2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Connecting Semantic Building Information Models and Robotics: An application to 2D LiDAR-based localization

R.W.M. Hendrikx[1], P. Pauwels[2], E. Torta[1], H.P.J. Bruyninckx[3] and M.J.G. van de Molengraft[1]

Abstract— **This paper proposes a method to integrate the rich semantic data-set provided by Building Information Modeling (BIM) with robotics world models, taking as use case indoor semantic localization in a large university building. We convert a subset of semantic entities with associated geometry present in BIM models and represented in the Industry Foundation Classes (IFC) data format to a robot-specific world model representation. This representation is then stored in a spatial database from which the robot can query semantic objects in its immediate surroundings. The contribution of this work is that, from this query, the robot's feature detectors are configured and used to make explicit data associations with semantic structural objects from the BIM model that are located near the robot's current position. A graph-based approach is then used to localize the robot, incorporating the explicit map-feature associations for localization. We show that this explainable model-based approach allows a robot equipped with a 2D LiDAR and odometry to track its pose in a large indoor environment for which a BIM model is available.**

I. INTRODUCTION

Mobile robots are deployed more and more in dynamic environments shared with and familiar to humans such as hospitals [1], restaurants [2] or nursing homes [3]. Due to the familiarity of the environment, there is an expectation that mobile robots will not only robustly perform long-term autonomous tasks, but that they do so using the same semantics that operators, bystanders and engineers use. To this end, adding semantics to existing geometric representations used for localization is an extensively researched topic. We present a case study for leveraging an existing standard and available models for semantic building representation from the built environment domain for indoor localization. These models, when available, can provide an alternative to robot-specific map creation. In this study, we use a robot equipped with a conventional 2D laser range finder, which is still to be found on many existing platforms. We show how semantic building information models can be used for explainable 2D LiDAR based localization in environments where other sensor modalities may be infeasible due to cost or privacy-by-design requirements.

[1]Bob Hendrikx, Elena Torta and René van de Molengraft are with the Faculty of Mechanical Engineering, Eindhoven University of Technology, The Netherlands {r.w.m.hendrikx,e.torta,m.j.g.v.d.molengraft} at tue.nl

[2]Pieter Pauwels is with the Faculty of Built Environment, Eindhoven University of Technology, The Netherlands p.pauwels at tue.nl

[3]Herman Bruyninckx is with the Faculty of Mechanical Engineering, Eindhoven University of Technology, The Netherlands and with the Faculty of Mechanical Engineering, KU Leuven, Belgium h.p.j.bruyninckx at tue.nl

A. Contribution and contents

In our work, we identify the following contributions:

- Proposing a workflow to leverage the semantic and geometric information in building models for indoor robot localization, via composition with a robot-specific property-graph representation in a spatial database.
- Demonstrate the feasibility of the approach for a location tracking task based exclusively on static semantic building feature associations.

The paper starts by discussing related work from both the robotics domain and the built environment domain in the next section. Hereafter, we introduce the property graph approach for world representation, followed by the step of populating this world model with BIM entities. This is followed by a concise treatment of our localization approach, which uses existing graph optimization techniques to obtain the robot pose from semantic data associations. In section V, an indoor experiment is then presented in which a robot localizes indoor based on a BIM model. Finally, the technical challenges that arise when using existing building models for localization are discussed, and topics for future work are identified.

II. RELATED WORK

Semantic indoor maps for robotics

Among the various representations for indoor semantic maps used in practice (see, e.g., [4] and [5] for a review), we focus on object-oriented representation standards that have been used for robot localization. OpenStreetMap (OSM) is an opensource crowd-sourced mapping initiative that has been extended for indoor robot navigation in [6]. The authors propose an indoor tagging schema to represent the indoor domain hierarchically using the OSM primitives. Objects such as walls and doors are tagged, and related to rooms and hallways. Furthermore, specific areas are represented and tagged to define, e.g., traffic lanes for navigation. The authors show an example of a hospital environment that was modeled and tagged manually using architectural drawings as a reference. The authors acknowledge that the manual creation of the maps using the available Geographic Information Systems (GIS) tools is tedious and that the choice of georeferenced (latitude/longitude) coordinates is not obvious for indoor geometry. Another important initiative for building modeling is indoorGML [7], which specifically targets indoor spaces and represents geometry, topology and semantics in a multi-layered space model. While indoorGML provides many relevant modeling concepts for indoor navigation, existing work is limited to automatic extraction of indoorGML models from occupancy grid maps in [8]. A different representation of semantic maps in spatial databases is suggested in [9] in the form of the SEMAP framework. In their work, the authors represent semantically annotated 3D object models in a PostGIS database. The models are geometrically represented in PostGIS' own 3D extension of the "Simple Feature Access" [10] specification with the addition of a spatial ontology for map queries. The authors show multiple applications, including topological location queries of the map for an agricultural harvesting scenario. The authors do not focus on metric robot localization from on-board sensor data. Another work that has similarities to ours is [11], in which the authors perform robust indoor localization with a laser scanner starting from architectural floor plans. They augment the floor plans using pose graph SLAM techniques and robust matching criteria. This makes their approach able to perform long-term navigation in the presence of map mismatches and environmental disturbances. Contrary to the approach we present, the authors do not focus on semantics or standardized map formats in their work. Furthermore, they propose a method to provide an updated scan-based map that is consistent with the prior floor plan and contains all geometry, which is not within the scope of our work.

In the above works, there is a strong reliance on maps with a geospatial background (e.g. GML, GIS), or architectural floor plans. These sources are not widely available, nor fully reliable. Yet, for many buildings, detailed Building Information Models (BIM) ([12], [13]) are currently available, which include 3D geometry as well as detailed semantic data (materials, object properties, etc.). In this work, we want to take advantage of the increasing availability of building information models as *digital twins* to propose an alternative approach to semantic map creation and representation for the robotics domain. We do so by creating an explicit link between the semantic information contained in Building Information Models (BIM) and robotics semantic maps. Our approach allows to automatically populate a semantic map, avoiding the manual effort cited by [6]. In addition, the robot's semantic map and the BIM model use the same semantics and the same reference coordinates to indicate features in the environment, which we consider an important step in reaching shared semantic understanding of indoor environments between humans and robots.

BIM models

3D BIM modelling software and BIM modelling processes are increasingly taking over the Architecture, Engineering, and Construction (AEC) industry. In many countries, newly built buildings are modelled in BIM software, and delivered to the client as an as-built model. The operational phase of the building predominantly relies on Facility Management Information Systems (FMIS), which typically have considerably less detailed 3D geometric data, and instead focus heavily on the collection of sensor data (access, temperature, air quality, ventilation, etc.). This collection of data is often termed 'digital twin': the digital counterpart of the physical building. A dominant data standard in the AEC industry is the Industry Foundation Classes (IFC) [14]. This standard focuses heavily on the interoperable exchange of 3D data across BIM authoring tools. An open and neutral IFC file can be exported from a BIM modelling tool, making semantic and 3D geometric data openly available (human- and machine-readible). Although this data source is used less often for existing buildings [15], it still provides an invaluable resource of information if it is available. Whereas IFC has always been available in the EXPRESS information modelling language, recent works have aimed elaborately at enabling XML, JSON and RDF formats for the same data. An XML format has been supported since the early 2010s, the RDF format for IFC data is available since 2016 [16], and a simplified JSON format is under construction at the time of research and writing [17].

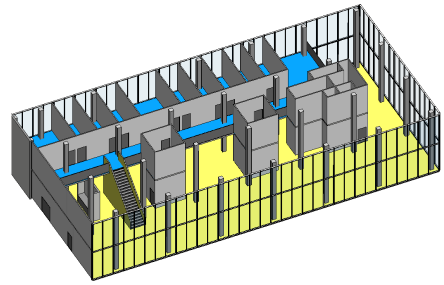


Fig. 1. Example of the geometry present in a BIM model (here shown as part of a storey). The model shown is part of the ATLAS building at TU Eindhoven and is used in this work. It is extracted from a larger BIM model.

Previous research has tried to leverage BIM models to tackle the challenge of path planning and localization. With respect to localization, BIM models have been mostly used for indoor image-based localization. Acharya et al. [18] generate a data-set of synthetic images with associated known 6-DOF camera locations and orientations from BIM models. The synthetic data-set is used to train a Deep Convolutional Neural Network (DCNN) which is used for indoor localization. Similarly, [19] generates a data set of synthetic images from the BIM model and trains a DCNN to extract features from the generated synthetic data set. Features extracted from the synthetic data are compared with features extracted from images of the physical environment and used to select the synthetic image (with known 6-DOF pose) that is closer to the physical image. Both works differ from what we present in this paper because the localization method is based on camera and a DCNN that has to be trained on the specific building model. Additionally, these papers do not focus on describing a methodology to automatically extract such information from the BIM models. Our approach differs because (a) the BIM model directly provides the features to look at without the need for training the DCNN and (b) we present a methodology to automatically extract relevant information from the model avoiding much of the error-prone manual effort. Other work has focused on using the BIM model to derive the topology of an indoor environment from which a path can be planned. In this case, localization is not further elaborated, as opposed to what we aim at in this article. In [20], the authors propose to extract information from BIM models to set-up a simulation environment (VEROSIM) for robotics development. The environment can connect the OMPL (Open Motion Planning Library) to the imported BIM model to generate collision free paths. On a similar line [21] derives a topological graph from BIM models upon which an A* planner can retrieve the optimal path. Although they are valuable and important reference works, our work aims precisely at a live localisation based on a current model of the building model and matching of geometric features.

III. CONNECTING BIM DATA TO THE ROBOT'S WORLD REPRESENTATION

A. World model representation

We use the term *world model* to describe the robot's internal representation of itself and its surroundings (i.e., the map) in relation to its sensors. To accommodate this representation, we use a property graph data model (Figure 3) which enables composition of different domain models. This general data model has been used extensively in knowledge representation for robotics (see [22] for a review). In this work, we focus only on the semantic map and its relation to available sensors. To represent the property graph we use the JSON-LD host language, which provides the mechanisms for attaching a unique symbolic id (`@id`), model id (`@type`) and namespace reference (`@context`) to every entity in the world model. For the geometric primitives in this work, we use 2D "Simple Feature Access" [10] representations. We augment these primitives by giving each point that belongs to a composition (such as polygons) an individual `id`, allowing to maintain topological consistency (e.g. walls that share a point which resembles a corner). These geometric representations are then associated with the semantic entities by `represented_by` relations, allowing to loosely couple different representations when desired. Furthermore, we explicitly relate representations of objects to the sensors through which they can be perceived. We introduce the `ObjectFeatureRepresentation` relation that connects the geometry representation, the object and the sensor, allowing the robot to query for features that it can perceive using its sensors.

The property graph model for localization has to be generated from the BIM model for all relevant objects. This procedure is schematically depicted in Figure 5. First, the BIM model is queried for relevant objects, which can appear at different positions in the model hierarchy (e.g., an `IfcWall` can be declared as part of a space, or as a connected attribute of another `IfcWall`). For this reason, the BIM model of a floor of the building is exported to an IFC-JSON representation [17] which is then made into valid JSON-LD by adding a `@context` specifier for the types and relations. This JSON-LD representation can then be "framed" by the JSON-LD API [23], turning it into a tree structure where objects of interest are at the root for processing without requiring a graph database. In this paper, we use the `IfcColumn` and `IfcWall` entities from the IFC model for localization. These entities are queried from the model together with their `representation` and `objectPlacement` relations. For the columns, the `sweptArea` 2D representation is converted into a polygon profile, for which the local coordinates are converted into global coordinates using the column's `objectPlacement`. For the walls, an `IfcPolyline` represents the center line, together with an offset for the thickness. This representation is converted into a slightly different representation with two (inner and outer) polylines, that connect to adjacent wall segments using corner points (see Figure 2). This preserves the topology of wall segments and corners.

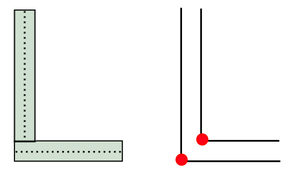


Fig. 2. Conversion of the BIM line-with-offset representation of a wall (left) to a representation consisting of two polylines for each wallsegment with a shared Point on the corner connecting two wallsegments (right).

The final result is exported as a JSON-LD property graph, which is partly visualized in Figure 3. A small section of the resulting map of a floor of the building considered later in this work is represented in Figure 4.

B. Data conversion

C. Spatial database and queries

For querying the spatial features and their semantic relations, we use the well-known PostgreSQL database with the PostGIS spatial extension. We store the property graph representation in the database as well, making it possible to query for relations and entities using the SQL query language. The database is queried for spatial features that are close to the robot. The sensor type is part of the query, resulting in features that are part of an `ObjectFeatureRepresentation` perceivable by the given sensor. The query returns the feature id, feature type, object id and object type for each feature, together with the spatial object that contains the actual coordinate data structure. For example, a query for perceivable features with a 2D LiDAR near the current position, may return the object `{type:"IfcColumn", id:"96033e"}`, together with its representation `{type:"Polygon", id:"553236"}`. This explicit symbolic link between the geometry, its interpretation and the object will be maintained in the association-based localization approach.

IV. LOCALIZATION

While the robot is tracking its location, it queries semantic map features from the database that are perceivable by its on board sensors and are currently within a certain radius around the robot's location on the map. We refer to these as *map features*. Our approach then tries to match them with features

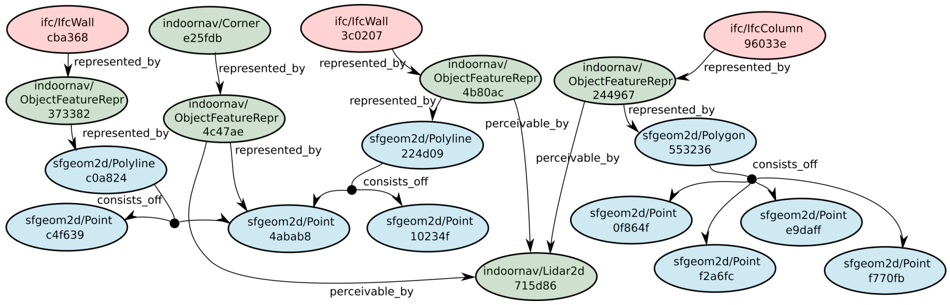


Fig. 3. Graph representation of the semantic entities from the BIM model and the geometry representations that are perceivable by the sensor, in this case the 2D LiDAR. Different domains (i.e., simple feature geometry, IFC entities and navigational relations) are shown in different colors. The id of entities that are not part of the BIM model are randomly generated.

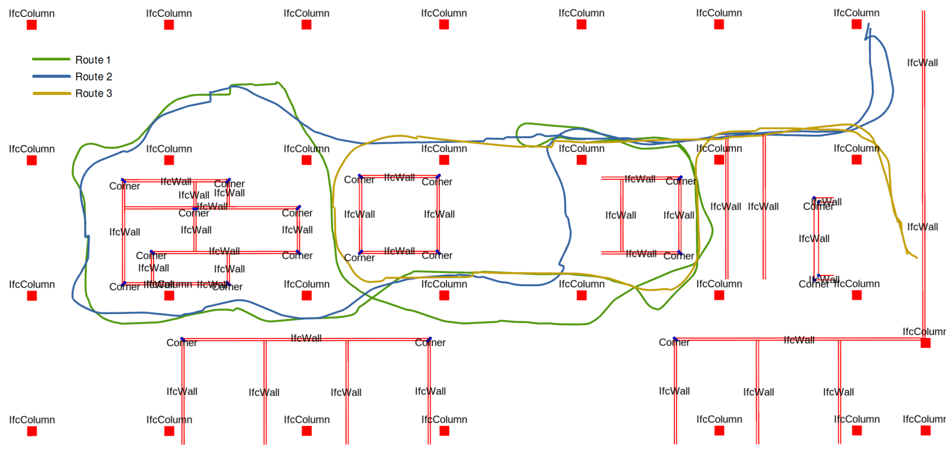


Fig. 4. The map considered in this work, as generated from the BIM model, with static features relevant for the LiDAR sensor. The features are annotated with the types of the objects they represent. The paths driven by the robot are shown, as determined by our localization approach, starting at different positions. All three paths finish at their respective starting positions. The final trajectory was recorded while three actors were actively obscuring the laser field of view. Notice that at some points small jumps occur in the map position, because of new associations with structural elements of the building.

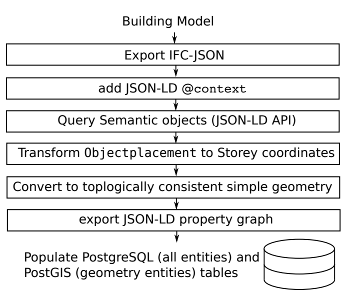
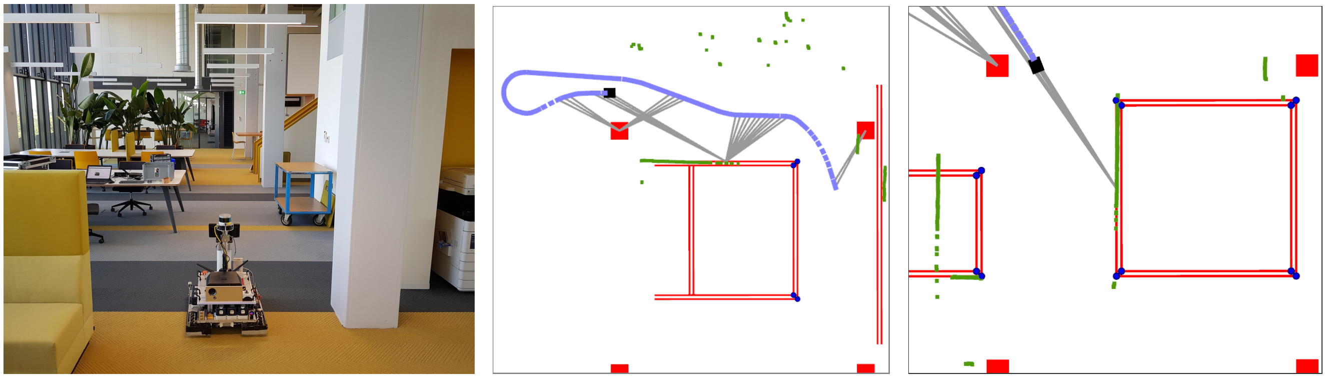


Fig. 5. Conversion from the BIM model to the representation used by the robot, stored in a database, as used in this work. This database contains both the property graph entities and the geometric entities (using PostGIS).

found in its sensor data, so called *sensor features*. This map-query-first approach allows to extract only sensor features that are currently of interest. The current implementation supports the extraction of line features, corner features and box features from the sensor data, based on a split-and-merge weighted line fitting implementation from [24], [25]. The corner and box features are obtained by checking if the lines support the well-known L-shape used often for rectangular objects. In the configuration of the detectors, we demand that the line segments used for wall and corner detection are sufficiently long to be insensitive to open doors. While this threshold is currently manually set to 1.2m, it could be derived from the `IfcDoor` representations in the BIM model if these are present.

The measurements are added to a factor graph containing the robot poses over a variable horizon, as well as range-bearing measurements to perceived objects. For columns and corners we use the well-known range-bearing measurement model. For walls, we use a model that constrains the angle and distance to the wall, but not the position alongside it. If a match in the sensor data is found based on the features suggested by the map (e.g., a line segment that falls within the line segment of the wall representation up to a threshold), the feature and measurement get added to the factor graph. We explicitly reference the `id` of the features in the factor graph, making the data associations with the map explicit. This feature-based approach has benefits over scan matching based approaches such as ICP, because it first checks if the sensor data supports the primitive feature suggested by the map to be usable. This also allows to disable individual features without removing them from the map by modifying their *perceivable_by* relation.



(a) (b) (c)

Fig. 6. (a) Picture of the indoor environment with the robot. (b) The same location showing the localization output. Matched semantic features are indicated by grey lines originating from their corresponding robot pose. LiDAR points are in green. (c) An example of the mismatch between the BIM model and the actual environment, making localization more difficult. The location of the square space showed a significant mismatch with reality, causing the robot location to jump while making correct associations. Furthermore, due to glass and doors, not all walls are always perceivable.

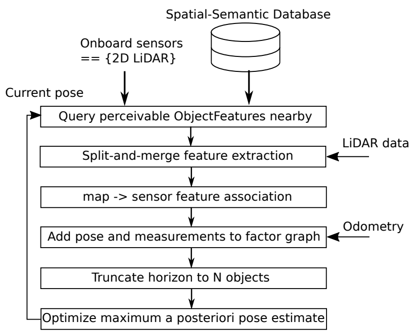


Fig. 7. Visualization of the localization approach that queries features from the database, matches sensor features to them and optimizes the resulting factor graph.

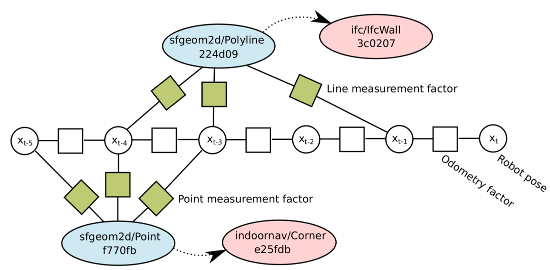
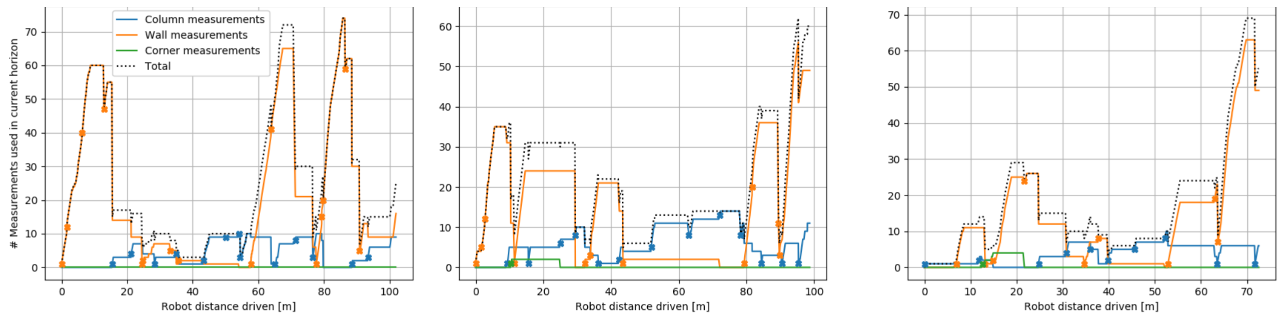


Fig. 8. Example of a factor graph containing the features and measurements within a certain horizon of robot poses.

Since the semantic features we use can be relatively sparse, substantial drift corrections are possible. We use graph optimization to avoid inconsistencies due to linearization. We do not focus on the trade offs of graph-based mechanisms for pure localization in this work (See, e.g. [26], [27] for a comparison of factor graphs and particle filters for pure localization). Our main focus is to show that this method makes observations explicit. Exploiting these explicit links for robust navigation tasks or updating the BIM model real-time using SLAM is potential future work. We use a moving-horizon approach in which the horizon is adjusted based on the number of geometric features that have been uniquely associated with the map. When more than N of these unique geometric map features are present in the horizon, we remove the oldest features and their corresponding measurements until again N features are present. We have implemented the localization in C++ using the GTSAM [28] optimization library together with our own in-memory semantic graph model and bookkeeping functions.

V. EXPERIMENT WITH A ROBOT PLATFORM

In this section, we show an experiment with a mobile robot on a floor of the building that is shown in Figure 1 (Atlas building, TU Eindhoven) for which an IFC model is available. We use a custom-made platform (Fig. 6a) equipped with mecanum wheels, wheel encoder odometry and a Hokuyo UTM30-LX 2D LiDAR scanner mounted upside down, close to the floor. Due to its mounting position, we have a 180° field of view consisting of 720 scan points. We teleoperate a total of three routes (approx. 100 meters each) with the robot, starting from different initial poses. The latter are provided manually to our algorithm by initial pose estimates. The environment is cluttered with both semi-static objects (furniture, plants) and dynamic objects (chairs, carts) which are not present in the semantic map used for localization, as shown in Figure 6a. In the third route, three actors are walking around in the field of view of the robot. The robot's feature detection is triggered whenever a distance of 15 cm has been driven and the map is then first queried for features that are visible to the 2D LiDAR within a range of 6 meters. The sensor data is then processed to search for sensor features that support the object features on the map and the pose estimate is updated. Figure 4 shows the 2D map with the features perceivable by the 2D LiDAR and the three trajectory estimates resulting from our localization approach. The horizon length is truncated at all times based on three semantic object references. Figure 6b shows an example of the horizon in which these references are visualized by



(1) (2) (3)

Fig. 9. The amount of feature measurements in the horizon used at each time step for pose optimization, grouped by the object type of the feature (note that there is overlap until an object gets removed from the horizon). The markers indicate a new object entering the horizon, after which the objects is often spotted multiple times, increasing the measurement count. The horizontal axis is discrete, taking steps of 15 cm (the update trigger distance).

edges (grey lines). Using our approach, the robot was able to successfully track its pose by incrementally associating perceived features with objects on the map generated from the BIM model. In the next section, the results are discussed in more detail.

VI. RESULTS AND DISCUSSION

The amount of measurements associated with the building model in the horizon at that timestep are visualized in Figure 9, grouped by object type. Both the number of associated measurements with an object of that type and newly detected objects entering the horizon are depicted (the latter by markers). The resulting trajectories are shown in Figure 4. The object types (and their features) used for localization in our approach were selected because of their availability in the BIM model and their saliency. No false positive association was made by the localization algorithm in our experiments. We selected these features because we predict that their saliency can carry over to indoor environments different than the one we targeted. In Figure 6 it can be seen that the wall features are very salient: once detected, they have a high recall and are spotted multiple times within the horizon. The column features are consistently spotted as well, but because of the L-shaped visibility requirement, they are only detectable from certain positions with respect to them. Again, this approach was deliberately chosen to favor precision (i.e., minimize the occurrence of false positives). The corner features are spotted less frequent because of the same L-shape detection requirement. Spurious corner measurements can give rise to inconsistent matches and are avoided by this stringent shape requirement. Furthermore, from Figure 6 we can see that columns and walls are both necessary for consistently having recent features within the horizon. These features are to be found in many buildings, making our approach applicable in many cases where a sufficiently rich BIM model is available. However, we emphasize that also the semantic explainability of local detections is an important feature of our work, that will come to its full right when exploited in context of different robot tasks, such as navigational rules (e.g. driving close to walls or using a column as natural waypoint) and when interaction between humans (e.g., facility managers) and robots will take place.

Another observation from our experiments is that the BIM model is not always accurate or complete. Spatial inaccuracies were present, one of which is shown in Figure 6c. Although we did not deal with these inconsistencies explicitly (recovery mechanisms are potential future work), our method is able to make the right associations in the considered building model. We do note that robustness against unmodeled dynamic clutter is not incidental, because we focus on features that are known to be static for localization. A final remark is that the relative inaccuracy of the BIM model raises an important question about the definition of accuracy. Whether we want to define accuracy as the metric deviation of the position in a map coordinate system, or as the correctness of perceiving local objects of interest with respect to the robot remains a relevant question. The latter enables a definition of accuracy in the case where supplied maps are not perfect but semantic navigation based on correct associations can be robust nonetheless.

VII. CONCLUSIONS AND FUTURE WORK

In this work, we showed that existing semantic building information models can provide a robot with enough information to localize itself, providing a great opportunity for automatic deployment in large buildings without prior work or adaptation. We also showed how this semantic information can be translated to explainable associations, used for localization by a robot equipped with a 2D LiDAR, using a property graph database to let the robot query its semantic environment. The opportunities we see for future work will consist of generalizing our approach to different semantic entities that are present in BIM models (such as doors or curtain walls) and using different sensors to perceive them. Furthermore, keeping the BIM map up-to-date and consistent is an important challenge, with great potential to be of use for maintaining digital twins in the operational phase of buildings. To conclude, we foresee that automatic deployment of robots in buildings can be useful in many scenarios and our work has explored important steps in making this a possibility.

References

[1] *Ropod project*. [Online]. Available: http://www.ropod.org/ (visited on 09/28/2020).

[2] T. Shimmura, R. Ichikari, T. Okuma, H. Ito, K. Okada, and T. Nonaka, "Service robot introduction to a restaurant enhances both labor productivity and service quality," in *Procedia CIRP*, vol. 88, Elsevier B.V., Jan. 2020, pp. 589–594.

[3] K. Dautenhahn, A. Campbell, and D. Syrdal, "Does anyone want to talk to me?: Reflections on the use of assistance and companion robots in care homes," in *Procs 4th Int Symposium on New Frontiers in Human-Robot Interaction*, The Society for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB), 2015, pp. 17–20.

[4] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[5] K. J. Li, S. Zlatanova, J. Torres-Sospedra, A. Perez-Navarro, C. Laoudias, and A. Moreira, "Survey on indoor map standards and formats," *2019 Int. Conf. on Indoor Positioning and Indoor Navigation*, 2019.

[6] L. Naik, S. Blumenthal, N. Huebel, H. Bruyninckx, and E. Prassler, "Semantic mapping extension for OpenStreetMap applied to indoor robot navigation," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 3839–3845, 2019.

[7] H.-K. Kang and K.-J. Li, "A Standard Indoor Spatial Data Model—OGC IndoorGML and Implementation Approaches," *ISPRS International Journal of Geo-Information*, vol. 6, no. 4, p. 116, Apr. 2017.

[8] L. S. Tessema, R. Jäger, and U. Stilla, "Extraction of IndoorGML Model from an Occupancy Grid Map Constructed Using 2D LiDAR," *Proceedings of the German Society for Photogrammetry, Remote Sensing and Geoinformation, 39st Conference, Vienna, Austria*, 2019.

[9] H. Deeken, T. Wiemann, and J. Hertzberg, "Grounding semantic maps in spatial databases," *Robotics and Autonomous Systems*, vol. 105, pp. 146–165, Jul. 2018.

[10] *Simple Feature Access - Part 1: Common Architecture | OGC*. [Online]. Available: https://www.ogc.org/standards/sfa (visited on 09/28/2020).

[11] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, "A pose graph-based localization system for long-term navigation in CAD floor plans," *Robotics and Autonomous Systems*, vol. 112, pp. 84–97, 2019.

[12] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, *BIM handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, 2nd. Hoboken, New York, United States: John Wiley & Sons, 2011, p. 650, ISBN: 978-0-470-54137-1.

[13] A. Borrmann, M. König, C. Koch, and J. Beetz, "Building Information Modeling: Why? What? How?" In *Building Information Modeling: Technology Foundations and Industry Practice*. Springer International Publishing, 2018, ch. 1, pp. 1–24, ISBN: 978-3-319-92862-3.

[14] *Industry Foundation Classes (IFC) - buildingSMART Technical*. [Online]. Available: https://technical.buildingsmart.org/standards/ifc/ (visited on 08/19/2020).

[15] R. Volk, J. Stengel, and F. Schultmann, "Building Information Modeling (BIM) for Existing Buildings - Literature Review and Future Needs," *Automation in Construction*, vol. 38, pp. 109–127, 2014.

[16] P. Pauwels and W. Terkaj, "EXPRESS to OWL for Construction Industry: Towards a Recommendable and Usable ifcOWL Ontology," *Automation in Construction*, vol. 63, pp. 100–133, 2016.

[17] *IFCJSON-Team/IFC.JSON-4: Repository containing the specification for IFC.JSON*. [Online]. Available: https://github.com/IFCJSON-Team/IFC.JSON-4 (visited on 08/19/2020).

[18] D. Acharya, K. Khoshelham, and S. Winter, "BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 245–258, Apr. 2019.

[19] I. Ha, H. Kim, S. Park, and H. Kim, "Image-based indoor localization using BIM and features of CNN," *ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things*, no. Isarc, 2018.

[20] C. Schlette and J. Roßmann, "Sampling-based floor plan analysis on bims," in *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC)*, International Association for Automation and Robotics in Construction (IAARC), Jul. 2016, pp. 28–35.

[21] W. Palacz, G. Ślusarczyk, B. Strug, and E. Grabska, "Indoor robot navigation using graph models based on BIM/IFC," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11509 LNAI, Springer Verlag, Jun. 2019, pp. 654–665, ISBN: 9783030209148.

[22] D. Paulius and Y. Sun, "A Survey of Knowledge Representation in Service Robotics," *Robotics and Autonomous Systems*, vol. 118, pp. 13–30, Aug. 2019.

[23] *JSON-LD 1.1 Framing*. [Online]. Available: https://json-ld.org/spec/latest/json-ld-framing/ (visited on 08/19/2020).

[24] *GitHub - kam3k/laser_line_extraction: A ROS package that extracts line segments from LaserScan messages*. [Online]. Available: https://github.com/kam3k/laser_line_extraction (visited on 09/29/2020).

[25] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 1, 2003, 1304–1311 vol.1.

[26] C. Wu, T. A. Huang, M. Muffert, T. Schwarz, and J. Grater, "Precise pose graph localization with sparse point and lane features," in *IEEE int. Conf. on Intelligent Robots and Systems*, vol. 2017, IEEE, Dec. 2017.

[27] D. Wilbers, C. Merfels, and C. Stachniss, "A comparison of particle filter and graph-based optimization for localization with landmarks in automated vehicles," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 220–225.

[28] F. Dellaert, "Factor Graphs and GTSAM: A Hands-on Introduction," Tech. Rep., 2012. [Online]. Available: http://tinyurl.com/gtsam.