

Semantic Optimal Robot Navigation Using Building Information on Construction Sites

Sina Karimi¹, Rafael Gomes Braga², Ivanka Iordanova¹ and David St-Onge²

¹Construction Engineering Department, École de Technologie Supérieure, Canada

²Mechanical Engineering Department, École de Technologie Supérieure, Canada

E-mail: sina.karimi.1@ens.etsmtl.ca, rafael.gomes-braga.1@ens.etsmtl.ca,
ivanka.iordanova@etsmtl.ca, david.st-onge@etsmtl.ca

Abstract -

With the growth in automated data collection of construction projects, the need for semantic navigation of mobile robots is increasing. In this paper, we propose an infrastructure to leverage building-related information for smarter, safer and more precise robot navigation during construction phase. Our use of Building Information Models (BIM) in robot navigation is twofold: (1) the intuitive semantic information enables non-experts to deploy robots and (2) the semantic data exposed to the navigation system allows optimal path planning (not necessarily the shortest one). Our Building Information Robotic System (BIRS) uses Industry Foundation Classes (IFC) as the interoperable data format between BIM and the Robotic Operating System (ROS). BIRS generates topological and metric maps from BIM for ROS usage. An optimal path planner, integrating critical components for construction assessment is proposed using a cascade strategy (global versus local). The results are validated through series of experiments in construction sites.

Keywords -

BIM/IFC, Semantic Navigation, Autonomous Robot, Path Planning

1 Introduction

Conventional methods of data collection for the purpose of progress monitoring rely on periodic observations, manual data collection (which is mostly textual data and a limited number of photos), and personal interpretation of the project progress [1]. These aforementioned conventions are error-prone, time-consuming and cost-ineffective since they are subjective processes [2]. Manual data acquisition by individuals would result in decentralized data; coming from different sources in different formats, thereby making it somewhat challenging to manage and analyze them. Automation of monotonous and repetitive construction processes would significantly enhance construction efficiency [3]. Hence, there is a growing need in the construction industry to automate data collection task. In addition, the applications of data collection using an Unmanned Ground Vehicle (UGV) can provide new kinds of

information and applications such as equipment tracking and 3D reconstruction which would ultimately have positive impacts on quality control, safety and sustainability of the construction projects.

With tremendous progress in mobile robots capabilities, the interest in adopting mobile robots for data collection on construction sites is increasing. Rugged platforms with high manoeuvrability are commercialized for this usage [4] and several works are enhancing their autonomy for navigating these challenging environments [5]. A handful of fundamental steps still need to be addressed for the deployment of robots on construction sites, such as their usage by non-experts (untrained) operators and the automatic integration of the diverse requirements related to construction management in their mission planning. Our solution leverages BIM semantics extracted in an interoperable data schema, IFC, and translated for robot indoor navigation. This semantic information, intertwined with the robot navigation and mission, help the operator manage the robotic system as they share conceptual knowledge of their environment.

This paper proposes a novel method for semantic robot navigation with an optimal path planning algorithm using building knowledge on construction sites. The optimal path is extracted from user inputs using BIM/IFC which provide digital representation of the construction project [6]. The resulting path (which is not necessarily the shortest path) can be altered with the weights of several criteria such as robot and workers safety, BIM new information requirement and sensors sensitivity to environmental features. In this step, the building semantics play an essential role on defining the start, the end and the transitional coordinates with which the robotic system plans the path. Furthermore, all along the mission, the local paths are computed based on the relevant complementary information for the low-level navigation extracted from IFC. This is essential to cope with limitations of the robot. For instance, a path planner should avoid trajectory near glass walls: they are hard to detect by many sensors. Luckily, information about wall materials can be retrieved from BIM. Among the conventional methods on path planning

[7], we use topological map representation in order to store the building semantics in nodes and graphs. The current paper contributions are as follows:

- An optimal high-level path planner integrated with the low-level navigation (cascade navigation stack);
- Semantic teleoperation and navigation for autonomous UGV during the construction process;
- Practical implementation of the proposed system deployed on an autonomous mobile robot navigating a construction site.

2 Related Work

Conventional methods of indoor path planning often refer to optimal path as the shortest path calculated by various algorithms such as A* and Dijkstra's [8]. To enhance the performance of these planners, many studies suggested ways to leverage BIM/IFC for indoor path planning. Wang et al. [9] develop a framework for converting the BIM digital environment to a cell-based infrastructure to support indoor path planning. In this work, they emphasize on the "*BIM voxelization*" process rather than the path planning problem. In another study, a BIM-based path planning strategy is used for equipment travel on construction sites [10]. The authors extract the start and end points from BIM and then generate the shortest sequence of rooms for the operator, but does not support robot path planning. Ibrahim et al. [11] propose a path planning strategy based on BIM for an Unmanned Aerial Vehicle (UAV) on construction sites which uses a camera for data capturing. They use BIM geometries to define a path for outdoor environments but do not address indoor semantic robot path planning. In this direction, Follini et al. [12] utilize BIM geometries for path planning of an UGV supporting construction logistics application. Their proposed system uses a human-assisted approach in a controlled environment and is yet to thoroughly leverage BIM/IFC semantics in a construction site. In [13], the optimal route for a data collection mission using an UAV is proposed. They utilize 4D BIM to identify which building spaces are expected to change during the construction phase (implemented in a simulated environment) so that the flight path navigate through those areas and collect data.

Delbrügger et al. [14] developed a framework supporting humans and autonomous robots navigation which mostly uses building geometries in a simulated environment. In [15], the indoor localization of an UAV is assessed using AprilTags with their known location in a BIM-generated map. They present this work as a proof-of-concept for the use of AprilTags in indoor environment. However, due to inaccuracy of localization in their work, they improve their previous work by using Extended

Kalman Filter (EKF) in their localization framework [16]. Another study examined the use of BIM in robot localization in which the proposed system uses a hierarchical reasoning for path planning [17]. BIM was also demonstrated to be powerful for the identification of different paths from which a hierarchical refinement process can find the shortest path [18]. That work provides only high-level path (rooms sequence) with respect to BIM geometries and the integration with ROS is not studied. An approach using hypergraphs generated from IFC files was also developed in which a modified A* algorithm is able to detect the optimal path among nodes in the graph [8].

In these inspiring works three aspects of the BIM potential for indoor robot path planning are yet to be thoroughly studied: (1) considering the full potential of the BIM/IFC semantic rather than only the geometry (2) integrating the high-level (rooms sequence) with the low-level sensor-based information in a full navigation stack (3) the field validation of strategy using BIM/IFC for both global and local path planning. In this paper, we cover these gaps by integrating Building Information Robotic System (BIRS) into a navigation system in ROS in order to determine the optimal path and then navigate autonomously.

3 Topological building maps created from BIM/IFC

IFC data schema provides construction stakeholders with semantic information of buildings containing attributes and relationships between different entities [19]. This information can be extracted in graph database [20]. However, the use of that information for reasoning is complex since the IFC files encompass large amounts of data. In order to cope with this, we first identify the required data for robot navigation on construction sites, then, we extract and store the data in an XML database. The conceptual semantic relation between BIM/IFC and robot navigation is covered in a previous paper on BIRS [21]. We extend the hypergraphs of Palacz et al. [8] with the semantic and geometric information of IFC files. All the semantic information required to the global and local planners retrieved from IFC is in the form of a topological map.

As IEEE 1873-2015 [22] defines, nodes and edges are the components of topological maps and we fill them with the following information:

- Nodes contain the rooms information namely: room's name, room's unique ID, room center, room area, walls' unique IDs, wall material, last scan date, construction activity (hazard for the robot)
- Edges contain the doors information namely: door's unique ID, door's location, doors opening direction

In the hypergraph, one node is created per *IfcSpace* and

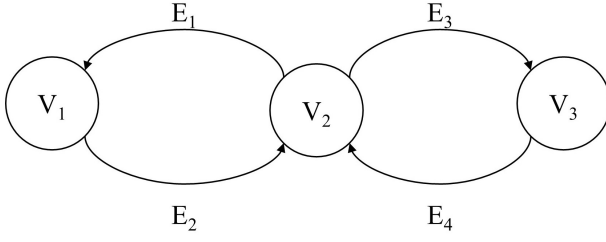


Figure 1. A directed hypergraph of $S = (V, E)$ where $V = \{V_1, V_2, \dots, V_n\}$ is a set of nodes and $E = \{E_1, E_2, \dots, E_m\}$ is a set of hyperedges. Each node (V_i) is an *IfcSpace* containing its relationships and each hyperedge (E_j) is an *IfcDoor* with its attributes extracted by BIRS [21].

for each *IfcSpace*, the bounding *IfcWall* and *IfcCurtain-Wall* elements are identified. With the above-mentioned information, a graph is generated as illustrated in fig. 1. Then, the edges need to be attributed with the cost (weight) of passing over each (from a room to another). In this direction, $W = (W_V, W_E)$ is a pair of weights where W_V and W_E are the node and hyperedge weights respectively. W_{V_i} is the i node total weight obtained from:

$$W_V = w_m + w_a + w_s + w_h \quad (1)$$

where w_m depends on the walls material, w_a , on the room area, w_s , on the room scan-age, and w_h , on the room hazards. W_{E_j} is the j hyperedge weight obtained from:

$$W_E = w_d \quad (2)$$

where w_d depends on the door opening direction. For passing from one node to the other, there might be several paths the robot can use. The overall weight of a path (from start to end node) is as follows:

$$W = \sum_{i=1}^n W_{V_i} + \sum_{j=1}^m W_{E_j} \quad (3)$$

One challenge for the robot is to be able to detect obstacles. To help the robot predict and avoid potential failures, the material properties of the walls are extracted through *IfcMaterial* and its super-type *IfcProduct*. The weight of each curtain wall, i.e. walls that are *invisible* by design, in each node is $w_m = 12$, while all others are $w_m = 4$ since they can be easily detected. The time required to go through a transition node is also taken into account, i.e. bigger rooms take more time for the robot to cross. Accordingly, the weight for the rooms less than $50m^2$, between $50m^2$ to $100m^2$ and more than $100m^2$ are $w_a = 2$, $w_a = 8$ and $w_a = 12$ respectively. Since one of the core purpose of deploying robots on construction sites is to collect data, the scanning age of all rooms is incorporated. The progress monitoring needs up-to-date data and when the robot is collecting data it can optimise its path to visit more rooms

and collect more data. The scanning periods are selected according to industry needs, therefore, we assign $w_s = 10$, $w_s = 6$, $w_s = 0$ for the scanning period of less than 1 week, between 1 week and 2 weeks, and more than 2 weeks respectively. Since the construction projects evolve constantly, the safety aspects of robot navigation are essential. In this direction, the data collection for the spaces with ongoing construction activities should be postponed to a safer moment for the robot to navigate those rooms. If the hazardous space is one of the transition nodes, an alternative route needs to be automatically planned so we assign $w_h = 500$ for the weight of passing through such spaces. In this case, another path will be selected by the algorithm if there is any. If there is not an alternative safe path for the robot, the algorithms provides a warning for high-weight paths so that the supervisor of the robotic deployment is warned. The hypergraph representing building topological map enables the robotic system to find the optimal path by running an algorithm. In this paper, we use directed hypergraph (with directed hyperedges) allowing us to assign cost for door opening directions. *IfcDoor* as a sub-class of *IfcBuildingElement* provides the center coordinates of the doors creating hyperedges (with their coordinates) in the hypergraph. *IfcDoor* also stores the opening direction through y-axis of *ObjectPlacement* parameter. For pushing and pulling the door, we assign $w_d = 2$ and $w_d = 6$ to the hyperedge's weight respectively. This is due to difficulty for pushing and pulling the doors respectively. Ultimately, the total weight of passing one to the other is the sum of nodes weights and edges.

4 Finding The Optimal Indoor Path

As Gallo et al. [23] define, directed hypergraphs are divided into two categories according to their hyperedges namely: forward hypergraph (F-hypergraph) and backward hypergraph (B-hypergraph). The former is a directed hypergraph in which one node diverges to several nodes and the latter is the one in which several nodes converge to one node. As an example of applications, F-hypergraphs are employed for time analysis on transportation networks [24]. Also, B-hypergraphs are used to perform deductive analysis to find the optimal path in a hypergraph. The combination of B-hypergraph and F-hypergraph is a BF-hypergraph having both divergent and convergent nodes [23]. In topological building layouts, we deal with BF-hypergraphs since we have spaces which connect several spaces to other spaces (an example of such nodes is corridors). In addition, we intend to find the optimal path (a "*deductive database analysis*" from several possible paths) based on several criteria which are represented as weights in the hypergraph, therefore, we use the "*Shortest Sum B-Tree*" algorithm which finds a hypertree (subhypergraph) of the nodes as explained in [23]. We also use *additive*

<pre> Inputs: layout_graph : hypergraph tail_room, head_room : node door : hyperedge path_weight : hyperedge_total_weight Outputs: semantic_path : list<node, hyperedge> x_y_path : list<nodes_coordinates, hyperedges_coordinates> hyperedge_total_weight : number </pre>
--

Figure 2. Data structure for IFC-based semantic optimal path planner algorithm

weighting function to calculate the cumulative weight of each possible route and then we choose the lighter route which is the optimal path for the robot.

In order to create the hypergraph, we first retrieve all the relevant IFC information. The process is done with a Dynamo script (a visual programming tool) to extract the IFC parameters in order to export the IFC information in a XML database. A Python script is developed to parse the XML data in order to translate meaningful data to ROS (for example, the rooms center coordinates are retrieved as strings so they need to be parsed to be integrated with the robotic system). With an hypergraph of the whole building, the user defines the start and end nodes (rooms), and let the algorithm find the optimal path. Since we are implementing BF-hypergraph, each pair of nodes is connected with two directed hyperedges together, thereby making a comprehensive B-hypergraph within the BF-hypergraph. This practice allows considering forward and backward direction in a path so that the door opening direction is considered. "*Shortest Sum B-Tree*" algorithm provides the possible hyperedges from a start node to other nodes [23]. Then, the retrieved information is used to create a sub-hypergraph from the start node to all other nodes representing all the possible paths. By giving the destination node to the sub-hypergraph, the possible paths from start to end node are identified and finally the lowest cumulative weight of the paths is retrieved. Having a set of nodes and hyperedges from the optimal path, the building information is extracted to enable semantic navigation. Each node is represented by the name of the corresponding space and the center coordinates of that room. As illustrated in fig. 2, the optimal path outputs a set room names, their coordinates and a set of door coordinates in the sequence of node location and hyperedge (door) location. The room names enable semantic navigation and the 2-D coordinates provides destinations one after the other.

5 Semantic Graphical User Interface

A Graphical User Interface (GUI) was developed based on BIM semantics to allow users to intuitively operate the robot and configure the path planner. The GUI connects to the ROS running in the robot and presents semantic information of the building and data from the robot in real time. The integrated high-level and low-level navigation system moves the robot to the destination. The GUI allows the non-expert users to work with their domain knowledge, thereby making robot deployment more intuitive and simpler. Figure 3 illustrates the interface window. The GUI is developed in Python notebooks, allowing for easy integration of visualization widgets and customization.

The GUI provides the building's rooms in a drop-down list, from which the user selects a destination and then launch the path planner to find the optimal path. The center area of the GUI shows a map of the building, with the robot's pose being updated in real time, along with the paths objectives. The left panel shows the selected room's (end node) attributes. The right panel allows the user to alter the weights of each parameters of the path planner. After changing and saving the new weights, the user can generate the path again and see the results on the map. Finally, the user can click on the *Move Robot* button to trigger the robot to start moving.

6 Field deployment

Our approach was validated from simulation to the field with an experimental case study. The goal was to drive a mobile robot through the corridors of one of the buildings at the École de technologie supérieure, for which an as-planned BIM was available, and collect data. The semantic path planner was used to generate a set of waypoints from the user inputs, then a low level A* path planner aided by a collision detection stack navigates the robot.

Our robotic platform, shown in fig. 4, is built from a four-wheeled UGV (Clearpath Jackal) equipped with wheel encoders, an internal IMU and an onboard NVidia Xavier computer. The Jackal is delivered with ROS nodes for control, odometry estimation (from encoders and IMU) as well as diagnostics tools provided by ROS.

The sensing system, which was envisioned for point cloud collection in construction sites, contains two Li-DARs, five depth cameras and one tracking camera. The sensors are positioned in different directions to cover as much as possible of the robot's surroundings. While all sensors collect and record data of the environment, most of them are also used by the navigation stack for localization and collision avoidance. Below we present a detailed description of each sensor or group of sensors:

- **Front facing cameras:** One Intel Realsense D435i depth camera and one Intel Realsense T265 tracking



Figure 3. Semantic GUI for the intuitive operation of the robot navigation on construction sites. The controls in the header allow selecting a destination and generating the path. The panel to the left shows the attributes of the selected room. The center contains a map of the environment, with the robot's pose in real time represented by the purple arrow. The center points of the rooms and doors in the path are represented in the map by the yellow circles. The right panel allows the user to reconfigure the different weights applied to the path generation.

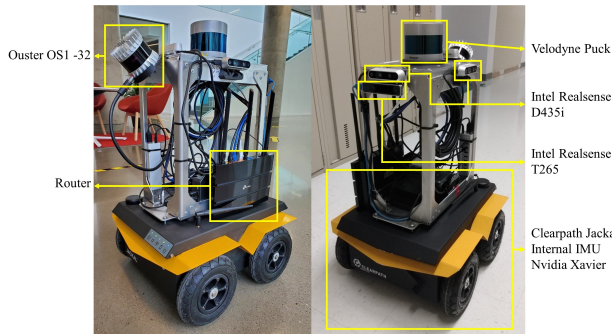


Figure 4. Mobile robot platform equipped with various sensors

camera are mounted in front of the robot. The T265 software estimates the camera's pose and integrates data from the base odometry (wheel encoders and IMU), providing accurate odometry that is fed to the localization algorithm. The D435i provides depth images that are used to detect obstacles immediately in front of the robot, triggering an emergency stop;

- **Velodyne Puck 32MR LiDAR:** Mounted horizontally on top of the robot, it captures laser scan data from all around the robot. This information is used by the localization algorithm to estimate the robot's global position on the building map;
- **Depth cameras:** Three Intel Realsense D435i depth cameras are mounted pointing to the top and left and right sides of the robot. Their purpose is to collect RGB images and depth images from the walls around the robot and from the ceiling;
- **Ouster OS1 LiDAR:** The last sensor, an Ouster OS1 LiDAR is mounted in the back of the robot, inclined

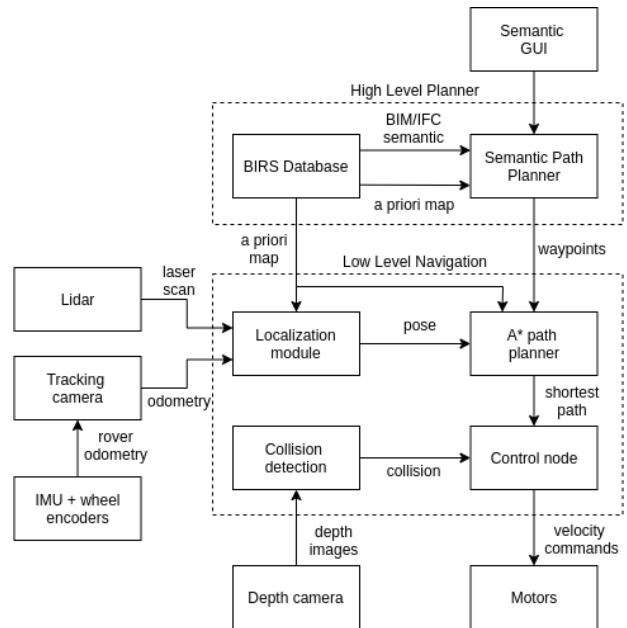


Figure 5. System Overview: A high level planner that process BIM/IFC information and user inputs is integrated to a low level navigation stack in a cascade design. The low-level module takes care of the localization, local path planning and collision avoidance tasks, while the high-level planner generates paths based on BIM/IFC semantics.

by an angle of 45 degrees in order to capture point clouds of the ceiling. Since this sensor has a large 90° field of view, it is also able to cover the walls and part of the back of the robot.

Figure 5 gives an overview of the system. The robot pose in the map is obtained through the use of a ROS

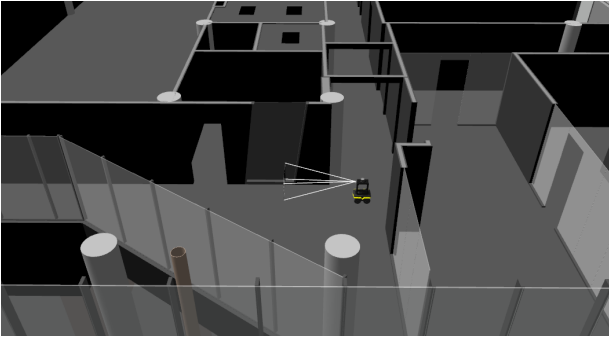


Figure 6. View of the simulated environment used to test the BIM/IFC optimal path planning approach. The building 3D model was built with geometry information extracted from the BIM. The robot model simulates the sensors and possesses the same characteristics as the real robot.

implementation of the Adaptive Monte Carlo localization algorithm[25][26]. Before deploying the robot, wall geometry is extracted from BIM to generate an occupancy grid of the building. During the robot navigation, this map, the odometry, and the laser scan data from the horizontally mounted Velodyne LiDAR are fed to the localization algorithm, which then estimates the robot's current pose in that map. When a destination room is selected, the semantic path planner outputs the preferred path to that room as a list of waypoints, containing the center points of each room, door and corridor in the path. An A* path planner[27] then calculates the shortest path from the robot's current position to the next waypoint in the list. Velocity commands are generated from the A* path and sent to the robot's internal controller to drive it through that path.

The simulation was performed using the Gazebo Simulator. The building information is exported to create a 3D model, a digital twin. Clearpath, Gazebo and the ROS community provide all the required software packages required to generate an accurate simulation of our robotic platform. Figure 6 shows the simulated robot and its environment with different wall textures and transparency.

7 Results

The experiment had two main objectives:

1. Test the effectiveness of the semantic path planner in generating the optimal path to reach the destination, given the building information obtained from BIM/IFC.
2. Test how changes in the building information affect the final path that is generated.

In our case study, the robot starts in a corridor (CORRIDOR OUEST) on the west side of the building and must reach an open area (CORRIDIR EST) on the eastern part

of the building. Figure 7 shows the building map, and the path in red line generated by applying the A* algorithm from start to end. This is the shortest possible path between the two points, taking into consideration only the building geometry and a small safety collision radius around the robot. When the Semantic Path Planner is applied to the same scenario, a similar result is obtained as expected, represented by the blue path in fig. 7. Since there are no doors, undesirable materials or hazards in the path, the algorithm outputs a list of rooms that must be visited by the robot that represent the shortest distance from start to end. The semantic path planner provided the order of rooms' names from the start to the end as it shows in the GUI in fig. 3. Therefore, the user operating the robot can intuitively track the path from the data collected. In this direction, the as-built data can be directly compared to the as-planned since the path is recorded semantically. Also, the waypoints of rooms' center coordinates and doors' center coordinates are provided by the semantic path planner. If there is a door made of materials invisible to sensors (such as glass), the complementary door coordinates help for safer, smarter, precise data collection. Following this, the A* algorithm finds the shortest path between the waypoints.

In a second run, the building information was altered to include a construction operation carried out in the area highlighted with a dashed box in fig. 7 (not visible in the GUI). Since the construction activity represents a hazard with a high cost for the Semantic Path Planner, a different path passing through another corridor is automatically selected, as illustrated by the orange path. Nevertheless, the high cost of the shortest path triggered a warning in the system indicating a hazard to the user through the semantic GUI. Therefore, the user can understand the risks associated with navigation through an active construction area and decide whether to scan the environment or postpone it to a safer time. The orange path was automatically generated, although it is not the shortest path, as the optimal path from the default parameters mentioned in section 4. This path passes along a large curtain wall invisible to the robot's sensors. The additional semantic information provided by the BIRS is given to the robot as well as the BIM occupancy grid so it contributes to collision avoidance with the wall. The GUI provides the user with the scan aging of the rooms so the user can decide which rooms to select as the destination for data collection. This allows the users to run multiple data collection missions with the robot which increases the efficiency of robot deployment on construction sites. As illustrated in fig. 7, the integrated BIM-ROS information provides a cascade navigation system on construction sites enabling autonomous and accurate data collection of the spaces scanned.

- [9] Qiankun Wang, Weiwei Zuo, Zeng Guo, Qianyao Li, Tingting Mei, and Shi Qiao. Bim voxelization method supporting cell-based creation of a path-planning environment. *Journal of Construction Engineering and Management*, 146(7):04020080, 2020.
- [10] Siyuan Song and Eric Marks. Construction site path planning optimization through bim. In *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*, pages 369–376. American Society of Civil Engineers Reston, VA, 2019.
- [11] Amir Ibrahim, Dominic Roberts, Mani Golparvar-Fard, and Timothy Bretl. An interactive model-driven path planning and data capture system for camera-equipped aerial robots on construction sites. In *Computing in Civil Engineering 2017*, pages 117–124. 2017.
- [12] Camilla Follini, Michael Terzer, Carmen Marcher, Andrea Giusti, and Dominik Tobias Matt. Combining the robot operating system with building information modeling for robotic applications in construction logistics. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 245–253. Springer, 2020.
- [13] Amir Ibrahim and Mani Golparvar-Fard. 4d bim based optimal flight planning for construction monitoring applications using camera-equipped uavs. In *Computing in Civil Engineering 2019: Data, Sensing, and Analytics*, pages 217–224. American Society of Civil Engineers Reston, VA, 2019.
- [14] Tim Delbrügger, Lisa Theresa Lenz, Daniel Losch, and Jürgen Roßmann. A navigation framework for digital twins of factories based on building information modeling. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2017.
- [15] Mohammad Nahangi, Adam Heins, Brenda McCabe, and Angela Schoellig. Automated localization of uavs in gps-denied indoor construction environments using fiducial markers. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 35, pages 1–7. IAARC Publications, 2018.
- [16] Navid Kayhani, Adam Heins, Wenda Zhao, Mohammad Nahangi, Brenda McCabe, and Angela P Schoellig. Improved tag-based indoor localization of uavs using extended kalman filter. In *Proceedings of the ISARC. International Symposium on Automation and Robotics in Construction, Banff, AB, Canada*, pages 21–24, 2019.
- [17] Barbara Siemiatkowska, Bogdan Harasymowicz-Boggio, Maciej Przybylski, Monika Różańska-Walczyk, Mateusz Wiśniowski, and Michał Kowalski. Bim based indoor navigation system of hermes mobile robot. In *Romansy 19–Robot Design, Dynamics and Control*, pages 375–382. Springer, 2013.
- [18] Ahmed Hamieh, Aicha Ben Makhlof, Borhen Louhichi, and Dominique Deneux. A bim-based method to plan indoor paths. *Automation in Construction*, 113:103120, 2020.
- [19] Ali Ismail, Barbara Strug, and Grażyna Ślusarczyk. Building knowledge extraction from bim/ifc data for analysis in graph databases. In *International Conference on Artificial Intelligence and Soft Computing*, pages 652–664. Springer, 2018.
- [20] Barbara Strug and Grażyna Ślusarczyk. Reasoning about accessibility for disabled using building graph models based on bim/ifc. *Visualization in Engineering*, 5(1):1–12, 2017.
- [21] Sina Karimi, Ivanka Iordanova, and David St-Onge. An ontology-based approach to data exchanges for robot navigation on construction sites. *arXiv:2104.10239*, pages 1–21, 2021.
- [22] Ieee standard for robot map data representation for navigation. *1873-2015 IEEE Standard for Robot Map Data Representation for Navigation*, pages 1–54, 2015. doi:10.1109/IEEESTD.2015.7300355.
- [23] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2-3): 177–201, 1993.
- [24] A Arun Prakash and Karthik K Srinivasan. Finding the most reliable strategy on stochastic and time-dependent transportation networks: A hypergraph based formulation. *Networks and Spatial Economics*, 17(3):809–840, 2017.
- [25] amcl. <http://wiki.ros.org/amcl>. Accessed: 2021-02-24.
- [26] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.
- [27] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.