EG-ICE 2025
GLASGOW

Automated construction progress monitoring coupling lidar inertial odometry and building information modeling

STÜHRENBERG, J.[1,*], TANDON, A.[1] & SMARSLY, K.[1]

[1]Institute of Digital and Autonomous Construction, Hamburg University of Technology, Hamburg, Germany
*Correspondence: jan.stuehrenberg@tuhh.de

ABSTRACT:
Building information modeling (BIM) models contain valuable information for mobile robots deployed to increase the efficiency of automated building inspections. To access the information, robots require accurate localization relative to BIM models. Since BIM models represent the "as-planned" rather than the "as-built" state, localization is challenging due to deviations between BIM models and the reality. In this paper, lidar inertial odometry (LIO) and BIM are coupled in the "LIO-BIM" framework for robust and accurate mobile robot localization and mapping relative to BIM models. LIO-BIM overlays the as-built map with BIM models, enabling automated progress monitoring. The framework is implemented and validated on the ConSLAM dataset, which includes periodically collected data from a 3D lidar, an inertial measurement unit, and a camera at a construction site. The validation tests show robust and accurate 3D localization and mapping relative to BIM models in real-time, enabling effective automated progress monitoring.

KEYWORDS:
Building information modeling (BIM), simultaneous localization and mapping (SLAM), scan matching, factor graphs, progress monitoring

1. INTRODUCTION

Construction projects may be delayed due to weather, human error, or supply chain disruptions (Lee *et al.*, 2018). Therefore, construction progress monitoring is conducted to enable corrective actions mitigating the impact of delays. Traditionally, project managers visit construction sites to monitor construction progress manually (Qu *et al.*, 2017).

To improve inspection frequency and operational efficiency, mobile robots can be employed to automate data collection and construction progress monitoring (Samsami, 2024).

Building Information Modeling (BIM) has become a cornerstone in the construction industry. BIM models provide rich semantic and geometric data critical for mobile robots deployed for automated building inspections and progress monitoring to enhance operational efficiency (Halder & Afsari, 2023).

However, leveraging the information embedded in BIM models, to perform automated building inspections and progress monitoring using robots requires precise localization of robots relative to the models (Smarsly et al. 2023). A challenge arises because BIM models typically represent the "as-

Stührenberg, J., Tandon, A., & Smarsly, K. Automated construction progress monitoring coupling lidar inertial odometry and building information modeling. In A. Moreno-Rangel, & B. Kumar (Eds.), EG-ICE 2025: AI-Driven Collaboration for Sustainable and Resilient Built Environments Conference Proceedings University of Strathclyde Publishing. https://doi.org/10.17868/strath.00093227

planned" state of a building, whereas real-world deviations lead to discrepancies between the BIM models and the "as-built" conditions (Vega Torres *et al.*, 2022).

Comparing as-built point clouds with BIM models to perform progress monitoring is challenging due to measurement errors and obstacles present on construction sites, such as scaffolding, building materials, or machinery, which may cause occlusions (Kavaliauskas *et al.*, 2022).

To localize robots relative to BIM models, 2D occupancy grid maps are generated from models and the adaptive Monte Carlo localization (AMCL) method is employed (Follini *et al.*, 2021). However, AMCL is bound to 2D and graph-based simultaneous localization and mapping (SLAM) approaches, such as the 2D SLAM framework developed in (Boniardi *et al.*, 2019), show improvements in accuracy in the pose tracking problem compared to AMCL. 2D approaches implicitly assume planar vertical walls and planar horizontal floors, limiting the applicability and accuracy of the approaches. In (Vega Torres *et al.*, 2024), robots are localized in 3D using synthetically generated session data in BIM models, but the framework developed is not real-time capable.

Automated progress monitoring approaches using mobile lidar devices, as proposed by Vassena *et al.* (2023) and Kavaliauskas *et al.* (2022), generate point clouds independently from the BIM model. The point clouds are subsequently registered to the BIM models, which can be a computationally intensive process.

To address the aforementioned issues, this paper couples lidar inertial odometry (LIO) and BIM in the "LIO-BIM" framework for robust and accurate mobile robot localization and mapping relative to BIM models. Building upon previous work of the authors (Tandon *et al.*, 2025a,b) and efficient SLAM techniques using 3D lidar, LIO-BIM overlays the as-built map with the BIM model in real time, enabling effective automated progress monitoring.

To determine whether building elements are built on time or delayed, confidence indicators and ray tracing indicators are calculated for each building element present in the BIM model. Based on the confidence and ray tracing indicators, the building elements are classified as *scanned*, *unscanned*, or *delayed*.

The remainder of the paper is organized as follows: In the next section, the methodology and implementation of the LIO-BIM framework and the automated progress monitoring approach are presented. In Section 3, tests conducted to validate the LIO-BIM framework and the automated progress monitoring approach are described, and the test results are displayed and discussed. Section 4 concludes the paper and gives an outlook on future work.

2. AUTOMATED CONSTRUCTION PROGRESS MONITORING USING LIO-BIM

The LIO-BIM framework extends LIO-SAM, a state-of-the-art SLAM framework (Shan *et al.*, 2020). LIO-BIM obtains sensor data from a camera, a 3D lidar, and an inertial measurement unit (IMU), and it calculates robot trajectories and point clouds of the environment relative to BIM models. LIO-BIM operates using several coordinate frames, i.e. coordinate systems: Sensor data is received in the robot frame **R** to estimate the robot trajectories and the point clouds in the map frame **M**. The BIM model is defined in the BIM model frame **B**. Using the transformation ${}^{B}\mathbf{T}_{M}$, the robot trajectories and point clouds are aligned with the BIM model frame **B**. An overview of the flow chart of LIO-BIM is shown in Fig. 1. The following subsections 2.1-2.4 discuss the steps undertaken in LIO-BIM to enable robust and accurate 3D localization and mapping relative to BIM models in real-time. The last subsection, Subsection 2.5, presents an approach to perform automated construction progress monitoring using the map, i.e. the point cloud, created by LIO-BIM.

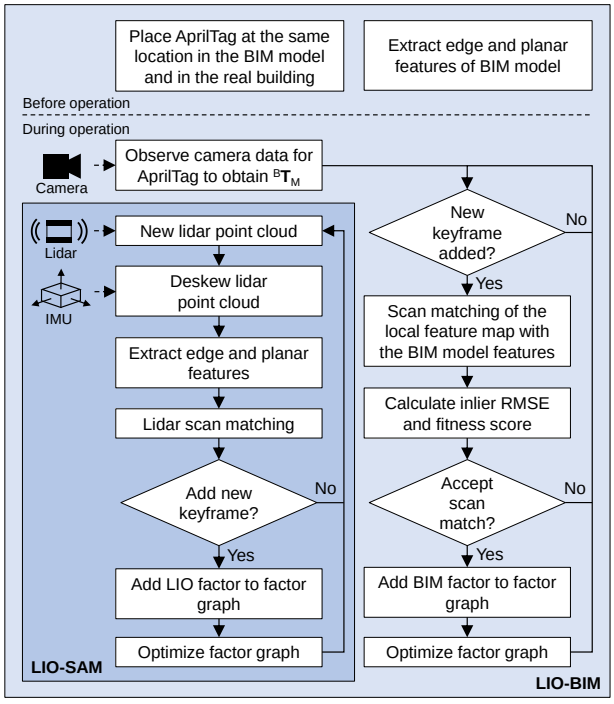

Figure 1: Flow chart of LIO-BIM

2.1 Extracting edge and planar features from BIM models

LIO-BIM performs efficient scan matching using a feature-based iterative closest point (ICP) algorithm. Feature points are selected on sharp edges and flat surface patches in point clouds. Scan matching is performed for the latest lidar point cloud

with immediately preceding recorded lidar point clouds for LIO as detailed in (Zhang & Singh, 2017), and for the local map around the robot with the building described by the BIM model. To enable scan matching with the BIM model, two steps are conducted. The first step involves the extraction of edge and planar feature points of the BIM model prior to operation. The process is shown in Figure 2. The second step is detailed in Section 2.2.

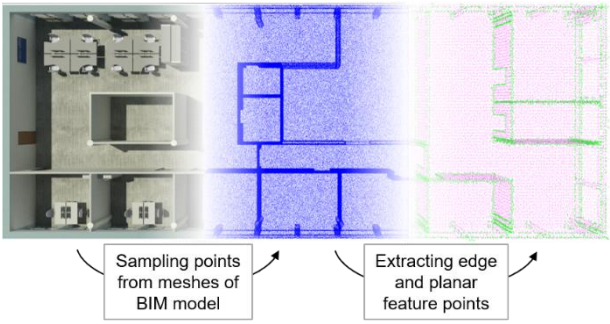

Figure 2: Extracting feature points of BIM models

First, the BIM model is exported to the Industry Foundation Classes (IFC) format. The IfcOpenShell is used to iterate over all elements in the IFC file (IfcOpenShell, 2025). Elements potentially deteriorating the quality of scan matching with BIM models during operation are excluded, such as windows, which refract laser beams from lidars. The remaining elements are converted into a mesh that represents all surfaces of the building. Then, a point cloud is generated by uniformly sampling points from the meshes using the Point Cloud Library (PCL) (Rusu & Cousins, 2011). By utilizing the surface normal, the principal curvatures k_1 and k_2, indicating the minimum and maximum curvature values, of sampled points are estimated. Based on the principal curvatures k_1 and k_2, the curvature score k_B is calculated according to Equation (1):

$$k_B = \text{abs}(k_1) + \text{abs}(k_2) \quad (1)$$

Edge and planar feature points are selected based on the curvature score (edge feature points: $k_B > 0.6$, planar feature points: $k_B < 0.1$). To improve the scan matching efficiency, described in detail in Section 2.3, the edge and planar feature point clouds must be sparsely distributed. To retain the most significant edge and planar feature points, the points are ranked based on their curvature score. Neighboring points within a small radius around the most prominent feature points are removed. Finally, the sparse feature point clouds are saved in the Point Cloud Data (PCD) file format.

Extracting edge and planar feature points is the first of two steps to enable scan matching with the BIM model. The second step involves obtaining an initial alignment with the BIM model using fiducial tags, as detailed in the following subsection.

2.2 Using fiducial tags to obtain an initial alignment with the BIM model

Fiducial tags are placed in the BIM model and at the corresponding locations in the real building before operation. LIO-BIM utilizes AprilTags, i.e. 2D tags similar to barcodes with a specific ID number (Olson, 2011).

During operation of the LIO-BIM framework, lidar and IMU data are used to perform LIO in the map frame. The camera data is observed for fiducial tags. Upon detecting a fiducial tag, the transformation ${}^{B}\mathbf{T}_{M}$ aligning the map frame with the BIM model frame is obtained by leveraging the correspondence of the real tag and the virtual tag in the BIM model. The transformation chain to obtain ${}^{B}\mathbf{T}_{M}$, described in the following paragraph, is shown in Figure 3.

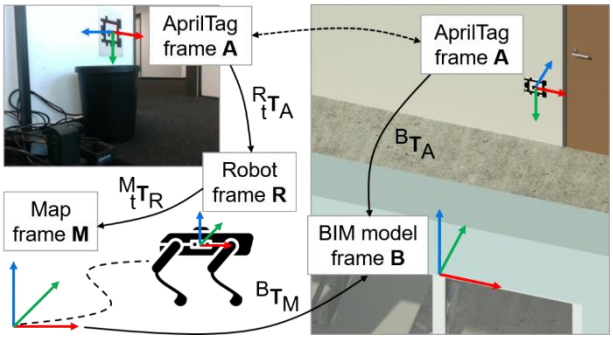

Figure 3: Obtaining the transformation ${}^{B}\mathbf{T}_{M}$ using fiducial tags

The transformation ${}^{B}\mathbf{T}_{A}$ from the AprilTag frame **A** to the BIM model frame **B** is embedded in the coordinates of AprilTags in the BIM model. The coordinates are obtained by parsing the IFC file for *IfcBuildingElementProxy* elements with object types labeled “AprilTag” and the corresponding ID number. The transformation ${}^{R}_{t}\mathbf{T}_{A}$ from the AprilTag frame **A** to the camera in the robot frame **R** at time *t* is computed using the AprilTag visual fiducial detector (AprilRobotics, 2025). The transformation ${}^{M}_{t}\mathbf{T}_{R}$ from the robot frame **R** to the map frame **M** at time *t* is estimated using the odometry of the SLAM algorithm, following the original LIO-SAM implementation up to this stage. Combining all transformations in Equation (2) yields the transformation ${}^{B}\mathbf{T}_{M}$:

$${}^{B}\mathbf{T}_{M} = {}^{B}\mathbf{T}_{A}\left({}^{M}_{t}\mathbf{T}_{R}\,{}^{R}_{t}\mathbf{T}_{A}\right)^{-1} \quad (2)$$

The transformation ${}^{B}\mathbf{T}_{M}$ is adjusted to meet the coordinate system requirements of the BIM model and the map, where the z-axis must point upward, opposite to the direction of gravity. Pitch and roll

angles present in ${}^{B}\mathbf{T}_{M}$ due to inaccuracies or measurement errors are corrected to zero. The transformation ${}^{B}\mathbf{T}_{M}$ serves as the initial estimate for scan matching between the local feature map and the BIM model, as discussed in the next subsection.

2.3 Scan matching of the local feature map with the BIM model

Once the transformation ${}^{B}\mathbf{T}_{M}$ is calculated, scan matching of the local map around the robot, referred to as the "local feature map", with the BIM model features begins. LIO-BIM reuses lidar point clouds already processed in the LIO pipeline, where edge and planar feature points are extracted based on the smoothness in a local area, for efficiency. While LIO is computed for every lidar point cloud, scan matching with the BIM model feature point cloud is performed using a reduced set of keyframes. Keyframes are added when the robot exceeds user-defined thresholds for distance changes (robot traverses > 1m or > 0.2 rad). The local feature map is constructed using all keyframes in a 5 m radius around the robot.

The scan matching is visualized in Figure 4. Utilizing a feature-based ICP algorithm, point-to-line distances of edge features and point-to-plane distances of planar features are minimized by the Levenberg-Marquardt algorithm to calculate the refined transformation ${}^{B}_{i}\mathbf{T}_{M}^{opt.}$. Hereby, the transformation ${}^{B}\mathbf{T}_{M}$ is used as the starting value. The ICP algorithm runs until convergence or aborts after 30 iterations. Based on a distance threshold of 0.6 m, the quality of converged solutions of ${}^{B}_{i}\mathbf{T}_{M}^{opt.}$ is evaluated by the root mean square error (RMSE) of all inlier correspondences and the fitness score measuring the percentage of inlier correspondences. The RMSE of inlier correspondences is defined as in Equation (3):

$$ {}_{i}\mathrm{RMSE}_{d}=\sqrt{\frac{1}{|\mathbf{S}_{d}|}\sum_{\mathbf{p}_{M}\in\mathbf{S}_{d}}\left\|\mathrm{NN}\left({}^{B}_{i}\mathbf{T}_{M}^{opt.}\,\mathbf{p}_{M}\right)-{}^{B}_{i}\mathbf{T}_{M}^{opt.}\,\mathbf{p}_{M}\right\|^{2}} \quad (3) $$

where $|\mathbf{S}_{d}|$ denotes the number of all points in the inlier set $\mathbf{S}_{d}$ from the local feature map, $\mathbf{p}_{M}$ denotes a feature point in the inlier set, and $\mathrm{NN}\left({}^{B}_{i}\mathbf{T}_{M}^{opt.}\,\mathbf{p}_{M}\right)$ finds the nearest neighbor of $\mathbf{p}_{M}$ transformed to the BIM model frame **B** in the feature point cloud of the BIM model. The fitness score equals the number of inlier points $|\mathbf{S}_{d}|$ divided by the number of points in the local feature map.

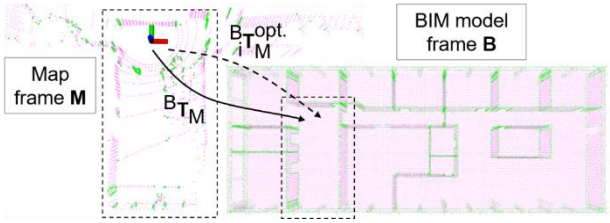

Figure 4: Scan matching of the local feature map with the BIM model

Refined transformations ${}^{B}_{i}\mathbf{T}_{M}^{opt.}$ with a low inlier RMSE (< 0.1 m) and a high fitness score (> 0.65) indicate high compliance of aligning the local feature map with the BIM model. To ensure accuracy and reliability, low-compliant scan matches are rejected. LIO-BIM only includes high-compliant scan matches to integrate so-called BIM factors in the factor graph, as explained in the next subsection.

2.4 Integration of BIM factors in the factor graph

Whenever the local feature map around a keyframe aligns with the BIM model with high compliance, a BIM factor associated with the keyframe is integrated into the underlying factor graph of LIO-BIM, implemented using the Georgia Tech Smoothing and Mapping (GTSAM) library (Dellaert & Kaess, 2017). BIM factors are unary factors that correct the robot trajectories and maps to align with BIM models. Since the factor graph is defined in the map frame **M**, refined transformations ${}^{B}_{i}\mathbf{T}_{M}^{opt.}$, representing a local correction of the transformation ${}^{B}\mathbf{T}_{M}$, are adjusted to derive the BIM factor ${}_{i}\mathbf{T}_{M}^{BIM}$. The BIM factors ${}_{i}\mathbf{T}_{M}^{BIM}$ are computed according to Equation (4):

$$ {}_{i}\mathbf{T}_{M}^{BIM}={}^{B}\mathbf{T}_{M}^{-1}\,{}^{B}_{i}\mathbf{T}_{M}^{opt.}\,{}^{M}_{i}\mathbf{T}_{R} \quad (4) $$

where ${}^{M}_{i}\mathbf{T}_{R}$ denotes the pose of the last keyframe i in the map frame **M**. The BIM factor is integrated with independent Gaussian noise, where the variance is set to the inlier RMSE calculated in Equation (3) for each of its six dimensions. Once a BIM factor is integrated into the factor graph, optimization is performed using the method proposed in (Kaess et al., 2012). The optimization of the factor graph computes the maximum a posteriori solution, ensuring accurate robot trajectories and mapping. To prevent redundant BIM factors in confined spaces and reduce computational complexity, scan matching with the BIM model is skipped for three keyframes after BIM factors are integrated. The following subsection details, how the as-built point cloud created by LIO-BIM and aligned with BIM models, is used to perform automated progress monitoring.

2.5 Automated progress monitoring

The output of LIO-BIM, an as-built point cloud aligned with the as-planned BIM model, can be used without registration for automated construction progress monitoring. The point clouds may vary in resolution and quality, depending on the environmental conditions, the lidar, and the parameters. Therefore, several indicators are calculated to objectively assess the confidence in the geometric quality and the presence (or absence) of building elements in the BIM models. Indicators assessing the coverage, the distance, and the distribution of building elements based on point clouds have been introduced in (Malihi *et al.*, 2024) in a scan-to-BIM context. The calculation of the three indicators is based on voxelization. Each building element is voxelized with a resolution δ. The coverage indicator I_{cov} is then calculated using Equation (5):

$$I_{cov} = \frac{|\gamma_c|}{|\gamma_m|} \quad (5)$$

where γ_m denotes the set of voxels intersecting the building element mesh, and γ_c denotes a subset of γ_m that contains points from the point cloud. The coverage indicator quantifies how much surface of the building elements is explained by the point cloud. Instead of using a fixed voxel resolution δ, this study utilizes adaptive voxelization based on the shape and size of individual building elements to compute optimal voxel resolutions for varying geometries and point cloud densities. The optimal voxel resolution maximizes the coverage indicator, where γ_m contains only voxels with at least n points. The optimal voxel resolution is computed using a binary search within a range determined by the bounding box dimensions of the building elements. The distribution and distance indicators are then calculated using the optimal voxel resolution according to the formulae given in (Malihi *et al.*, 2024).

While the coverage, distance, and distribution indicators allow the assessment of the confidence in the geometric quality of building elements, the indicators do not allow the distinction between building elements that are either delayed or unscanned due to occlusion. To assess the potential visibility of building elements, and therefore whether building elements are delayed or unscanned, this study introduces the ray tracing indicator $I_{raytracing}$. To calculate the ray tracing indicator, the point cloud created by LIO-BIM is sliced at a given radius r around the building elements and voxelized using the optimal voxel resolution. For each uncovered voxel in the voxel set $\gamma_m - \gamma_c$, an unobstructed line is searched from the voxel to the positions in the trajectory of LIO-BIM within the radius. Furthermore, the line must not be blocked by other voxels in γ_m and must lie within the field of view of the lidar. If only one unobstructed line is found, the voxel is considered potentially visible, otherwise, it is occluded. Finally, the ray tracing indicator is calculated according to Equation (6):

$$I_{rt} = \frac{|\gamma_{visible}|}{|\gamma_m|} \quad (6)$$

where $\gamma_{visible}$ denotes the set of potentially visible voxels of building elements.

The building elements in the BIM model are classified as *scanned*, *unscanned*, or *delayed* as follows: The coverage, distance, and distribution indicators are combined in the confidence indicator according to Equation (7):

$$I_{conf} = 0.7*I_{cov} + 0.15*I_{dist} + 0.15*I_{distr} \quad (7)$$

Four thresholds are introduced to classify the building elements: Two confidence thresholds $c_{conf,1}$ and $c_{conf,2}$ and two ray tracing thresholds $c_{rt,1}$ and $c_{rt,2}$. Building elements with a high confidence indicator that exceed the confidence threshold $c_{conf,1}$ represent high geometric quality and are classified as *scanned*.

Building elements with a mediocre confidence indicator, between the confidence thresholds $c_{conf,1}$ and $c_{conf,2}$, are further examined by the ray tracing indicator. There are two main reasons for mediocre confidence indicators: If the building element is built, the confidence is negatively affected by occluded parts of the building element. If the building element is delayed, the confidence indicator is positively affected by clutter in the same location as the building element. Therefore, building elements with a ray tracing indicator below the ray tracing threshold $c_{rt,1}$ are classified as *scanned*, while building elements exceeding $c_{rt,1}$ are classified as *delayed*.

Building elements with a small confidence indicator below $c_{conf,2}$ represent either delayed or occluded building elements. Occluded building elements are characterized by a small ray tracing indicator below the ray tracing threshold $c_{rt,2}$ and the building elements are classified as *unscanned*. Building elements exceeding $c_{rt,2}$ are classified as *delayed*.

In the following section, the validation tests and results are presented.

3. VALIDATION TESTS AND RESULTS

This section describes the tests conducted to validate the LIO-BIM framework and the automated progress monitoring approach. First, the objectives and metrics of the validation tests are defined in Subsection 3.1. Subsequently, the process of the

validation tests is described in Subsection 3.2. Finally, the results of the tests are presented and discussed in Subsection 3.3.

3.1 Objectives of the validation tests

The validation tests are designed to validate (i) the localization accuracy of LIO-BIM, (ii) the mapping accuracy of LIO-BIM, (iii) the real-time capabilities of LIO-BIM, and (iv) the precision and recall of the automated progress monitoring approach.

The ***localization accuracy*** is assessed by comparing the trajectories created by LIO-BIM against ground truth trajectories using the absolute pose error (APE). Corresponding poses in the trajectories created by LIO-BIM and the ground truth trajectories are identified by matching timestamps. Let ${}_i\hat{\mathbf{p}}_{est}$ denote an estimated pose from the trajectory created by LIO-BIM and let ${}_i\hat{\mathbf{p}}_{ref}$ denote a reference pose from the ground truth trajectory at timestamp *i*. The absolute relative pose $\mathbf{E}_i$ is calculated according to Equation (8):

$$\mathbf{E}_i = {}_i\hat{\mathbf{p}}_{est}^{-1}\ {}_i\hat{\mathbf{p}}_{ref} \tag{8}$$

Then, translational and rotational APEs are obtained using Equations (9) and (10):

$$tAPE_i = \|trans(\mathbf{E}_i)\| \tag{9}$$

$$rAPE_i = \left|angle\left(\log_{SO(3)}(rot(\mathbf{E}_i))\right)\right| \tag{10}$$

The translational and rotational APEs of all timestamps in the trajectories are combined using the RMSE.

The ***mapping accuracy*** is validated by comparing the point cloud created by LIO-BIM against point clouds created by a terrestrial laser scanner (TLS), which is considered the ground truth. The accuracy is assessed by calculating the inlier RMSE of the point cloud created by LIO-BIM against the TLS point cloud with a distance threshold chosen as 0.3 m.

The ***real-time capabilities*** of LIO-BIM are assessed by observing processing times, the numbers of skipped lidar scans to compute the LIO, and skipped keyframes for the scan matching with the BIM model. Lidar scans are skipped if the processing times of the LIO exceed the times of incoming lidar point clouds (the lidar operates at 10 Hz). Keyframes are skipped if processing times of the scan matching with the BIM model exceed the times of keyframes being added. If lidar scans and keyframes are skipped only occasionally during operation, LIO-BIM is considered to be real-time capable.

The ***precision and recall*** of the automated progress monitoring approach are assessed by determining delayed building elements using the point cloud created by LIO-BIM and by counting the true positives, false positives, and false negatives. In the validation tests, the parameters are chosen according to Table 1.

Table 1: Parameters for the automated progress monitoring approach used in the validation tests

Parameter	n	r	$c_{conf,1}$	$c_{conf,2}$	$c_{rt,1}$	$c_{rt,2}$
Value	5	10 m	0.5	0.35	0.4	0.25

3.2 Description of the validation tests

The LIO-BIM framework is validated on the ConSLAM dataset (Trzeciak et al., 2023). The data has been recorded through a handheld system (consisting of a 3D lidar, an IMU, and a camera) for five days over a period of 4.5 months, and it comprises five data sequences. The first data sequence contains faulty IMU measurements, therefore the LIO-BIM framework is validated on the remaining sequences number two (S2), three (S3), four (S4), and five (S5). Furthermore, the dataset includes TLS scans for each sequence. The BIM model created based on the TLS scan of S2 and supplied by Vega Torres *et al.* (2024) is used in the validation tests.

The validation tests are conducted as follows: AprilTags are placed in the BIM model at the corresponding locations where the AprilTags appear in the camera images of the dataset. The feature points are extracted from the BIM model as described in Subsection 2.1, yielding 82,503 edge feature points and 1,110,114 planar feature points. The data of S2, S3, S4, and S5 is played back to the LIO-BIM framework. At the end of each sequence, the trajectory consisting of timestamps and poses for every keyframe, the point cloud, processing times, numbers of skipped keyframes, and numbers of skipped lidar frames are saved. The trajectories created by LIO-BIM are compared against the ground truth trajectories provided by the ConSLAM dataset. According to Vega Torres *et al.* (2024), discrepancies may be present in the ground truth trajectories provided by the ConSLAM dataset. Hence, the trajectories created by LIO-BIM are also compared against trajectories calculated using the SLAM2REF framework (Vega Torres *et al.* 2024). The point cloud created by LIO-BIM is compared against the TLS point clouds provided in the ConSLAM dataset. The processing times and skipped keyframes of respective lidar frames are observed to assess the real-time capabilities.

To detect delayed building elements, the automated progress monitoring approach iterates over each wall and column present in the BIM model. Thirteen walls and ten columns are placed

additionally in the BIM model to represent delayed building elements. The classification is performed based on the confidence indicators and ray tracing indicators. The recall and precision of the delayed building elements are calculated for S2, S3, S4, and S5. Furthermore, the processing times of the automated progress monitoring approach are recorded. All computations are performed on the Intel NUC11TNKV7 mini PC equipped with the Intel Core i7-1185G7 CPU and 32 GB of RAM (Intel, 2023).

3.3 Results of the validation tests

This subsection presents the results regarding the localization accuracy, mapping accuracy, and real-time capabilities of LIO-BIM, and the precision and recall of the automated progress monitoring approach.

The results of the ***localization accuracy*** of LIO-BIM are summarized in Table 2 and Table 3, which display the translational and rotational errors. Table 2 shows the comparison against the ground truth trajectory provided in the ConSLAM dataset. Table 3 shows the comparison against the ground truth provided in Vega Torres *et al.* (2024). Figure 5 displays the translational error mapped onto the trajectories created by LIO-BIM compared against the ground truth trajectories provided in Vega Torres *et al.* (2024) for each sequence.

Table 2: Localization accuracy of LIO-BIM against the ground truth provided in the ConSLAM dataset

Metric		S2	S3	S4	S5
tAPE [cm]	RMSE	10.21	10.40	12.85	15.68
	Max	103.84	32.06	37.29	73.66
rAPE [deg]	RMSE	1.3665	1.3791	1.1199	1.4438
	Max	16.8973	7.5847	4.2863	7.1668

Table 3: Localization accuracy of LIO-BIM against the ground truth provided in Vega Torres et al. (2024)

Metric		S2	S3	S4	S5
tAPE [cm]	RMSE	5.97	26.68	10.25	12.57
	Max	20.67	195.70	23.20	34.88
rAPE [deg]	RMSE	0.7992	1.4730	1.0910	1.2362
	Max	4.4042	7.7004	4.5813	6.8999

The trajectories created by LIO-BIM exhibit errors within the low centimeter range. Overall, the comparison with the ground truth provided in Vega Torres *et al.* (2024) results in lower errors, except for S3. In S3, outliers are visible in small parts of the trajectory, as shown in Figure 5. Since the trajectory created by LIO-BIM aligns closely with the ground truth trajectory from the ConSLAM dataset in the relevant section, the outlier may stem from an error

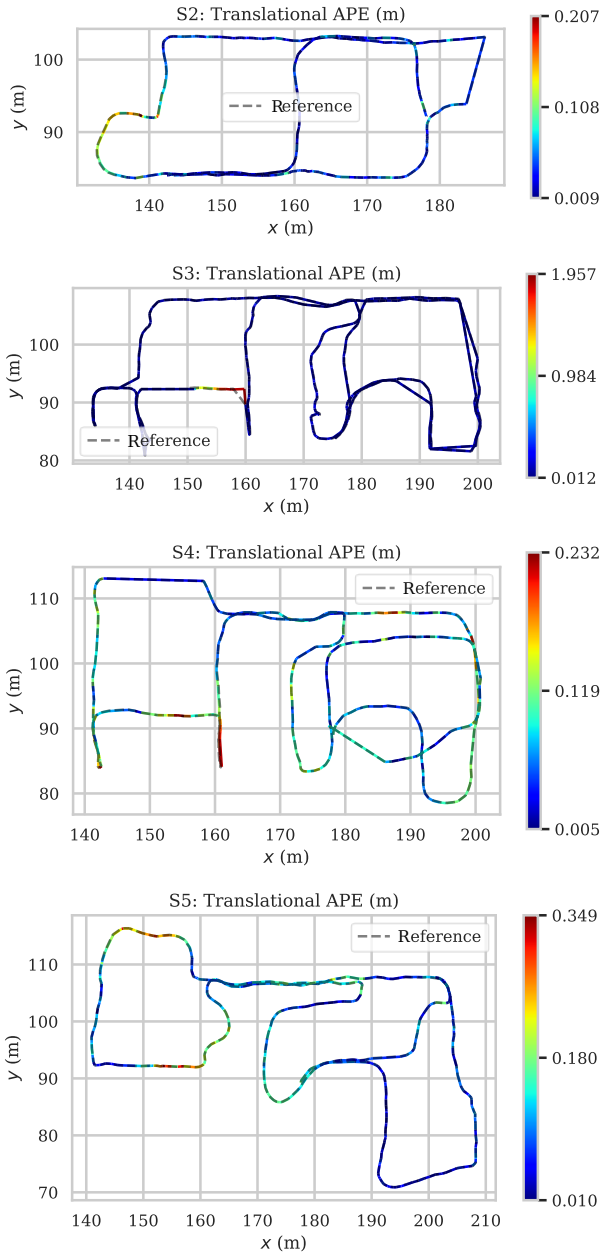

Figure 5: Translational absolute pose error mapped onto trajectories created by LIO-BIM

in the trajectory provided in Vega Torres *et al.* (2024).

The results of the ***mapping accuracy*** are provided in Table 4, where inlier RMSEs and fitness scores of the point clouds created by LIO-BIM are compared to the TLS point clouds using a distance threshold of 0.3 m. The alignment of the point cloud created LIO-BIM and the TLS point cloud of S3 are visualized in Figure 6. A color gradient is used to display point-to-point distances ranging from green (0 cm) to yellow (15 cm) to red (30 cm and above) showing outliers and errors in the point cloud created by LIO-BIM.

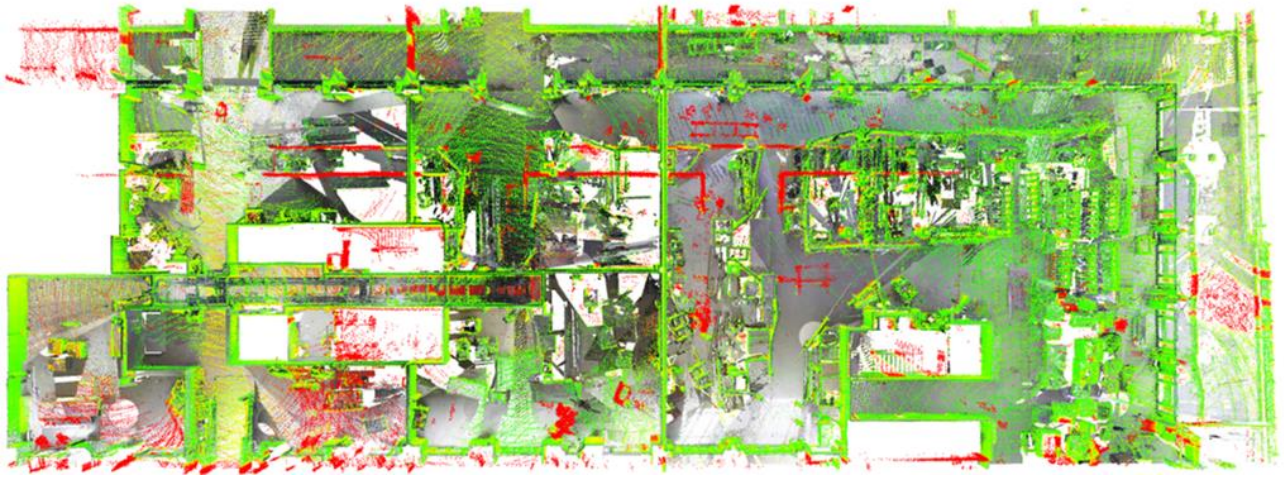

Figure 6: Alignment of point cloud created by LIO-BIM and TLS point cloud of S3, the color gradient displays point-to-point distances ranging from green (0 cm) to yellow (15 cm) to red (30 cm and above)

Table 4: Mapping accuracy of LIO-BIM with a distance threshold of 0.3 m

Metric	S2	S3	S4	S5
$RMSE_{0.3}$ [cm]	6.33	6.30	6.38	7.89
Fitness score	0.9776	0.9650	0.9292	0.9290

Similar to the trajectories, the point clouds created by LIO-BIM exhibit errors within the low centimeter range. The fitness scores indicate a high overlap between the point clouds created by LIO-BIM and the TLS point clouds. As can be seen in Figure 6, most outliers are present in areas observed with the lidars, but not with the TLS. However, some outliers are present originating from reflections in windows.

The results of the ***real-time capability*** of LIO-BIM are shown in Table 5 and Table 6. Table 5 displays the number of total and skipped keyframes and the mean, standard deviation, and maximum of processing times of the scan matching with the BIM model. Table 6 displays the number of total and skipped lidar frames and the same statistics regarding the processing times of the lidar scan matching.

Table 5: Processing times of the scan matching with the BIM model

Metric		S2	S3	S4	S5
Key-frames	Total	399	712	529	542
	Skipped	99	236	165	128
Times	Mean [s]	1.481	1.717	1.582	1.766
	SD [s]	0.768	0.848	0.745	0.924
	Max [s]	3.177	3.253	3.227	3.362

Table 6: Processing times of the lidar scan matching

Metric		S2	S3	S4	S5
Lidar frames	Total	4163	6244	5125	5835
	Skipped	255	856	683	836
Times	Mean [s]	0.082	0.094	0.092	0.095
	SD [s]	0.041	0.053	0.054	0.055
	Max [s]	0.487	0.511	0.838	0.449

Table 5 presents the percentage of keyframes skipped, ranging from 23.616 % (S5) to 33.146 % (S3). With a local map radius set to five meters and keyframes added after every meter of movement or an angular change exceeding 0.2 radians, consecutive local feature maps around keyframes maintain sufficient overlap. As a result, scan matching with the BIM model is considered to operate in real time. Table 6 shows that between 6.125 % (S2) and 14.327 % (S5) of lidar frames are skipped. Since the lidar frames are skipped only occasionally, the lidar scan matching for LIO is considered to operate in real time, as well.

The ***precision and recall*** results are shown in Table 7. In all sequences, the automated progress monitoring approach achieves a high precision of more than 90 % for detecting delayed building elements. While high recall results are also achieved in S2, S3, and S4, the recall in S5 is relatively low due to occlusions. Here, some of the additional building elements are classified as *unscanned* instead of *delayed*.

Table 7: Precision and recall of the automated progress monitoring approach

Metric	S2	S3	S4	S5
TP	23	19	22	13
FP	1	1	0	1
FN	0	4	1	10
Precision	0.958	0.950	1.000	0.929
Recall	1.000	0.884	0.957	0.565
F1 score	0.979	0.884	0.978	0.703

The output of the automated progress monitoring approach is visualized in Figure 7 for S2, S3, S4, and S5. Scanned building elements are displayed using a color gradient representing the confidence indicator. Delayed building elements are visualized in blue color. Unscanned building elements are displayed in gray color.

Table 8 shows the computation times of the automated progress monitoring approach. As can be seen, fast data collection and classification is facilitated, indicating speedups compared to traditional methods, such as manual TLS.

Table 8: Data collection times and computation times of the automated progress monitoring approach

Metric	S2	S3	S4	S5
Data collection times [min:s]	7:00	10:30	8:37	9:49
Processing times [min:s]	11:18	13:31	13:19	14:21

The following and final section summarizes the paper and draws conclusions.

4. SUMMARY AND CONCLUSIONS

Mobile robots deployed to automate building inspections and construction progress monitoring can leverage information stored in BIM models to enhance operational efficiency. Accessing the information requires accurate robot localization relative to the BIM models, which is challenging due to deviations between BIM models and reality. Mapping the reality using 3D lidars allows the comparison of point clouds with the BIM model to identify delays in construction projects.

The results of the validation tests show robust and accurate 3D localization and mapping relative to BIM models in real time. The automated progress monitoring approach detects delayed building elements on construction sites with high precision and is faster than traditional methods, such as manual terrestrial laser scanning. In conclusion, the LIO-BIM framework and the automated progress monitoring approach provide an efficient alternative to traditional methods.

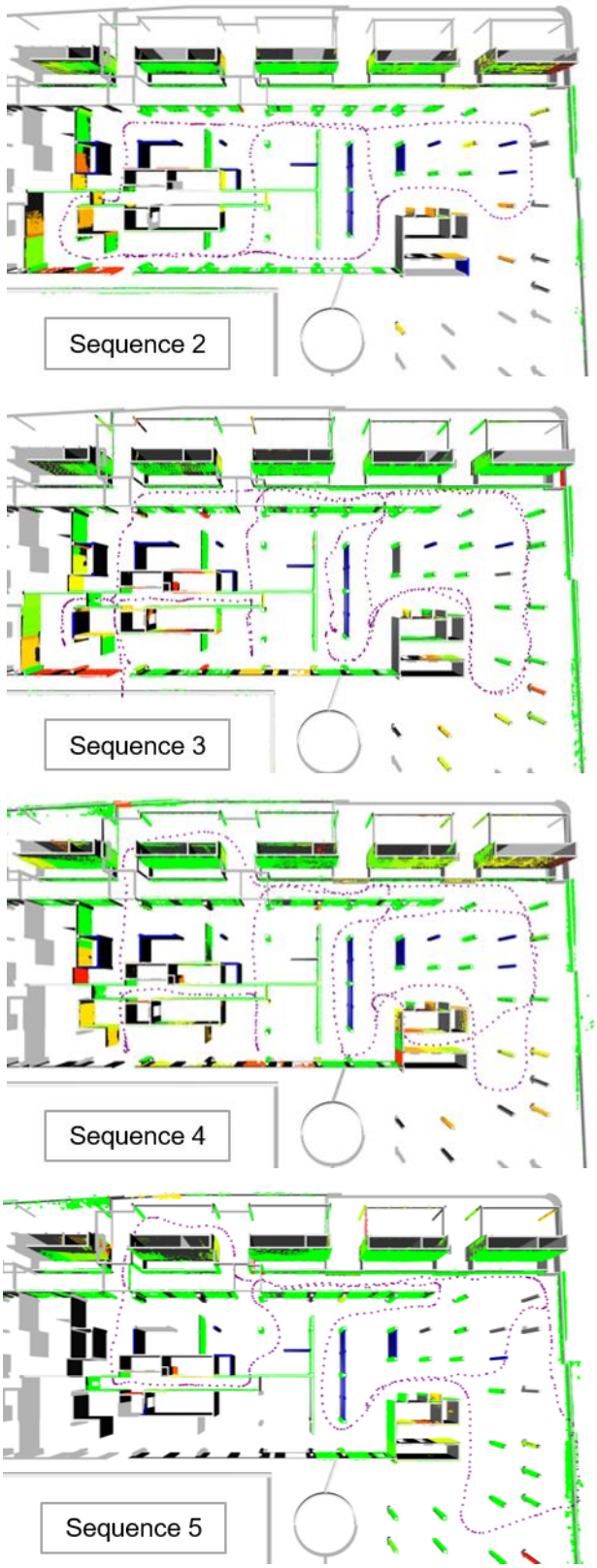

Figure 7: Visualization of the progress monitoring, the color gradient from red (0.350) to yellow (0.425) to green (0.500 and above) displays scanned building elements, blue indicates delayed building elements, gray indicates unscanned building elements

In future work, LIO-BIM may be improved by using different techniques other than fiducial tags to obtain an initial alignment with BIM models that do not require modifications to the environment. Furthermore, the problem of lidar reflections may be investigated. Finally, changes and delays detected in the real building may be automatically integrated into the BIM model.

REFERENCES

AprilRobotics (2025). AprilTag 3, [Online], Available: https://github.com/AprilRobotics/apriltag [6 February 2025].

Boniardi, F., Caselitz, T., Kümmerle, R., & Burgard, W., (2019). A pose graph-based localization system for long-term navigation in CAD floor plans. *Robotics and Autonomous Systems*, 112: p. 84-97.

Dellaert, F. & Kaess, M., (2017). Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1-2): p. 1-139.

Follini, C., Magnago, V., Freitag, K., Terzer, M., Marcher, C., Riedl, M., Giusti, A., & Matt, D., (2021). BIM-integrated collaborative robotics for application in building construction and maintenance. *Robotics*, 10(1): 2.

Halder, S. & Afsari, K., (2023). Robots in inspection and monitoring of buildings and infrastructure: A systematic review. *Applied Sciences*, 13(4): 2304.

IfcOpenShell (2025). IfcOpenShell: The open source IFC toolkit and geometry engine, [Online], Available: https://ifcopenshell.org/ [6 February 2025].

Intel (2023). Intel NUC Board/Kit/Mini PC: NUC11TNi3 / NUC11TNi5 / NUC11TNv5 / NUC11TNi7 / NUC11TNv7 Technical Product Specification, [Online], Available: https://www.intel.com/content/dam/support/us/en/documents/intel-nuc/NUC11TN_TechProdSpec.pdf [12 February 2025].

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F., (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2): p. 216-235.

Kavaliauskas, P., Fernandez, J.B., McGuinness, K., & Jurelionis, A., (2022). Automation of construction progress monitoring by integrating 3D point cloud data with an IFC-based BIM model. *Buildings*, 12(10): 1754.

Lee, J.H., Park, J.-H., & Jan, B.-T., (2018). Design of Robot based Work Progress Monitoring System for the Building Construction Site. In *Proceedings of the 2018 International Conference on Information and Communication Technology Convergence.* Jeju, South Korea, November 18.

Malihi, S. & Bosché, F., (2024). Integrating surface-related indicators of coverage, distance and distribution for quantifying scan-to-BIM confidence level. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4-2024: p. 223-229.

Olson, E., (2011). AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*. Shanghai, China, May 9.

Qu, T., Zang, W., Peng, Z., Liu, J., Li, W., Zhu, Y., Zhang, B., & Wang, Y., (2017). Construction Site Monitoring Using UAV Oblique Photogrammetry and BIM Technologies. In *Proceedings of 22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia*. Suzhou, China, April 5.

Rusu, R.B. & Cousins, S., (2011). 3D is here: Point Cloud Library (PCL). In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation.* Shanghai, China, May 9.

Samsami, R., (2024). A Systematic Review of Automated Construction Inspection and Progress Monitoring (ACIPM): Applications, Challenges, and Future Directions. *CivilEng,* 5: p. 256-287.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D., (2020). LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Las Vegas, NV, USA, October 24.

Smarsly, K., Dragos, K., Stührenberg, J., & Worm, M., (2023). Mobile structural health monitoring based on legged robots. *Infrastructures*, 8: 136.

Trzeciak, M., Pluta, K., Fathy, Y., Alcalde, L., Chee, S., Bromley, A., Briliakis, I., & Alliez, P., (2023). ConSLAM: Construction data set for SLAM. *Journal of Computing in Civil Engineering*, 37(3): 04023009.

Tandon, A., Stührenberg, J., Dragos, K., Mohite, I., & Smarsly, K., 2025a. BIM-based human-robot collaboration for building inspections using mixed reality. In: *Proceedings of the 2025 European Conference on Computing in Construction*. Porto, Portugal, July 14.

Tandon, A., Stührenberg, J., & Smarsly, K., 2025b. Automated building inspections coupling building information modeling and behavior trees. In: *Proceedings of the 2025 European Conference on Computing in Construction*. Porto, Portugal, July 14.

Vega Torres, M.A., Braun, A., & Borrmann, A., (2022). Occupancy grid map to pose graph-based map: Robust BIM-based 2D LiDAR localization for lifelong indoor navigation in changing and dynamic environments. In *Proceedings of the 14th European Conference on Product and Process Modelling*. Trondheim, Norway, September 14.

Vega Torres, M.A., Braun, A., & Borrmann, A., (2024). SLAM2REF: Advancing long-term mapping with 3D LiDAR and reference map integration for precise 6-DoF trajectory estimation and map extension. *Construction Robotics*, 8(2): 13.

Zhang, J. & Singh, S., (2017). Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2): p. 401-416.