2025 European Conference on Computing in Construction
CIB W78 Conference on IT in Construction
Porto, Portugal
July 14-17, 2025

AUTOMATED BUILDING INSPECTIONS COUPLING BUILDING INFORMATION MODELING AND BEHAVIOR TREES

Aditya Tandon, Jan Stührenberg, and Kay Smarsly
Hamburg University of Technology, Hamburg, Germany
Email: aditya.tandon@tuhh.de

Abstract

Automated building inspections using robots are increasingly employed to enhance safety standards and optimize maintenance processes. However, robot-based inspection methods lack adaptability in inspection planning and BIM integration for context-aware navigation. This paper presents a BIM-based inspection framework, leveraging behavior trees for automated planning and execution of robot tasks. The framework extracts spatial and semantic data from BIM models to advance automated inspection. The BIM-based inspection framework is implemented using quadruped robots, and validated by performing inspection tasks in indoor office environments. The results demonstrate that coupling BIM and behavior trees improves accuracy and reliability of automated building inspections.

Introduction

Regular maintenance inspections of buildings are essential for maintaining quality standards that ensure structural safety and integrity (Halder and Afsari, 2023). Building inspections are conventionally performed manually, with trained inspectors conducting visual and structural assessments (Shariq and Hughes, 2020). However, manual inspections are often time-consuming, labor-intensive, and hazardous, while being prone to inconsistent or subjective quality assessments (Tandon et al., 2024). As a result, mobile robots have been proven to be a viable solution for automating maintenance inspections of buildings, offering the potential for safer, more frequent, and objective quality assessments compared to traditional manual inspections (Smarsly et al., 2023), taking advantage of the semantics and geometries provided by building information models.

Building information modeling (BIM) is increasingly adopted for utilizing standardized digital building models throughout the building lifecycle (Borrmann et al., 2018). BIM models contain both semantic and geometric information relevant to robot inspection tasks. Additionally, BIM models may help in localizing robots within buildings, identifying specific building elements to be inspected, planning optimal paths from the current position of robots to the position of building elements, and in facilitating the completion of inspection tasks (Liu et al., 2021).

To advance inspection task planning, semantics and geometries provided by BIM models may be utilized for the execution of robot inspection tasks precisely and efficiently (Odugu et al., 2024). Bahreini et al. (2024) have proposed the OBRNIT ontology for BIM-based robotic navigation and inspection tasks by semantically modeling the relationships between robots, buildings, navigation, and inspection processes. The ontology demonstrates the advantages of utilizing BIM for inspection planning. However, the ontology operates at a conceptual level and does not incorporate robot-specific knowledge. Kim and Peavy (2022) have developed a framework that converts BIM data into the Universal Robot Description Format, to plan sequences of inspections, optimize inspection paths, and update building models based on inspection results. Although the framework successfully demonstrates the feasibility of using BIM for automated robot task planning, the validation is performed solely in simulated settings, which may oversimplify the complexities of the real world. Other approaches rely on closed-world assumptions that restrict the adaptability to dynamic real-world environments. For example, Chen et al. (2022) have proposed a BIM-based system for automated facility inspections using robots. However, real-world validation and the interoperability between BIM and software utilized by robots have rarely been considered, and the adaptability to dynamic environments or inspection re-planning is limited.

In summary, several challenges related to the interoperability between BIM and robots remain unsolved. Moreover, deviations arising from discrepancies between the "as-planned" state reflected in BIM models and the "as-built" state may adversely affect robot task and motion planning when automating building inspections. To improve the adaptability of robots in dynamic environments, task control structures, such as behavior trees, have gained popularity due to their robust, modular, and reusable structure. Behavior trees provide a hierarchical framework that facilitates adaptive and reactive task planning coupled with flexible task execution strategies in uncertain and dynamic environments (Ghzouli et. al., 2023). Recent attempts to enhance the decision-making capabilities of robots with behavior trees have yielded effective results, ensuring inspection mission completion even in unfavorable conditions (Rocamora et. al., 2024). In this paper, a BIM-based inspection framework is proposed that couples BIM and behavior tree (BT) execution control structures to automate building inspections using quadruped robots. BIM models are automatically analyzed to identify and extract building elements requiring inspection, thereby capturing specific requirements and spatial information for inspection. To enable robot navigation using BIM semantics and geometries, fiducial tags are employed to accurately align BIM models with maps devised for robot navigation. Inspection missions are generated as BTs, based on the type of building elements and the capabilities of robots to perform inspections. Inspection mission BTs consist of unique sequences of tasks to be executed by robots based on the characteristics of building elements and a set of generic robot-specific tasks such as navigation between inspection poses.

The remainder of this paper is organized as follows. Next, the design of the BIM-based inspection framework is presented. Thereupon, the implementation and validation of the framework are described, and the validation test results are discussed. Finally, the paper concludes by summarizing the findings and proposing potential future research directions.

A BIM-based inspection framework using behavior tree execution control structures

The objective of the framework is to automate building inspections by coupling BIM models and mobile robots in an accurate and reliable manner. This section presents the design of the BIM-based inspection framework. First, the extraction of building elements from BIM models for the generation of inspection tasks is presented, followed by the alignment of BIM models with maps for robot navigation, and the automated generation of behavior trees from building elements and robot capabilities.

Extraction of building elements from BIM models

As a basis to achieve reliable and accurate inspections in buildings, information about specific building elements to be inspected ("target building elements") is extracted from BIM models and utilized for generating inspection tasks. Essentially, for the extraction of target building elements, the BIM-based inspection framework leverages the hierarchical structure inherent within BIM models, where building elements are organized based on spatial and relational contexts (Theiler and Smarsly, 2018). All instances of target building elements, along with properties, such as dimensions and location information, are extracted by traversing the structure of the BIM models and filtering elements based on the type of the target building elements. Additionally, the BIM-based inspection framework defines inspection poses at predefined distances from the target building elements, ensuring the inspection poses are correctly oriented, to directly face the target building elements for optimal inspection. The inspection poses ${}^{\mathrm{B}}_{\mathrm{I}}\mathbf{P}$ with reference to the BIM model coordinate system, referred to as the BIM model frame **B**, for the *n*-th building element with type T is calculated using:

$$ {}^{\mathrm{B}}_{\mathrm{I}}\mathbf{P}_{\mathrm{T_n}} = \mathbf{H}_{\mathrm{T}}\, {}^{\mathrm{B}}_{\mathrm{E}}\mathbf{P}_{\mathrm{T_n}} \quad \in \mathrm{SE}(3) \qquad (1) $$

where ${}^{\mathrm{B}}_{\mathrm{I}}\mathbf{P}_{\mathrm{T_n}}$ denotes the inspection pose for the *n*-th element of type T in the BIM model frame **B**, $\mathbf{H}_{\mathrm{T}}$ denotes the homogenous transformation matrix containing the fixed distance and orientation for elements of type T, and ${}^{\mathrm{B}}_{\mathrm{E}}\mathbf{P}_{\mathrm{T_n}}$ denotes the building element pose for the *n*-th element of type T in the BIM model frame **B**.

Aligning BIM models with maps for robot navigation using fiducial tags

The target building elements are defined to match the BIM model frame **B**. In traditional robot navigation approaches, maps utilized for localization and path planning are "learned" from sensors mounted on the robots. The maps are defined in a separate coordinate system, referred to as the map frame **M**, which does not coincide with the BIM model frame **B**. Therefore, a transformation ${}^{\mathrm{B}}\mathbf{T}_{\mathrm{M}}$ is needed to align the BIM model frame **B** and the map frame **M**, transforming locations of building elements into the map frame.

The BIM-based inspection framework calculates accurate rigid transformations aligning the BIM model frame and the map frame by merging multiple correspondences of fiducial tags. Fiducial tags are artificial landmarks designed to be easily recognized, which are placed at the same locations in the real building and in the corresponding BIM model. In the BIM-based inspection framework, AprilTags are employed as fiducial tags, i.e. 2D barcode tags with an ID number that can be printed on paper (Olson, 2011). By placing "virtual" AprilTags in the BIM model, poses of the AprilTags can be retrieved with respect to the BIM model frame **B**. Observing AprilTags placed in real buildings with cameras mounted on robots allows calculating transformations from the robots to the AprilTags. Combining the transformations with the current poses of robots in the map yields poses of the AprilTags with respect to the map frame **M**. Since AprilTags comprise ID numbers, corresponding AprilTags in BIM models and in real buildings are identified fast and unambiguously.

While robots traverse buildings and observe AprilTags during navigation, correspondences of real and virtual AprilTags are acquired. When the first AprilTag is observed, the first correspondence is acquired, and the rigid transformation ${}^{\mathrm{B}}\mathbf{T}_{\mathrm{M}}$ that aligns the BIM model frame **B** and the map frame **M** can be calculated as follows:

$$ {}^{\mathrm{B}}\mathbf{T}_{\mathrm{M}} = {}^{\mathrm{B}}\mathbf{T}_{\mathrm{A_i}}\, {}^{\mathrm{M}}_{\mathrm{t}}\mathbf{T}^{-1}_{\mathrm{A_i}} \in \mathrm{SE}(3) \quad \text{with} \quad {}^{\mathrm{M}}_{\mathrm{t}}\mathbf{T}_{\mathrm{A_i}} = {}^{\mathrm{M}}_{\mathrm{t}}\mathbf{T}_{\mathrm{R}}\, {}^{\mathrm{R}}_{\mathrm{t}}\mathbf{T}_{\mathrm{A_i}} \qquad (2) $$

where ${}^{B}\mathbf{T}_{A_i}$ denotes the pose of the AprilTag with ID i in the BIM model frame $\mathbf{B}$ and ${}^{M}_{t}\mathbf{T}_{A_i}$ is the transformation estimate of the AprilTag with ID i to the map frame $\mathbf{M}$ at time t. The transformation ${}^{M}_{t}\mathbf{T}_{A_i}$ is obtained by multiplying ${}^{M}_{t}\mathbf{T}_{R}$, which indicates the pose of the robot in the map frame $\mathbf{M}$ at time t, with ${}^{R}_{t}\mathbf{T}_{A_i}$ describing the transformation from the AprilTag with ID i observed by the camera to the robot frame $\mathbf{R}$ at time t. However, the accuracy of ${}^{M}_{t}\mathbf{T}_{A_i}$, and consequently the accuracy of ${}^{B}\mathbf{T}_{M}$, is affected by measurement noise and the movement of the robot, including vibrations of legged robots when the feet hit the ground, which may result in a misalignment between the BIM model and the map. To obtain an accurate transformation ${}^{B}\mathbf{T}_{M}$, the accuracy of the pose of AprilTags with respect to the map frame $\mathbf{M}$ is assured as follows. Therefore, m detections of AprilTags in subsequent camera images are considered to calculate the mean rigid transformation ${}^{M}\overline{\mathbf{T}}_{A_i}$, and the variance of the m transformations is evaluated based on the root mean square error (RMSE). Low-quality transformations, which may cause misalignments between the BIM model and the map, are identified by the RMSE exceeding the threshold τ_{RMSE} and are rejected for further calculations.

The calculation of ${}^{B}\mathbf{T}_{M}$ using a single AprilTag correspondence may not be sufficient for accurate alignments. Small errors in the rotation of ${}^{B}\mathbf{T}_{M}$ result in larger misalignments with increasing distance. In addition, deviations in the location of AprilTags in the BIM model and the real building, due to inaccurate placement or errors in the BIM model, may lead to misalignment. By leveraging multiple AprilTag correspondences, observed at different locations, the errors are minimized by calculating the mean transformation ${}^{B}\overline{\mathbf{T}}_{M}$ to improve the alignment of the BIM model and the map. The transformation ${}^{B}\overline{\mathbf{T}}_{M}$ is recalculated once a new AprilTag correspondence with a high-quality transformation ${}^{M}\overline{\mathbf{T}}_{A_i}$ is accepted.

Automated generation of behavior trees from building elements and robot capabilities

Reliable and accurate inspection task planning requires structures that are capable of dynamically adapting the execution control flow of robots. In this study, behavior trees are proposed due to their modularity, adaptability, and reusability. Essentially, target building elements are extracted from BIM models to facilitate the creation of inspection tasks. The target building elements are analyzed to extract semantic and geometric properties to determine the type and the location of inspection tasks necessary for building inspection. For every target building element, a unique set of inspection tasks is defined based on functional and operational requirements. The inspection tasks are mapped to specific robot capabilities, ensuring robots are capable of addressing the functional and operational requirements of each building element. Based on the mapping of inspection tasks and robot capabilities, BT subtrees are generated that encapsulate the logic, parameters, and execution sequences required to execute fundamental robot tasks.

In this study, the navigation subtree and the visual inspection subtree are developed as fundamental robot tasks. The navigation subtree, shown in Figure 1, facilitates autonomous planning and execution of navigation paths and recovery in case of navigation failures.

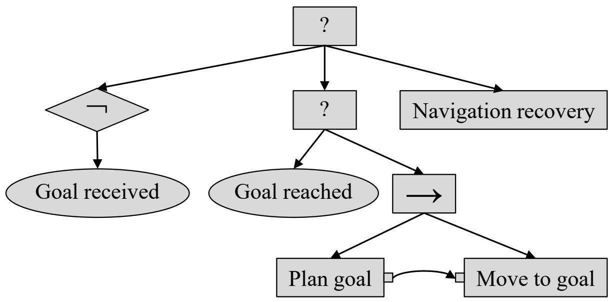

Figure 1: Navigation subtree

The navigation subtree evaluates the *Goal received* and *Goal reached* conditions, before navigation goals are planned and executed using the *Plan goal* and *Move to goal* action nodes. The visual inspection subtree, shown in Figure 2, is designed to perform a range of visual analysis tasks within the operational environment. Upon evaluating the *Start inspection* condition, the subtree activates modules necessary for visual inspection (e.g., camera and object detection) and captures images required for documentation.

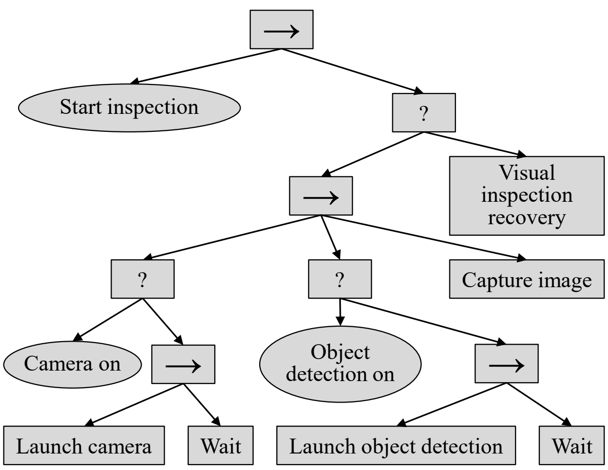

Figure 2: Visual inspection subtree

To create larger sequences of inspection tasks that cater to numerous inspection scenarios, the navigation subtree and the visual inspection subtree are chained in BT execution control structures to enable robots to navigate to inspection waypoints, perform visual inspections, and move to subsequent inspection waypoints, as shown in Figure 3.

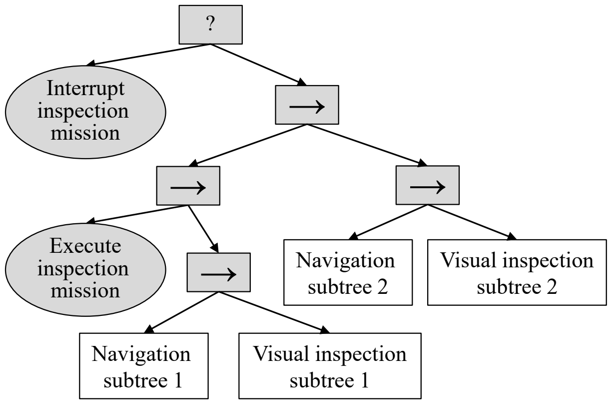

Figure 3: Subtrees chained together in a BT execution control structure

In the following section, the implementation and validation of the BIM-based inspection framework is described, and the results are discussed.

Implementation and validation of the BIM-based inspection framework

This section presents the implementation and validation of the BIM-based inspection framework. First, the implementation of the extraction of building elements from BIM models, the alignment of BIM models with maps for robot navigation, and the BT execution control structures are discussed. Then, tests designed to validate the accuracy and reliability of the BIM-based inspection framework are described. Finally, the results of the validation tests are presented and discussed.

Implementation

To enable the extraction of the building elements from BIM models, the Industry Foundation Classes (IFC) schema is utilized, as the IFC schema provides an open and standardized format for representing and sharing BIM data. Specifically, to implement the BIM-based inspection framework, the open-source library IfcOpenShell (IfcOpenShell, 2024) is employed, facilitating parsing and manipulation of IFC files. The extraction of building elements is performed using the Python API of the IfcOpenShell library by loading an IFC file and retrieving specific entities that have been predefined. Once target building elements are obtained, corresponding pose information for the elements is extracted and used to calculate inspection poses for the robots at pre-defined distances from the elements in the BIM model frame.

To align the BIM model with the maps utilized by the robots, a custom AprilTag family compatible with Autodesk Revit (Autodesk, 2024) is used to place AprilTags in the BIM model. From the BIM model, the tags are exported as *IfcBuildingElementProxy* elements into IFC format and the poses of the AprilTags ${}^{\mathrm{B}}\mathbf{T}_{\mathrm{A_i}}$ are automatically extracted using the IfcOpenShell library. Therefore, in the real building to be inspected, the AprilTags are printed on paper, placed in correspondence to the locations in the BIM model, and the library *apriltag_ros* is used to detect the tags in the real building (AprilRobotics, 2024). The optimization problem of calculating the mean transformations ${}^{\mathrm{M}}\overline{\mathbf{T}}_{\mathrm{A_i}}$ and ${}^{\mathrm{B}}\overline{\mathbf{T}}_{\mathrm{M}}$ is implemented in Python using the *scipy.optimize.least_squares* package (SciPy, 2024).

The BT execution control structures are implemented using Python in the Robot Operating System (ROS) (Quigley, 2009). To generate inspection mission BTs, the *ros_bt_py* framework (Heppner, 2023) is extended to create custom nodes that enable robot behaviors and functionality. Fundamental nodes, such as condition nodes, improve the reactive capabilities of robots, and high-level behavior nodes, such as *Plan goal* (in the navigation subtree, Figure 2) and *Launch camera* (visual inspection subtree, Figure 1), extend the functionality of the framework. The navigation and visual inspection subtrees are implemented through an application programming interface (API) using a pre-defined logical structure and are utilized as a template to adapt the subtree structures for different navigation and inspection tasks. Subtrees, stored as *.yaml* files, are chained together using ROS service calls, provided by the *ros_bt_py* framework. Custom classes that interface directly with core modules of the *ros_bt_py* framework are implemented to construct large inspection plan BTs by augmenting the BT structures of the mission plan BT. For more details on the operation of the motion planning framework used by the robots, the interested reader is referred to Tandon et al. (2024).

Validation

Validation tests are conducted in an indoor office environment with dimensions of 39 m x 16 m and with a corresponding BIM model (Figure 4). The tests are designed to validate the accuracy and reliability of the BIM-based inspection framework.

- The ***accuracy*** is evaluated by measuring the distances between ground truth inspection poses to inspection poses reached by robots. The distances of five inspection poses achieved in three validation tests are assessed by calculating the RMSE.
- The ***reliability*** of the BIM-based inspection framework is evaluated through the capability to perform inspection tasks automatically without human intervention, i.e. (i) navigating to the inspection poses, (ii) executing the BT structures, and (iii) detecting target building elements successfully.

The validation tests are conducted as follows: Electrical outlets are chosen as target building elements, i.e. visual inspections are performed on the electrical outlets. The BIM model of the indoor office environment, exported into IFC format, is searched for the corresponding IFC element *IfcOutlet*. Then, the inspection missions, consisting of navigation and visual inspection tasks, are planned to validate the BIM-based inspection framework. The inspection poses, relevant to the inspection missions, are calculated to lie 1 m in front of each electrical outlet.

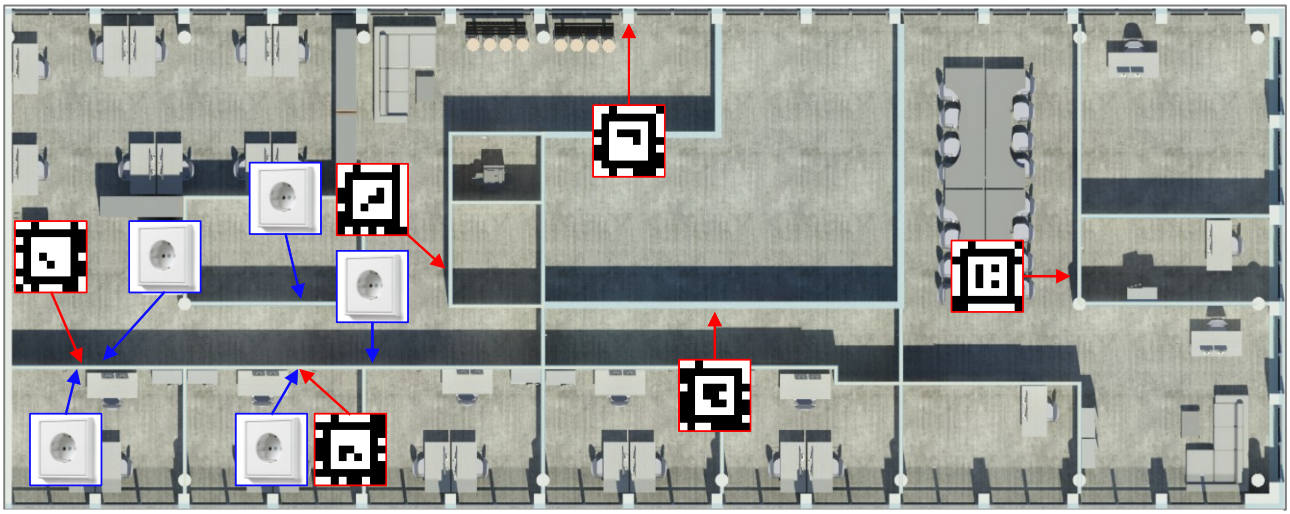

Figure 4: BIM model of the indoor office environment with electrical outlets and AprilTags

To obtain an accurate alignment of the BIM model and the map frame, six AprilTags are strategically placed on the walls of the indoor office environment and in the BIM model at corresponding locations. The placement of the AprilTags and the electrical outlets in the BIM model is shown in Figure 4.

The validation tests are conducted using a quadruped robot, the "Intelligent Documentation Gadget" (IDOG), shown in Figure 5, which has successfully been deployed in previous studies to perform inspections in the context of structural health monitoring (Smarsly et al., 2022) and structural assessment (Johann et al., 2024). In the validation tests, the IDOG employs the motion planning framework described in Tandon et al. (2024). Physical inspection pose markers, shown in Figure 5, are placed at the corresponding test locations, i.e., 1 m in front of the electrical outlets, in the real building. Three inspection missions are executed to test the accuracy of the target element extraction, BIM alignment, and the navigation accuracy of the BIM-based inspection framework. At each inspection pose in each inspection mission, deviations between the planned pose and the real pose of the IDOG are compared.

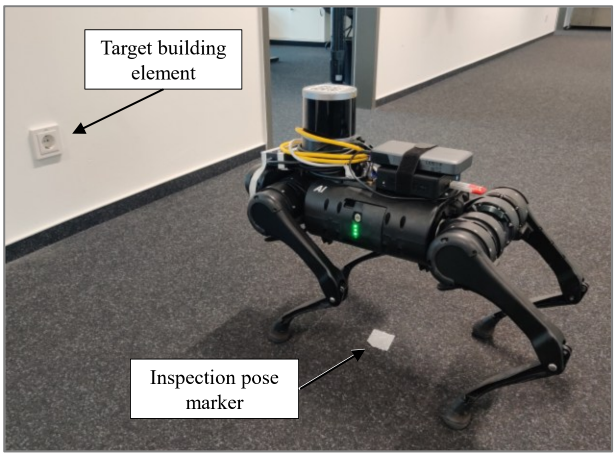

Figure 5: Quadruped robot "IDOG" used in validation tests

As mentioned above, the reliability of the BIM-based inspection framework is evaluated through the navigation, the BT structure execution, and target building element detection capabilities. These metrics are validated upon reaching the inspection poses, when the visual inspection subtrees are executed to detect and document the state of target building elements, i.e., the electrical outlets. The results are summarized in the following subsection.

Results

Figure 6 shows a screenshot of the validation tests. As can be seen, all five inspection poses are generated successfully. The alignment of the BIM model and the map for robot navigation is accomplished with small deviations.

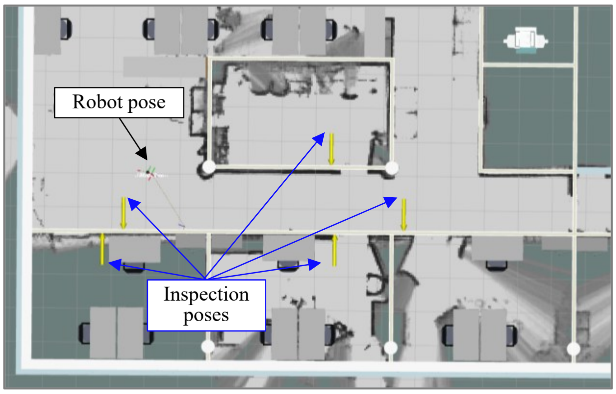

Figure 6: Alignment of the BIM model and the map for robot navigation, including inspection poses

Table 1: Distances between ground truth inspection poses and inspection poses reached by the IDOG in meters

Test Nr.	Pose 1	Pose 2	Pose 3	Pose 4	Pose 5
1	0.18	0.12	0.18	0.14	0.11
2	0.09	0.06	0.16	0.07	0.22
3	0.13	0.06	0.13	0.07	0.07

The results of the validation tests regarding the accuracy are displayed in Table 1. Combined in the RMSE, an error of 0.129 m is observed. The validation tests performed to validate the reliability of the BIM-based inspection framework demonstrate that the IDOG navigates to inspection poses without intervention. Figure 7 shows a screenshot of the IDOG navigating to an inspection pose. Furthermore, it could be proven that the inspection module is capable of reliably execution the BT structure and detecting the target building elements, i.e. the electrical outlets, and successfully documenting the state of the target building element by capturing images: In each of the five inspection poses in each of the three inspection missions, images are taken, timestamped and saved on the IDOG (Figure 8).

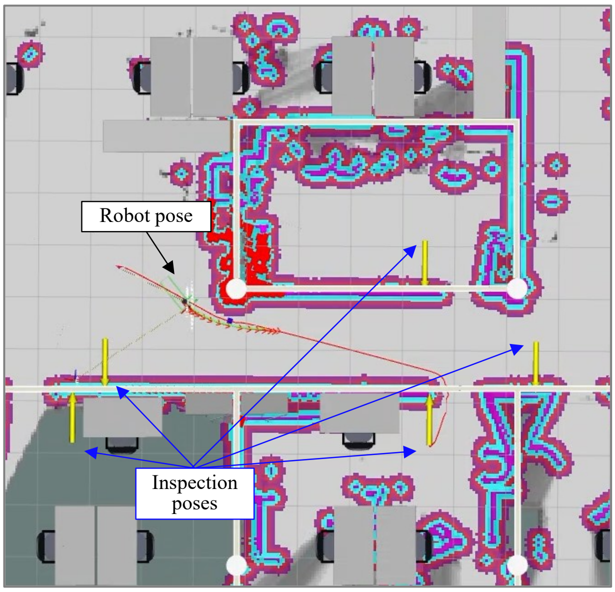

Figure 7: IDOG navigating to inspection pose

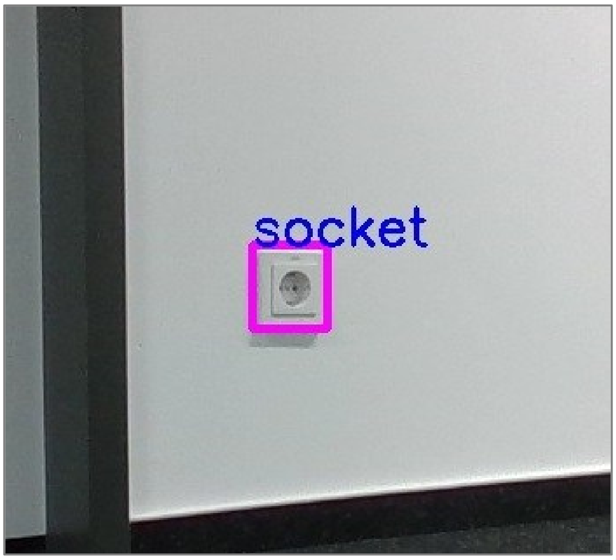

Figure 8: Image of the electrical outlet automatically captured during the visual inspection task

Discussion

Although promising results have been achieved, it should be mentioned that the deviations between ground truth inspection poses and inspection poses reached by the IDOG are influenced by the accuracy of the placement of fiducial tags, target building elements in the BIM model, and the spatial accuracy of the BIM model itself. The deviations between the poses of the real fiducial tags and the virtual fiducial tags placed in the BIM model result in misalignments of the map frame and the BIM model frame. The deviations between the poses of real target building elements and the virtual target building elements in the BIM model result in deviations between ground truth inspection poses and generated inspection poses. Overall, the accuracy, reflected by an RMSE of 0.129 m, is sufficient for visual inspections. Although validation tests have been performed over three inspection missions, each consisting of five inspection poses, further inspection missions with more inspection poses are likely to confirm the accuracy of the framework. Thus, the BIM-based inspection framework has demonstrated reliable performance, as all inspection missions have been executed by the IDOGs without human intervention. In the next section, the paper is summarized, and conclusions are drawn.

Summary and conclusions

BIM models contain valuable semantic and geometric information that may be effectively utilized to enhance automated building inspections using mobile robots. In this paper, the challenges of coupling BIM models with robot navigation have been addressed, using a map alignment strategy based on fiducial tags and the employment of behavior trees, respectively. Specifically, as has been shown in this paper, the BIM-based inspection framework proposed in this paper provides an accurate and reliable foundation for automated building inspections, utilizing information inherent to BIM models to guide robot inspection tasks.

The BIM-based inspection framework has been validated by performing automated visual inspection of electrical outlets using a quadruped robot, deployed in an indoor office environment. The validation tests demonstrate the capability of the framework to enable accurate and reliable building inspections by automatically aligning BIM models with real-world buildings. The extraction of building data from BIM models has enabled context-aware navigation, the integration of fiducial tags has enabled accurate map alignment, and behavior trees have provided a robust mechanism for adapting inspection tasks to real-time conditions despite environmental variations.

In future work, the framework proposed in this study may be extended to integrate different types of building elements and inspection tasks and to enable multiple robots to perform inspection tasks collaboratively. Furthermore, the alignment of BIM models with real-world buildings may be investigated using input provided by proprioceptive sensors only to avoid manual tag placement.

References

AprilRobotics (2024) AprilTag 3. Available online: https://github.com/AprilRobotics/apriltag (accessed 08/09/2024).

Autodesk (2024) Autodesk Revit: BIM software to design and make anything. Available online: https://www.autodesk.com/eu/products/revit/overview (accessed 08/09/2024).

Bahreini, F., Nasrollahi, M., Taher, A., & Hammad, A. (2024) Ontology for BIM-Based Robotic Navigation and Inspection Tasks. Buildings, 14(8), 2274.

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018) Building information modeling: Technology foundations and industry practice. Cham, Switzerland: Springer.

Chen, J., Lu, W., Fu, Y., & Dong, Z. (2023) Automated facility inspection using robotics and BIM: A knowledge-driven approach. Advanced Engineering Informatics, 55, 101838.

Ghzouli, R., Berger, T., Johnsen, E.B., Wasowski, A., & Dragule, S. (2023) Behavior trees and state machines in robotics applications. IEEE Transactions on Software Engineering, 49(9), pp. 4243-4267.

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

Heppner, G., Berg, N., Oberacker, D., Spielbauer, N., Roennau, A., & Dillmann, R. (2023) Distributed behavior trees for heterogeneous robot teams. In: Proceedings of the 2023 IEEE 19th International Conference on Automation Science and Engineering. Auckland, New Zealand, 08/26/2023.

IfcOpenShell (2024) IfcOpenShell: The open source IFC toolkit and geometry engine. Available online: https://ifcopenshell.org/ (accessed 08/09/2024).

Johann, S., Stührenberg, J., Tandon, A., Dragos, K., Bartholmai, M., Strangfeld, C., & Smarsly, K. (2024) Implementation and validation of robot-enabled embedded sensors for structural health monitoring. In: Proceedings of the 11th European Workshop on Structural Health Monitoring. Potsdam, Germany, 06/10/2024.

Kim, K. and Peavy, M. (2022) BIM-based semantic building world modeling for robot task planning and execution in built environments. Automation in Construction, 138, p.104247.

Liu, L., Li, B., Zlatanova, S., & Van Oosterom, P. (2021) Indoor navigation supported by the Industry Foundation Classes (IFC): A survey. Automation in Construction, 121, 103436.

Odugu, O., Ghafari, F., Shourangiz, E., Khan, M.T., & Wang, C. (2024) Building information model (BIM) and robotic systems integration for construction: A comprehensive workflow analysis and future perspectives. In: Proceedings of the 5th International Conference on Artificial Intelligence in Human-Computer Interaction. Washington, DC, USA, 06/29/24.

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, 05/09/2011.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. & Ng, A.Y., (2009) ROS: An open-source robot operating system. In: ICRA workshop on open source software. Kobe, Japan, 05/12/2009.

Rocamora, B.M., Simplicio, P.V., & Pereira, G.A., (2024) A behavior tree approach for battery-aware inspection of large structures using drones. In: Proceedings of the 2024 International Conference on Unmanned Aircraft Systems. Chania, Greece, 06/04/2024.

SciPy (2024) scipy.optimize.least_squares. Available online: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html (accessed 12/09/2024).

Shariq, M.H. & Hughes, B.R. (2020) Revolutionising building inspection techniques to meet large-scale energy demands: A review of the state-of-the-art. Renewable and Sustainable Energy Reviews, 130, 109979.

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

Smarsly, K., Dragos, K., Stührenberg, J., & Worm, M. (2022) Structural health monitoring of civil infrastructure using mobile robots. In: Proceedings of the 19th International Conference on Computing in Civil and Building Engineering. Cape Town, South Africa, 10/26/2022.

Tandon, A., Stührenberg, J., Dragos, K., & Smarsly, K. (2024) Autonomous navigation of quadruped robots for monitoring and inspection of civil infrastructure. In: Proceedings of the 20th International Conference on Computing in Civil and Building Engineering. Montréal, Canada, 08/25/2024.

Theiler, M. & Smarsly, K. (2018) IFC Monitor – An IFC extension for modeling structural health monitoring systems. Advanced Engineering Informatics, 37, pp. 54-65.