logical locations, such as where dividing rooms or spaces would make sense. These walls are not present in the physical environment.

The objective is to simulate a scenario with missing walls,

TABLE II: Absolute Trajectory Error (ATE) in meters (m) for the evaluated VSLAM algorithms across the collected indoor datasets. The best and second-best results are **boldfaced** and underlined, respectively. Values are averaged over five runs, excluding any failed tracking attempts. Dashes indicate unavailable data due to tracking failures. The final column reports the percentage improvement of *ivS-Graphs* over the *vS-Graphs* baseline.

Sequence	ATE (m)				
	<i>BAD SLAM</i> [15]	<i>ORB-SLAM</i> 3.0 [13]	<i>vS-Graphs</i> [14]	<i>ivS-Graphs</i> (ours)	Diff. (%)
office1-1	-	0.462	<u>0.343</u>	<b>0.226</b>	+33.99%
office1-2	1.653	0.365	<u>0.353</u>	<b>0.183</b>	+48.15%
office1-3	-	4.674	<u>0.253</u>	<b>0.175</b>	+30.83%
office1-4	-	<u>0.101</u>	<b>0.080</b>	0.120	-33.24%
office2-1	1.879	<b>0.160</b>	0.223	<u>0.214</u>	+4.04%
office2-2	-	0.461	<u>0.348</u>	<b>0.224</b>	+35.76%
office2-3	-	4.946	<u>0.280</u>	<b>0.176</b>	+36.85%
office2-4	7.926	0.553	<u>0.386</u>	<b>0.123</b>	+68.10%
CS1	2.798	0.136	<u>0.112</u>	<b>0.100</b>	+10.51%
CS2-MidFurn	1.383	1.252	<u>0.158</u>	<b>0.156</b>	+1.22%
CS3-Furn	-	2.964	<u>0.254</u>	<b>0.149</b>	+41.38%
<b>Mean</b>	2.826	1.461	<u>0.254</u>	<b>0.168</b>	+23.71%

TABLE III: Root Mean Square Error (RMSE) in meters (m) of the reconstructed point clouds for the evaluated VSLAM algorithms across the collected indoor datasets. The best and second-best results are **boldfaced** and underlined, respectively. Dashes indicate unavailable data due to mapping failures. The final column reports the percentage improvement of *ivS-Graphs* over *vS-Graphs* baseline.

Sequence	Point Cloud RMSE (m)				
	<i>BAD SLAM</i> [15]	<i>ORB-SLAM</i> 3.0 [13]	<i>vS-Graphs</i> [14]	<i>ivS-Graphs</i> (ours)	Diff. (%)
office1-1	-	0.416	<u>0.403</u>	<b>0.331</b>	+17.88%
office1-2	<u>0.442</u>	0.537	0.444	<b>0.396</b>	+10.72%
office1-3	-	0.360	<b>0.340</b>	<u>0.354</u>	-4.15%
office1-4	-	0.349	<u>0.327</u>	<b>0.306</b>	+6.35%
office2-1	<u>0.394</u>	0.407	0.432	<b>0.382</b>	11.55%
office2-2	-	0.487	<b>0.470</b>	<u>0.473</u>	-0.80%
office2-3	-	<u>0.412</u>	0.481	<b>0.411</b>	14.60%
office2-4	-	<b>0.386</b>	0.546	<u>0.504</u>	7.64%
CS1	0.299	<u>0.282</u>	0.307	<b>0.277</b>	+9.63%
CS2-MidFurn	<b>0.357</b>	0.425	0.378	<u>0.374</u>	+3.62%
CS3-Furn	-	0.588	<u>0.298</u>	<b>0.287</b>	+0.99%
<b>Mean</b>	-	0.422	<u>0.402</u>	<b>0.372</b>	+7.14%

reflecting real-world conditions where environments are still under construction. The SLAM pipeline remains unchanged, with plane-to-wall factors being created only for walls that meet the association criteria. We compute the ATE following the protocol outlined in Section IV-B.1. Table IV presents the trajectory evaluation results under various conditions, demonstrating how the system handles different levels of missing walls.

4) **Runtime Analysis:** To verify that the system operates in real time, we measured the processing frame rate (FPS) and the time required for initial alignment across four representative sequences as shown in Table V.

### C. Results and Discussion

From Table II can be observed that *ivS-Graphs* reduces mean ATE relative to the baseline by 23.71% on average. These improvements are more noticeable in larger environments, where drift accumulates in pure visual SLAM systems. For example, in smaller environments like *office1-4*, the drift is minimal, and there is not improvements over the baseline. However, when sequences start to be larger,

TABLE IV: Absolute Trajectory Error (ATE) [m] of *ivS-Graphs* across different levels of incomplete buildings. The best and second-best results are **boldfaced** and underlined, respectively. Each value is averaged over 5 runs. The final column reports the similarity in ATE between the nominal case and the maximum missing-wall condition.

Sequence	Nominal	20% missing	30% missing	Similarity (%)
office1-1	<b>0.120</b>	<u>0.126</u>	0.132	89.41%
CS1	<u>0.100</u>	0.104	<b>0.094</b>	93.80%
CS2-MidFurn	<b>0.156</b>	0.188	<u>0.166</u>	93.35%

TABLE V: Runtime statistics for *ivS-Graphs* across representative sequences. FPS is averaged over the entire trajectory. Initial alignment time corresponds to the time required to compute  $\hat{T}_A$  after detecting the first two walls..

Sequence	<i>vS-Graphs</i> FPS [14]	<i>ivS-Graphs</i> FPS (ours)	Initial Alignment Time (ms)
office1-1	23.8	21.8	0.244
office1-2	25.5	24.9	0.150
CS1	23.3	22.5	0.234
CS2-MidFurn	24.2	23.9	0.337
<b>Mean</b>	<b>24.2</b>	<b>23.3</b>	<b>0.241</b>

the baseline systems shows an increasing drift over time, whereas *ivS-Graphs* gradually corrects the drift, maintaining a more stable error throughout the duration of the sequence. As an example of this behavior, Fig. 4 shows how the trajectory error increases linearly for the baseline systems, while our approach creates wall association constraints that mitigate drift and maintain a more stable ATE.

As shown in the qualitative comparison of Fig. 3, the trajectory estimated by *ivS-Graphs* with BIM constraints (left) is topologically consistent with the floor layout, whereas the baseline *vS-Graphs* (right) exhibits deviations, with the trajectory crossing through BIM walls.

Mapping performance follows a trend similar to the trajectory results. As shown in Table III, in smaller environments the RMSE of *ivS-Graphs* is comparable to or slightly better than the baseline, as drift does not yet accumulate significantly. However, in larger environments, such as *office1-1*, *office1-2*, and *CS1*, *ivS-Graphs* consistently outperforms the baseline, with RMSE improvements ranging from +9.63% to +17.88%. This shows that the integration of BIM constraints has a significant impact on map consistency in larger environments, whereas in smaller environments the impact of BIM is minimal and mapping performance is primarily influenced by the SLAM backbone.

By simulating partially built environments, the robustness experiment shows that *ivS-Graphs* maintains a similarity of 92.19% in ATE when up to 30% of walls are missing due to incomplete construction as shown in Table IV. This indicates that the system can operate robustly in realistic scenarios where some walls are not yet built, with trajectory accuracy remaining largely consistent under these conditions.

Finally, we report runtime analysis to confirm that the system operates in real time as illustrated in Table V. Across the experiments, *ivS-Graphs* consistently operated above the commonly accepted threshold of 20FPS for real-time applications. The baseline without the BIM achieved processing rates between 23.3 and 25.5FPS, while our *ivS-Graphs* implementation reached between 21.8 and 24.9FPS, with an average of 23.3FPS compared to 24.2FPS for the baseline. This shows that the introduction of BIM constraints does not incur a significant computational overhead, allowing continuous monitoring of construction progress without sacrificing efficiency. In addition, the average time required to perform the initial alignment after detecting the first two

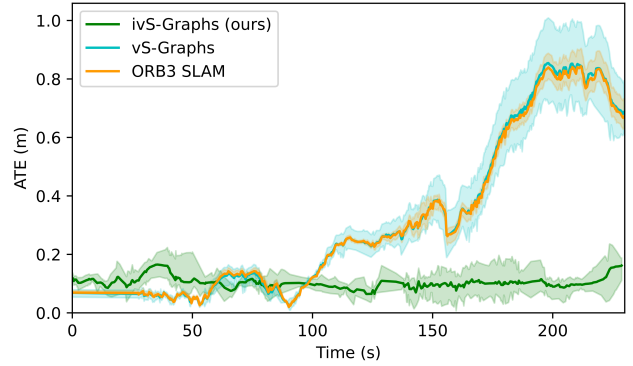


Fig. 4: Accumulated trajectory drift error over time in *office1-2* sequence, showing by the ATE for baselines and our *ivS-Graphs* approach except *BAD SLAM* due to tracking failures.

walls is only 0.241 ms, which is negligible for the starting of monitoring.

## V. CONCLUSIONS AND FUTURE WORKS

In this work we presented *ivS-Graphs*, a BIM-informed RGB-D SLAM system that reduces drift and improves map consistency for construction monitoring. Validation on both construction sites and offices shows that integrating BIM priors into the SLAM back-end enhances long-term monitoring, reducing ATE by 23.71% and RMSE by 7.14% compared to state-of-the-art visual SLAM baselines. These improvements are more significant in larger environments, where drift accumulates faster, highlighting *ivS-Graphs*' effectiveness in complex scenarios.

In future work, we aim to integrate additional BIM elements, such as floors, columns, and pipes, to further enhance the alignment between the *as-built* and *as-planned* maps, even in smaller environments where trajectory drift is limited. Building on this, we plan to investigate alternative initialization strategies that leverage these additional semantic cues to provide more informative initial alignment, reducing the need for specifying BIM wall IDs and minimizing the required environment coverage.

## REFERENCES

- [1] M. Shaheer, J. A. Millan-Romera, H. Bavle, J. L. Sanchez-Lopez, J. Civera, and H. Voos, "Graph-based global robot localization informing situational graphs with architectural graphs," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 9155–9162.
- [2] E. Yigitbas, A. Nowosad, and G. Engels, "Supporting construction and architectural visualization through bim and ar/vr: A systematic literature review," in *IFIP Conference on Human-Computer Interaction*, Springer, 2023, pp. 145–166.
- [3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [4] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, L. Nogueira, M. Palieri, P. Petráček, M. Petrлік, A. Reinke, V. Krátký, S. Zhao, A.-a. Agha-mohammadi, K. Alexis, C. Heckman, K. Khosoussi, N. Kottege, B. Morrell, M. Hutter, F. Pauling, F. Pomerleau, M. Saska, S. Scherer, R. Siegwart, J. L. Williams, and L. Carlone, "Present and future

- of slam in extreme environments: The darpa sub challenge,” *IEEE Transactions on Robotics*, vol. 40, pp. 936–959, 2024.
- [5] T. Sun, Y. Hao, S. Huang, S. Savarese, K. Schindler, M. Pollefeys, and I. Armeni, “Nothing stands still: A spatiotemporal benchmark on 3d point cloud registration under large geometric and temporal change,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 220, pp. 799–823, 2025.
  - [6] A. Braun, S. Tutas, U. Stilla, and A. Borrmann, “Bim-based progress monitoring,” in *Building Information Modeling: Technology Foundations and Industry Practice*. Springer, 2018, pp. 463–476.
  - [7] M. A. V. Torres, A. Braun, and A. Borrmann, “Bim-slam: Integrating bim models in multi-session slam for lifelong mapping using 3d lidar,” *arXiv preprint arXiv:2408.15870*, 2024.
  - [8] H. Blum, J. Stiefel, C. Cadena, R. Siegwart, and A. Gawel, “Precise robot localization in architectural 3d plans,” *arXiv preprint arXiv:2006.05137*, 2020.
  - [9] H. Yin, J. Liew, W. Lee, M. Ang Jr, and J. Yeoh, “Towards BIM-based robot localization: a real-world case study,” *International Symposium on Automation and Robotics in Construction*, 2022.
  - [10] M. Shaheer, J. A. Millan-Romera, H. Bavle, M. Giberna, J. L. Sanchez-Lopez, J. Civera, and H. Voos, “Tightly coupled slam with imprecise architectural plans,” *IEEE Robotics and Automation Letters*, vol. 10, no. 8, pp. 8019–8026, 2025.
  - [11] N. Zimmerman, M. Sodano, E. Marks, J. Behley, and C. Stachniss, “Constructing Metric-Semantic Maps Using Floor Plan Priors for Long-Term Indoor Localization,” in *International Conference on Intelligent Robots and Systems*, Oct. 2023, pp. 1366–1372.
  - [12] A. Bikandi, M. Shaheer, H. Bavle, J. Jevanesan, H. Voos, and J. L. Sanchez-Lopez, “Bim-constrained optimization for accurate localization and deviation correction in construction monitoring,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.17693>
  - [13] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
  - [14] A. Tourani, S. Ejaz, H. Bavle, D. Morilla-Cabello, J. L. Sanchez-Lopez, and H. Voos, “vs-graphs: Integrating visual slam and situational graphs through multi-level scene understanding,” *arXiv preprint arXiv:2503.01783*, 2025.
  - [15] T. Schöps, T. Sattler, and M. Pollefeys, “Bad slam: Bundle adjusted direct rgb-d slam,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 134–144.
  - [16] A. Korovko, D. Slepichev, A. Efitov, A. Dzhamuratova, V. Kuznetsov, H. Rabeti, and J. Biswas, “cuvslam: Cuda accelerated visual odometry,” *arXiv preprint arXiv:2506.04359*, 2025.
  - [17] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.
  - [18] V. Thein, “Industry foundation classes (ifc),” *BIM interoperability through a vendor-independent file format*, p. 152, 2011.
  - [19] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte carlo localization for mobile robots,” in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, vol. 2. IEEE, 1999, pp. 1322–1328.
  - [20] H. Yin, J. Liew, W. Lee, M. Ang Jr, and J. Yeoh, “Towards BIM-based robot localization: a real-world case study,” *International Symposium on Automation and Robotics in Construction*, 2022.
  - [21] A. Longo, C. Chung, M. Palieri, S.-K. Kim, A. Agha, C. Guaragnella, and S. Khattak, “Pixels-to-graph: Real-time integration of building information models and scene graphs for semantic-geometric human-robot understanding,” *arXiv preprint arXiv:2506.22593*, 2025.
  - [22] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, “S-graphs 2.0—a hierarchical-semantic optimization and loop closure for slam,” *arXiv preprint arXiv:2502.18044*, 2025.
  - [23] R. Merat, G. Cioffi, L. Bauersfeld, and D. Scaramuzza, “Drift-free visual slam using digital twins,” *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1633–1640, 2025.
  - [24] J. A. Gallego, F. A. González, and O. Nasraoui, “Robust kernels for robust location estimation,” *Neurocomputing*, vol. 429, pp. 174–186, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231220317033>
  - [25] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.
  - [26] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.
  - [27] P. M. B. Soares, A. Tourani, M. Fernandez-Cortizas, A. B. Noya, J. L. Sanchez-Lopez, and H. Voos, “Smapper: A multi-modal data acquisition platform for slam benchmarking,” 2025. [Online]. Available: <https://arxiv.org/abs/2509.09509>