

Article

Robot Localization Using Situational Graphs (S-Graphs) and Building Architectural Plans

Muhammad Shaheer ^{1,*}, Hriday Bavle ¹, Jose Luis Sanchez-Lopez ¹ and Holger Voos ^{1,2}

¹ Automation & Robotics Group, Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, 1359 Kirchberg, Luxembourg; hriday.bavle@uni.lu (H.B.); joseluis.sanchezlopez@uni.lu (J.L.S.-L.); holger.voos@uni.lu (H.V.)

² Faculty of Science, Technology and Medicine, University of Luxembourg, 1359 Kirchberg, Luxembourg

* Correspondence: muhammad.shaheer@uni.lu

Abstract: This paper presents robot localization using building architectural plans and hierarchical SLAM. We extract geometric, semantic as well as topological information from the architectural plans in the form of walls and rooms, and create the topological and metric-semantic layer of the Situational Graphs (*S-Graphs*) before navigating in the environment. When the robot navigates in the construction environment, it uses the robot odometry and 3D lidar measurements to extract planar wall surfaces. A particle filter method exploits the previously built situational graph and its available geometric, semantic, and topological information to perform global localization. We validate our approach in simulated and real datasets captured on ongoing construction sites presenting state-of-the-art results when comparing it against traditional geometry-based localization techniques.

Keywords: localization; SLAM; Building Information Modelling (BIM)

1. Introduction

Robots are starting to be used on construction sites to perform various tasks such as autonomous data capturing to continuously monitor the evolution of the construction site, while construction environments are challenging because of their dynamic nature, as they can change significantly over a span of a few days, they have some features that are common in most of them, for example, walls, rooms, or pillars. Moreover, many companies are now integrating digital tools such as Building Information Modelling (BIM) [1] in their project design and execution phases. BIM contains geometric, semantic, topological, and physical information about each and every element of the building. Such crucial information is invaluable for a robot, which can use this knowledge for localization and understanding the construction site around it. Nevertheless, nowadays robots are unable to exploit the power of these digital plans. When they navigate in the environment, they either build their internal maps from scratch, which makes it difficult for them to know where they are, or they require fiducial markers to localize properly.

Most of the ongoing construction sites can be well represented using walls and rooms, to provide the robot with pre-existing knowledge of the environment. Robots utilizing only geometric information for localization within a construction site may not be able to localize correctly due to the fact the ongoing site does not resemble yet the architectural plans. Semantic information can help improve the localization but still can suffer inaccuracies due to large deviation of the robot pose estimate from its actual estimate and similar semantic information in case of repetitive environments. Thus, research is still required to combine the complete geometric, semantic, and topological information provided by the BIM models, to robustly localize a robot within an ongoing construction environment.

This paper proposes a novel method in this direction to localize a robot in the construction environments using information extracted from architectural plans, and Situational Graphs (*S-Graphs*) [2]. We extract this geometric (planar walls), semantic (type of planar



Citation: Shaheer, M.; Bavle, H.; Sanchez-Lopez, J.L.; Voos, H. Robot Localization Using Situational Graphs (S-Graphs) and Building Architectural Plans. *Robotics* **2023**, *12*, 65. <https://doi.org/10.3390/robotics12030065>

Academic Editor: Dan Zhang

Received: 21 March 2023

Revised: 21 April 2023

Accepted: 25 April 2023

Published: 28 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

walls), and topological information (room constraining four walls) from the architectural plans, which is first utilized to estimate the initial pose of the robot in the building, using Monte Carlo Localization. Once the initial pose is estimated, the robot uses the knowledge from topological and metric-semantic layers of *S-Graph*, created in the first step, to further localize itself as it navigates in the environment, and improve the mapping accuracy of *S-Graph*. It is to be noted that we are assuming the accuracy of the knowledge derived from the architectural plans. Hence, in the event of a discrepancy between an entity (wall or room) observed by the robot and the corresponding entity in the architectural plan, the error is reduced by correcting the observation instead of the plan.

Our main contributions to this paper are:

- Creating topological and metric-semantic layers of *S-Graph* by extracting geometric, semantic, and topological information of a building from its architectural plan.
- Particle filter-based estimate of the global pose of the robot within the architectural plans using the extracted geometric, semantic as well as topological information.
- Hierarchical *S-Graphs* [2] based localization of the robot, utilizing the global estimate from the particle filter and the extracted information from the architectural plans.

2. Related Works

2.1. Global Localization

Global localization involves determining the position of a robot on a map that is already known. It is typically done in two steps. The first step is initial localization, which involves matching sensor readings to the map. This is sometimes called *the kidnapped robot problem* [3], *re-localization* [4], or *place recognition* [5]. Multiple sensor modalities, including LiDAR [6], monocular [7], and stereo [8] cameras, can be employed for such tasks. One of the initial studies in this area was conducted by Fox et al. [9], who utilized a Markov model to update the probability distribution over potential locations in the environment based on sensor readings. Another method proposed by [10], called Monte Carlo Localization (MCL), models the robot state as a group of particles randomly sampled from the probability density function. MCL has been found to be highly effective in various situations [11–14]. Koide et al. [15] proposed global localization using 3d LiDAR measurements based on an Unscented Kalman Filter (UKF).

Alternative methods such as [16–18] utilize the extraction and matching of local point cloud descriptors. Key-point-based approaches face limitations due to the restricted uniqueness of local structures in LiDAR data. Nevertheless, they have gained traction as a substitute for those based on scan matching, attracting significant interest from the research community in recent years [19,20]. Learning-based localization has also shown promise in recent research. One example is SegMap, as presented in [21], which employs a distinct map representation for localization by extracting segments from 3D point clouds.

2.2. Architectural Plans Based Localization

In robotics use of 2D CAD-based models is very common to generate a 2D map of the environment and to localize a robot using the on-board range sensors [22,23]. Ref. [24] utilizes a monocular system to localize the robot in the 2D floor plans extracting the room layout edges. Refs. [25,26] incorporate semantic textual cues into the 2D floor plans to further improve the localization accuracy.

BIM on the other hand provides additional semantic and topological information as compared to 2D CAD plans which can be exploited to improve the planning and localization of mobile robots. Authors in [27–29] utilize BIM information for extracting 3D maps utilized by path-planner but do not include this information for robot localization. Authors in [30] only present a small case study localizing the robot through image matching between real images and images from BIM plans, but only restricted to a certain perspective. Ref. [31] present an approach to extracting information from BIM, splitting it into different floor levels, and then converting it to a relevant format used by a state-of-the-art pose-graph SLAM algorithm. Since the pose-graph algorithm cannot take as input an entire 3D mesh

of the area, the authors have to create an additional module to generate trajectories within the mesh storing pose and sensor data at different intervals to later input it the SLAM for localization. Whereas in [32], authors present a case study converting BIM into 3D meshes at different floor levels and using a simple Iterative Closest Point (ICP) algorithm to localize the robot, thus depending highly on the metric ICP algorithm for convergence, which can show inaccuracies if an on-going construction site does not yet resemble the BIM plans. Compared to previous approaches, ref. [33] extract semantic information in addition to geometric information from the BIM plans to localize the robot using an on-board 2D LiDAR and pose-graph optimization, but the authors do not consider additional topological information connecting different walls to a room, etc.

2.3. Scene Graph Based Localization

Scene graphs are emerging research to represent the scene in several geometric, semantic, and topological dimensions [34–38]. *S-Graphs* [2] is a real-time optimizable hierarchical scene graph that represents the environment in geometric, semantic, and topological dimensions. *S-Graphs* is capable of not only representing the environment as a scene graph but also simultaneously optimizing all the elements within it including the robot pose. All of these methods need the robot to navigate the environment in order to create the scene graphs, which means they need to know the starting point of the robot beforehand. They lack the capability of performing global localization. Some methods have also emerged exploiting scene graphs for localization such as [39–41] performing graph-based localization utilizing spatial information between the neighboring landmarks. Similarly, authors in [42] create a global topological semantic map of the environment and later use similarity between semantic scene graphs to localize the robot. Ref. [43] combine traditional 3D object alignment with graph-based matching to perform global localization in small-sized areas such as rooms.

None of these approaches use the knowledge of the environment available in the form of digital plans. Therefore, inspired by the current state-of-the-art, we present our novel approach which combines BIM and scene graphs to provide robust localization of robots. Our approach extracts information from BIM, but we can efficiently input the extracted geometric, semantic, and topological data to a hierarchical graph [2] without the need of moving the robot or generating additional trajectories within the virtual 3D mesh to create the map of the environment.

3. Proposed Approach

3.1. Overview

Our approach can be divided into three main modules: building information extraction module, particle filter-based localization module, and *S-Graphs* [2] localization module. We define the global world frame of reference W which coincides with the origin frame of reference defined within the architectural plans. The odometry of the robot frame R_t is computed with respect to the odometry frame O . Lidar measurements from the 3D LiDAR are received in the LiDAR frame L_t . The building information extraction module extracts all the relevant information from the architectural plans. The particle filter utilizes the geometric/semantic information i.e., mapped planar walls along with the type of wall (vertical x/y -axis) as well as the topological information of the rooms, to estimate the initial alignment between the odometry frame and the world frame ${}^W x_{t_1 O}$. The *S-Graphs* localization module receives the mapped walls and the room information to generate beforehand the top two layers of the *S-Graphs* (i.e., metric-semantic layer and topological layer), which start getting connected to the first layer (robot tracking layer) which is generated after the reception of the initial pose ${}^W x_{t_1 O}$ and at different intervals as the robot navigates. An overview of our approach is shown in Figure 1.

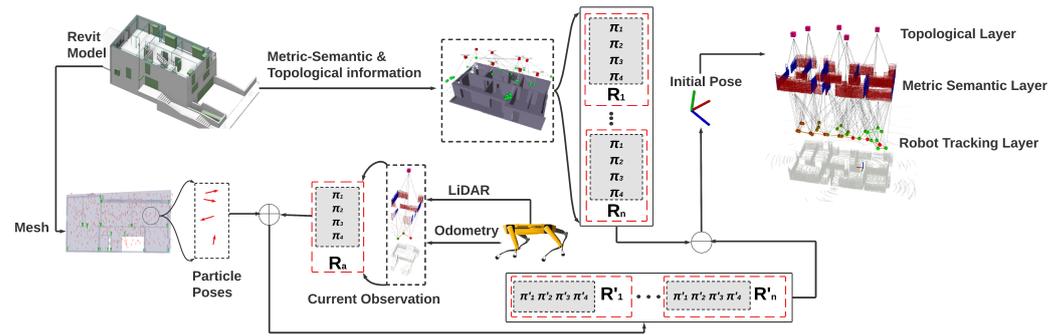


Figure 1. Visualization of the proposed approach. Initially, the *S-Graph*'s topological and metric-semantic layers are generated by extracting data from a building's architectural plan, made in Revit. Each room \mathbf{R} 's four walls are denoted by $\pi_1, \pi_2, \pi_3,$ and π_4 . Following that, a particle filter is initialized, and particles are dispersed throughout the entire environment. As the robot navigates the environment, it observes the walls and rooms. These observations are converted into each particle's frame of reference, and global localization is achieved by comparing them to the data extracted from the architectural plans. Finally, the robot tracking layer is added to the *S-Graph* created in the first step.

3.2. Building Information Extraction

Architecture plans of buildings are made in various formats. In this paper, we use the International Foundation Class (IFC) format [44]. An IFC file is a Building Information Model (BIM) file that is platform-independent, meaning that an IFC file can be used in any BIM software. We propose to extract geometrical, semantic, and topological information from the IFC file and exploit it for localization and environmental understanding by the robot.

3.2.1. Rooms Information Extraction

In IFC files, the rooms are defined by a parameter named *IfcSpace*. This parameter further has more attributes that give the information about the area and the position of the room $[^W\rho_x, ^W\rho_y]$ in the world frame.

3.2.2. Walls Information Extraction

Walls are arguably the most common elements to be found in an IFC file. Each wall has its own geometrical and physical properties. Each of these walls can be used as a landmark as it has a fixed defined position and thickness. In the IFC file, the wall is defined by *IfcWall* and *IfcWallStandardCase*. We extract the location, orientation, and thickness of each wall from the IFC file. We use the general form of a plane equation to extract the perpendicular distance of the wall $^W\mathfrak{B}$ in the world frame W as:

$$^Wd = -1 \cdot (a \cdot ^Wn_x + b \cdot ^Wn_y + c \cdot ^Wn_z) \tag{1}$$

Here $a, b,$ and c are the $x, y,$ and z coordinates of the starting point of the wall, respectively, and $^W\mathbf{n}_\pi = [^Wn_x, ^Wn_y, ^Wn_z]$ are the normal orientations of the wall. In the IFC file, each wall is considered as a single wall, but for the robot, they are two different walls with opposite orientations known as the double-sided issue [45]. We solve this issue by creating a duplicate wall for $^W\mathfrak{B}$ with opposite orientation and increasing its perpendicular distance Wd by the wall thickness as:

$$^W\mathfrak{B}^D = \begin{bmatrix} -^W\mathbf{n}_\pi \\ ^Wd_\pi + \mathcal{T}_\pi \end{bmatrix} \tag{2}$$

Here $^W\mathfrak{B}^D$ is the opposite wall of $^W\mathfrak{B}$ and \mathcal{T}_π is the thickness of wall $^W\mathfrak{B}$.

3.2.3. Topological and Metric-Semantic Layer Generation

We create the topological and metric-semantic layers of *S-Graph* of a building by using this extracted information from the BIM model. The topological layers consist of *Rooms*

of the building, whereas the metric-semantic layers consist of the *Walls*. These two layers of the *S-Graph* are then used by the robot to extract information and localize itself in the environment, further explained in Section 3.4.

3.3. Localization Using Particle Filter

We introduce a novel topological information factor in particle filter-based localization. The particles are matched with only the walls of the rooms in which they lie in, instead of matching them with all the walls of that story of the building. The particles are uniformly distributed within the boundaries of the environment, in world reference frame W . The state of each particle is defined as $x^n = [x, y, z, \mathbf{q}]^T$ where n is from 1 to N , and x, y, z are positions of a particle in world frame W , and $\mathbf{q} = [q_x, q_y, q_z, q_w]$ is a quaternion representing the orientation of the particle in W frame. Figure 2 shows an overview of our particle filter algorithm.

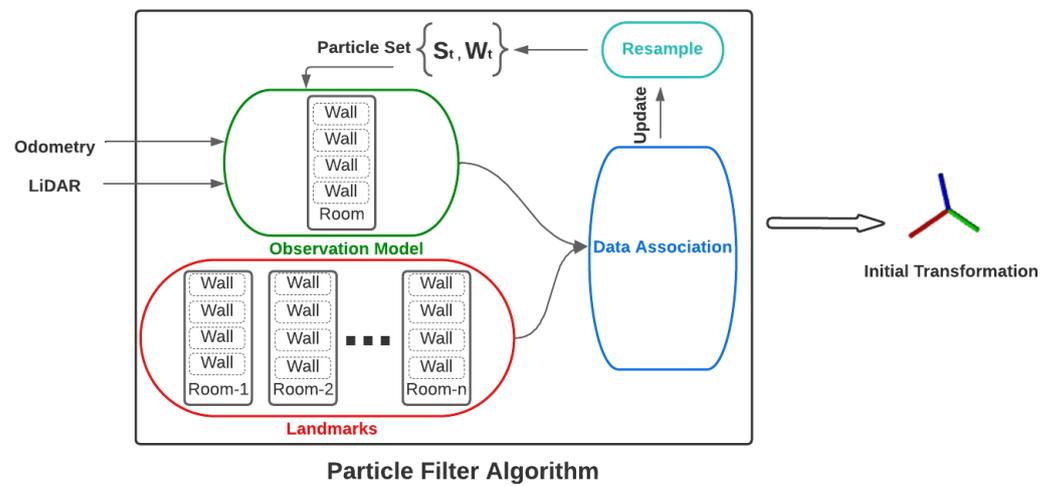


Figure 2. Particle filter algorithm overview. Walls and rooms are detected by using robot odometry and LiDAR data and transformed into each particle’s frame. These observations are then associated with landmark walls and rooms extracted from the architectural plan. Afterward, the particle weights are updated and resampling is performed which eventually gives the initial transformation upon convergence.

3.3.1. Observations

In our case observations are detected and segmented planar walls from the 3D LiDAR measurements as the robot navigates within its environment. The plane extraction algorithm is inspired by [2] and uses a sequential RANSAC to extract all the planar coefficients of all the planar surfaces at a given time instant. The planes are detected in LiDAR frame L_t and converted to their Closest Point (CP) representation ${}^{L_t}\mathbf{\Pi}$ [46] are defined as:

$${}^{L_t}\mathbf{\Pi} = {}^{L_t}\mathbf{n}' \cdot {}^{L_t}d' \longrightarrow \begin{bmatrix} {}^{L_t}\mathbf{n} \\ {}^{L_t}d \end{bmatrix} = \begin{bmatrix} {}^{L_t}\mathbf{\Pi} / \|{}^{L_t}\mathbf{\Pi}\| \\ \|{}^{L_t}\mathbf{\Pi}\| \end{bmatrix} \quad (3)$$

where ${}^{L_t}\mathbf{n} = [{}^{L_t}n_x, {}^{L_t}n_y, {}^{L_t}n_z]^T$ is the plane normal and ${}^{L_t}d$ is the distance to the origin, both in the LiDAR frame.

3.3.2. Landmarks

We use walls of the building, extracted from the architectural plans, as landmarks in our approach. These walls are represented as planar surfaces in the W frame. The planar coefficients of all the walls are obtained using Equations (1) and (2).

3.3.3. Prediction

In the prediction step, the particle pose is updated using 3D robot pose obtained from the odometry either provided by the encoders of robot platform or using the measurements

of the 3D LiDAR. The state of each particle is propagated from time $t - 1$ to t based on a motion model defined as:

$$\mathbf{x}_t^{[n]} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{[n]}, \mathbf{u}_{R_t}) \quad (4)$$

$\mathbf{x}_{t-1}^{[n]}$ is the pose of n -th particle and \mathbf{u}_{R_t} is the incremental pose of robot between time instance $t - 1$ and t :

$$\mathbf{u}_{R_t} = {}^O\mathbf{x}_{R_t}^{-1} \oplus {}^O\mathbf{x}_{R_{t-1}} \quad (5)$$

${}^O\mathbf{x}_{R_t}$ and ${}^O\mathbf{x}_{R_{t-1}}$ are the robot poses at time t and $t - 1$ in odometry frame O , respectively. Figure 3 shows the initialization, clustering, and convergence of the particle filter.

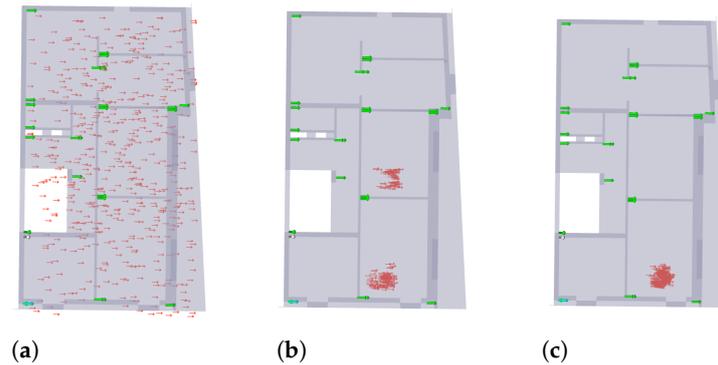


Figure 3. Top view of the particle filter localization with topological rooms information (a) The particles are initialized in the entire floor. (b) Particles form two clusters after the update step. Note the ‘clusters’ are formed in 2 rooms with similar geometry. (c) The particles successfully converge in the correct room and the initial pose is published.

3.3.4. Data Association

Our novel data association step, exploits not only the geometric/semantic information but also the topological information, exploiting the available top layers of the S-Graph and performing the data association in two different steps, as follows:

Room Association. The BIM model gives us the starting points $[{}^W\rho_x, {}^W\rho_y]$ of the rooms in W frame, and the areas of the rooms, as well as the walls constrained by each room, which can help define the boundaries of each room. We thus check the position of each particle, and, if it lies within the boundaries of a room, the currently observed planes are matched only to the walls belonging to that particular room instead of matching that particle’s observations with all the landmarks of the building. This procedure significantly reduces the ambiguity in data association, computation time, and cost as well.

Wall Association. After the room association step, we perform wall association by matching the observations to the mapped planar walls of the associated room’s. All the planar walls are semantically categorized either x vertical wall (with normal coefficient n_x greater) and y vertical wall (with normal coefficient n_y greater). As all the landmarks are defined in world frame W , for each i th particle with pose ${}^W\mathbf{x}_{P_i}$, we transform the observed planes in LiDAR frame L_t to the world frame W by :

$${}^W\mathbf{b}' = \begin{bmatrix} {}^W\mathbf{n}' \\ {}^Wd' \end{bmatrix} = {}^W\mathbf{T}_{\pi_i}({}^W\mathbf{x}_{P_i}) \begin{bmatrix} {}^{L_t}\mathbf{n} \\ {}^{L_t}d \end{bmatrix} \quad (6)$$

Here ${}^W\mathbf{T}_{\pi_i}$ is the transformation matrix that transforms the normal and distance of detected planes in world frame W . In order to compute the matching error between the observed and the mapped landmarks of each category (x and y wall), we use the minimal plane parametrization [2], where each plane is represented as ${}^W\pi = [{}^W\phi, {}^W\theta, {}^Wd], {}^W\phi$ and

${}^W\theta$ are the azimuth and elevation of the plane in world frame W , while matching, we check the error between observations and landmarks as:

$$\varepsilon = \|\Lambda^t \pi_i - \Lambda^t \tilde{\pi}_i\|_{\Lambda^t \tilde{\pi}_i}^2 \quad (7)$$

If the error between an observed plane and a landmark wall is less than a threshold $\varepsilon < th$, we associate the observation with that particular landmark, else the observation is rejected. We use the Mahalanobis distance between each observed plane and the landmark planes.

3.3.5. Update

Once the incoming observations are associated with the corresponding landmarks, the weight of the matching particle is calculated as:

$$\omega_i = \mu \cdot \exp(-(\phi_{diff} + \theta_{diff} + d_{diff})) \quad (8)$$

where,

$$\phi_{diff} = \tau \cdot ({}^W\phi' - {}^W\phi) \quad (9)$$

and $\mu = (2 \cdot \pi \cdot \sigma^2)^{-1/2}$, $\tau = (2 \cdot \sigma^2)$. θ_{diff} and d_{diff} are also defined similarly. The calculated weights are first normalized and then assigned to the respective particle. Re-sampling of the particles is then performed using importance sampling principle to draw N samples with probability proportional to the particle's weight.

3.3.6. Initial Transformation Estimation

Our implementation of the particle filter only estimates the initial transformation between the odometry frame O and the world frame W , once all the particles converge. In order to verify if the particles have converged, we take the current pose of the particle with the highest weight at time t , and check difference from the average pose of all the particles. If the difference is below a defined threshold, we consider the particles converged. We then take the pose of the particle with the highest weight, and compose it with the inverse of the robot pose estimated by the odometry in frame O at time t , to estimate the initial transformation between the odometry frame and the world frame:

$${}^W\mathbf{x}_{t_1O} = {}^W\mathbf{x}_{P_t} \oplus {}^O\mathbf{x}_{R_t}^{-1} \quad (10)$$

${}^W\mathbf{x}_{t_1O}$ is the initial transformation between frame O and W , whereas ${}^O\mathbf{x}_{R_t}$ is the robot pose in O frame and ${}^W\mathbf{x}_{P_t}$ is a pose of the particle with highest weight at the time of convergence.

3.4. S-Graph Based Localization

S-Graphs detects planes, rooms, and corridors as the robot navigates the environment to create the optimizable graph.

- First layer consists of the robot poses ${}^M\mathbf{x}_{R_t}$, $t \in \{1, \dots, T\}$ at T selected keyframes.
- The second layer is the metric-semantic layer which consists of P detected planes ${}^M\pi_i$, $i \in \{1, \dots, P\}$.
- The third and final layer is a topological layer which consists of S rooms ${}^M\rho_j$, $j \in \{1, \dots, S\}$, and, K corridors ${}^M\kappa_k$, $k \in \{1, \dots, K\}$.

In this work, we create the top two layers of *S-Graphs* from the information extracted from BIM. We create the metric-semantic layer (planar wall nodes) and topological layer (room node with appropriate edges constraining the room with its planar walls) of the *S-Graphs* for a given environment.

When the particle filter (Section 3.3) provides the initial pose estimation, *S-Graphs* starts creating its first robot-tracking layer and connects it with the metric-semantic and topological layers. *S-Graph* minimizes the cost function associated with each of the three layers to estimate the global state of the environment. Localization is performed by associating the detected

planes and rooms, to the walls and rooms of the building extracted from architectural plans, respectively. It is to be noted here that *S-Graph* based localization further refines localization and mitigates the error estimated by particle filter in initial transformation.

3.4.1. Plane Association

Building information extraction module of Section 3.2 creates a two-layered scene graph whose first layers comprise the walls of the building. As mentioned earlier, walls of the building are represented in the form of planes ${}^W\pi = [{}^W\phi, {}^W\theta, {}^Wd]$. Therefore, observed x vertical walls are matched with all the mapped x vertical walls of the building and similarly for y vertical walls. We check the Mahalanobis distance (Equation (7)) between the observed plane and all the walls of the building. If the Mahalanobis distance is less than the threshold, that observation is associated with that particular wall, else a new wall is created. Errors in the observation are thus corrected after plane association.

3.4.2. Room Association

Similar to the plane association, whenever a room is detected by *S-Graph*, the pose of the a room is matched with the pose of all the rooms of the building, and if the difference between the pose of the detected room and a particular room of the building is within a threshold, the observed room is matched to that room. We can safely tune the matching threshold close to the room widths, as rooms do not overlap. This allows us to merge planar walls duplicated due to inaccuracies in plane data association (Section 3.4.1).

4. Experimental Evaluation

4.1. Experimental Setup

We validated our proposed approach in multiple construction environments. In this paper, we performed experiments on single storeys only. We extracted semantic, geometrical, and topological information of various building models created in Autodesk Revit 2022 software (<https://www.autodesk.com/products/revit/architecture>, accessed on 31 August 2022). Experiments were performed both in simulated and real environments. The robot platform we used in these experiments was Boston Dynamics *Spot* (<https://www.bostondynamics.com/products/spot>, accessed on 31 August 2022) robot which was equipped with a Velodyne VLP-16 3D LiDAR. *Spot* was teleoperated in both simulated environments and real construction sites to collect data. The entire implementation of the proposed methodology was done in C++, and the experiments were done on an Intel i9 16-core workstation.

Building Information Extraction

We extracted information from multiple construction models created in Revit software. 3D Revit models were first exported to IFC files. These IFC files were then read by an open source library IFC++ (<https://ifcquery.com/>, accessed on 30 June 2022) and BimVision (<https://bimvision.eu/>, accessed on 20 February 2023) software to extract semantic, geometrical, and topological information of the building. IFC++ library was used to extract the floors/storeys of the entire building and its elevation value. Then for each storey, the walls and rooms contained within them were extracted.

Once we know which walls and rooms belong to which storey of the building, we extract their geometrical information from BimVision software. BimVision gives the starting location, i.e., x, y, z coordinates of every wall and room. This information is stored in a CSV file and then used to generate the topological and metric-semantic layer of the *S-Graph*. This information is published over ROS [47], which is then subscribed by the robot.

4.2. Results and Discussion

4.2.1. Simulated Experiments

We performed global localization on different environments to estimate the initial transformation between the *odom* frame and the *world* frame. These environments were

created by extracting the meshes from the Revit models of actual architectural models. Mesh files of each storey of the building are also extracted from BimVision using an IFC file. These meshes are then used in simulated experiments. The meshes were imported in Gazebo (<http://gazebosim.org/>, accessed on 30 May 2022) physics simulator to create the replica environments of building models. In the simulated experiments, the odometry was estimated only from LiDAR. All of these environments contain multiple rooms in them, and we performed the experiments by placing the robot in different rooms to estimate the robustness of the approach. The legged robot was placed in a random room and then teleoperated to collect data. Figure 4 shows the initial transform estimated by particle filter between the *odom* frame and the *world* frame.

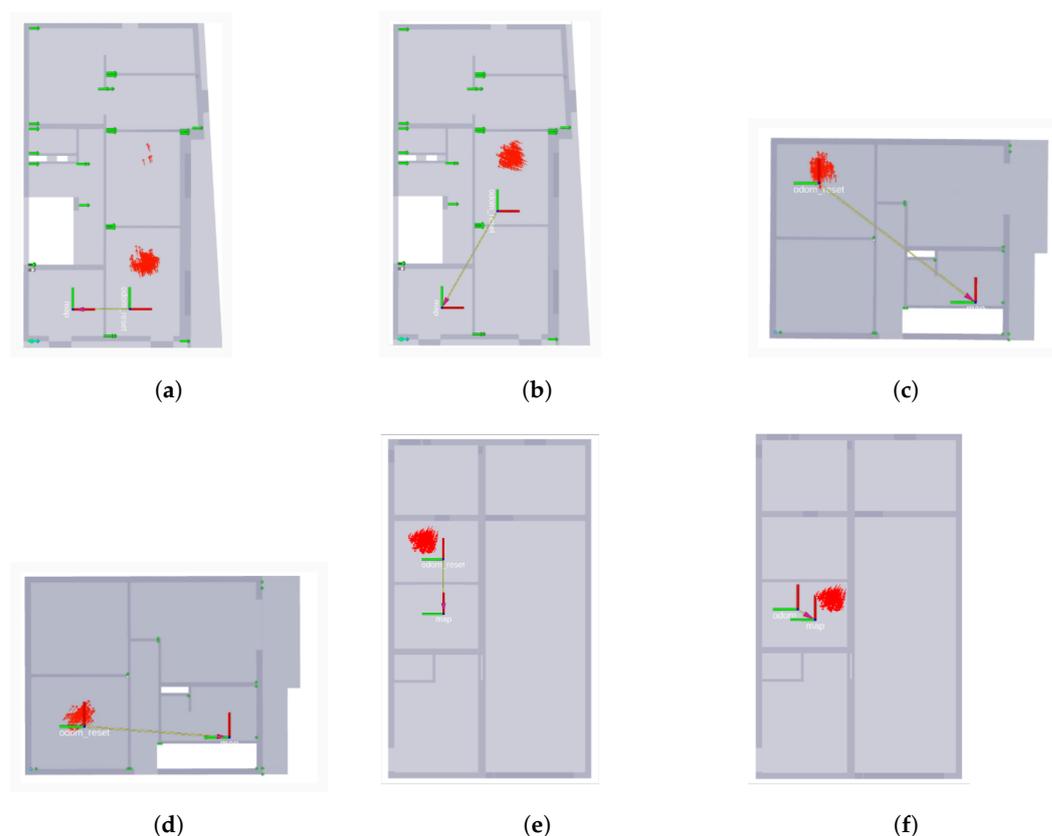


Figure 4. Top view of the initial transformation estimation by our approach in various environments. (a) D_1 sequence; (b) D_2 sequence; (c) D_3 sequence; (d) D_4 sequence; (e) D_5 sequence; (f) D_6 sequence.

We compared the performance of our proposed method against two state-of-the-art localization algorithms namely *AMCL* [11] and *UKF Localization* (Unscented Kalman Filter based Localization) [15]. Table 1 shows the absolute pose error (APE) in the global coordinates of each algorithm. We test each algorithm in six different environments described as D_i . It is to be noted that because of the introduction of the topological factor, our algorithm is able to localize and estimate the starting point of the robot in every scenario, whereas *AMCL* fails to localize the robot in multiple scenarios.

Table 1. Absolute Pose Error (APE) [m] for several LiDAR-based localization baselines and our *S-Graph Localization*. Datasets have been recorded in simulated environments. ‘–’ stands for localization failure.

Method	APE [m] ↓					
	Datasets					
	D_1	D_2	D_3	D_4	D_5	D_6
AMCL [11]	2.04	1.71	2.03	2.01	–	–
UKFL [15]	0.97	0.78	–	0.74	0.70	0.88
<i>S-Graph Localization (ours)</i>	0.28	0.15	0.20	0.24	0.29	0.22

Figures 5–10 show the estimated trajectories in dataset D_1, D_2, D_3, D_4, D_5 and D_6 , respectively, by all the baseline algorithms and our proposed method.

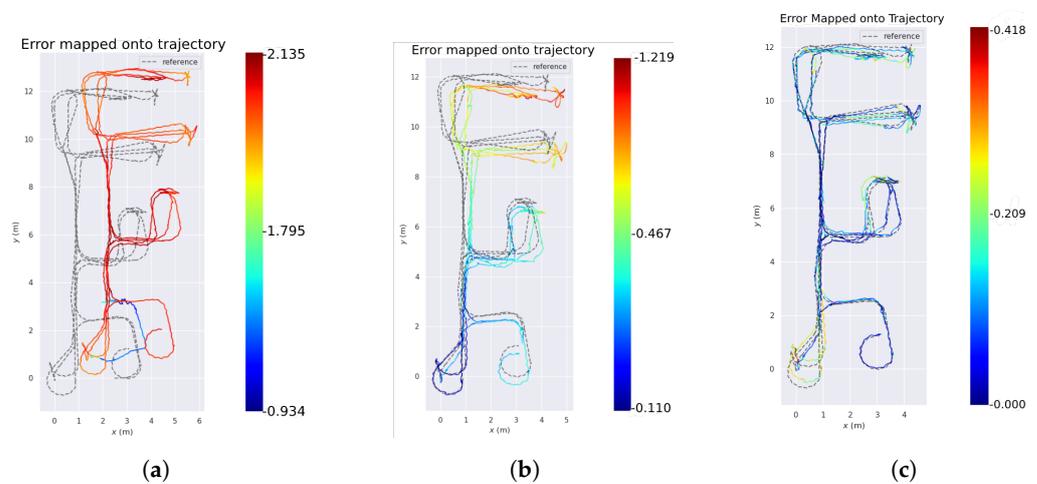


Figure 5. Top view of estimated trajectories for all baselines and our *S-Graph Localization* in the D_1 sequence of our simulated data. The dotted line shows the ground truth trajectory. Our *S-Graph Localization* presents the lowest errors followed by UKF localization. (a) AMCL (D_1); (b) UKFL (D_1); (c) *S-Graph Localization* (D_1).

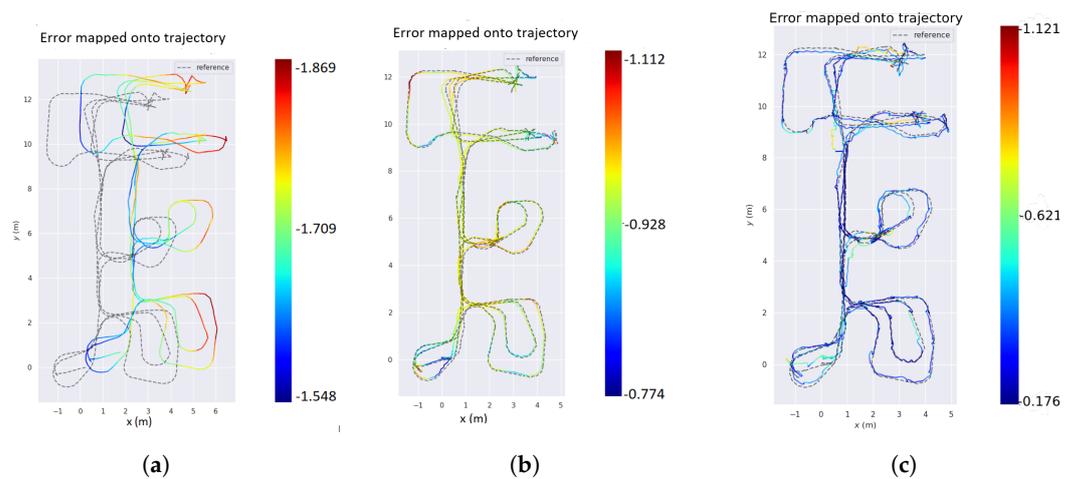


Figure 6. Top view of estimated trajectories for all baselines and our *S-Graph Localization* in the D_2 sequence of our simulated data. The dotted line shows the ground truth trajectory. Our *S-Graph Localization* presents the lowest errors, followed by UKF localization. (a) AMCL (D_2); (b) UKFL (D_2); (c) *S-Graph Localization* (D_2).

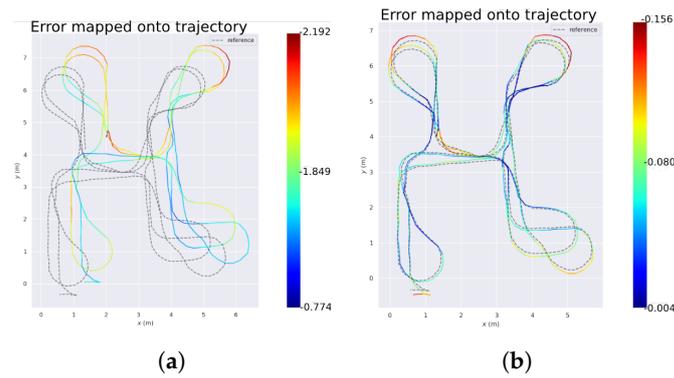


Figure 7. Top view of estimated trajectories for our *S-Graph Localization* and AMCL in the D_3 sequence of our simulated data. The dotted line shows the ground truth trajectory. UKF localization failed to localize in this dataset. (a) AMCL (D_3); (b) *S-Graph Localization* (D_3).

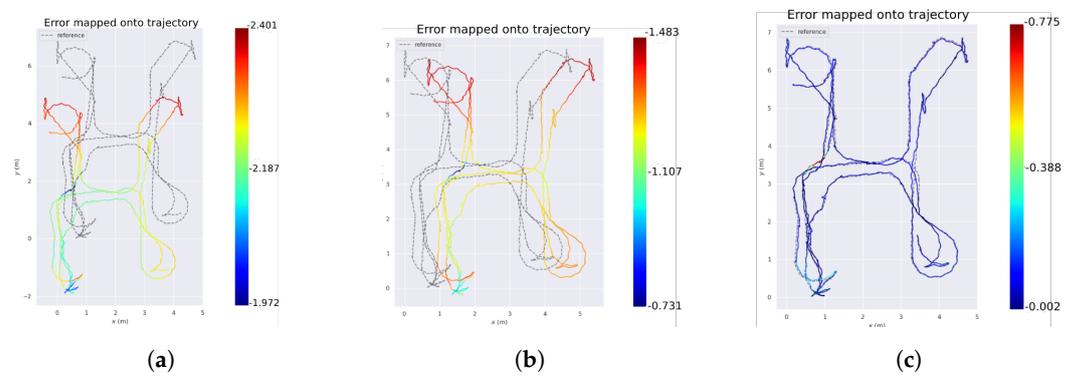


Figure 8. Top view of estimated trajectories for all baselines and our *S-Graph Localization* in the D_4 sequence of our simulated data. The dotted line shows the ground truth trajectory. Our *S-Graph Localization* presents the lowest errors, followed by UKF localization. (a) AMCL (D_4); (b) UKFL (D_4); (c) *S-Graph Localization* (D_4).

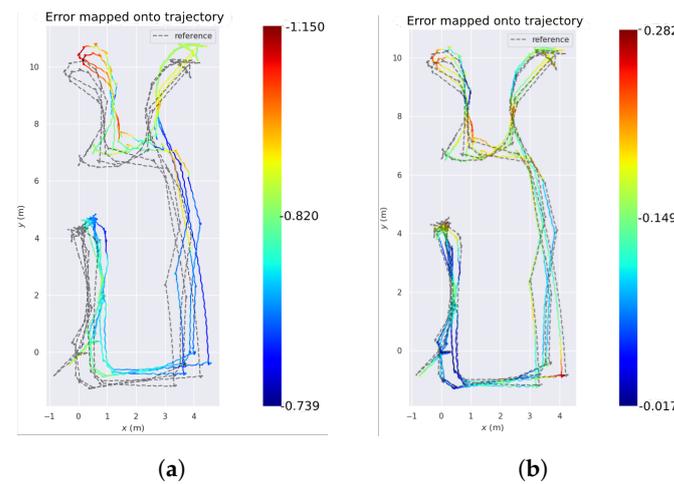


Figure 9. Top view of estimated trajectories for Our *S-Graph Localization* and UKF localization in the D_5 sequence of our simulated data. The dotted line shows the ground truth trajectory. AMCL failed to localize in this dataset. (a) UKFL (D_5); (b) *S-Graph Localization* (D_5).

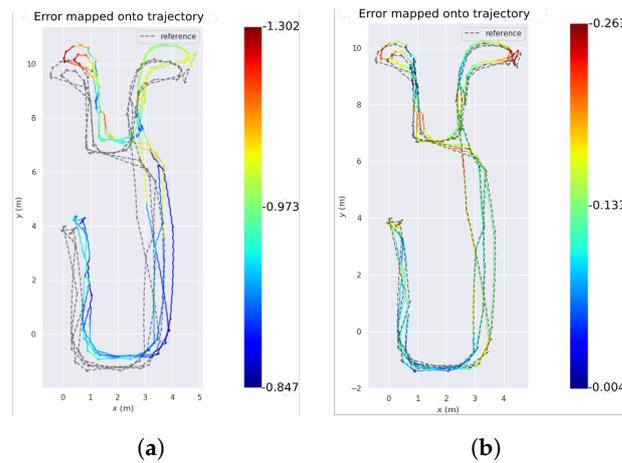


Figure 10. Top view of estimated trajectories for Our *S-Graph Localization* and UKF localization in the D_6 sequence of our simulated data. The dotted line shows the ground truth trajectory. AMCL failed to localize in this dataset. (a) UKFL (D_6); (b) *S-Graph Localization* (D_6).

Furthermore, we have also analyzed the success rate of our algorithm. We performed the same experiment 10 times for each of the 6 simulated datasets (D_1 – D_6) to estimate the success rate for every dataset. Table 2 shows the success percentage of our algorithm. It is to be noted that D_3 D_4 D_5 & D_6 are highly symmetrical as shown in Figure 4; therefore, our method struggles to localize in such environments.

Table 2. Localization success rate of our method in our simulated datasets. Every experiment was performed 10 times for each dataset. Note that D_3 D_4 D_5 & D_6 has a lower success rate because of symmetry.

Method	Convergence Rate [%]					
	Datasets					
	D_1	D_2	D_3	D_4	D_5	D_6
AMCL [11]	100	100	50	40	–	–
UKFL [15]	100	100	–	70	40	70
<i>S-Graph Localization (ours)</i>	100	100	60	50	50	80

We also investigate the time it takes to estimate the initial transformation of our algorithm and compare it to other baseline algorithms. Table 3 shows the convergence time in seconds of each algorithm. It is to be noted that this is not the average time of 10 successful experiments but only of a single successful experiment.

Table 3. Convergence Time [s] for several LiDAR-based localization baselines and our *S-Graph Localization* in simulated environments. ‘–’ stands for localization failure.

Method	Time [s] ↓					
	Datasets					
	D_1	D_2	D_3	D_4	D_5	D_6
AMCL [11]	22	27	76	69	–	–
UKFL [15]	16	25	–	83	88	71
<i>S-Graph Localization (ours)</i>	16	23	72	69	73	65

4.2.2. Real-World Experiments

We performed experiments on three actual construction sites to estimate if our approach works on real data sets. The legged robot we used in our experiments was equipped

with encoders, and we used these encoders to estimate the robot's odometry. On real datasets, we performed the experiments in one room per construction site.

We tested our algorithm on real data which was collected on actual construction sites of the corresponding buildings. Figure 11 shows the environmental setup in a real-world construction site where the experiments were performed. It is to be noted that it was an ongoing construction project therefore only walls and floors were constructed at that time. Table 4 shows the root mean square error between the maps estimated after using different localization algorithms, and, the ground truth of that environment. The ground truth is obtained by extracting the mesh of the environment from the Revit model using BimVision software which is then converted to a point cloud. *AMCL* was unable to localize in some real datasets because the initial estimate was not precise. Moreover, LiDAR measurements in real environments are quite noisy, and *AMCL* only uses 2D measurements. Since both *UKF Localization* and our approach use 3D information, they were able to localize successfully. However, our proposed localization method showed promising results compared to the other two baselines.

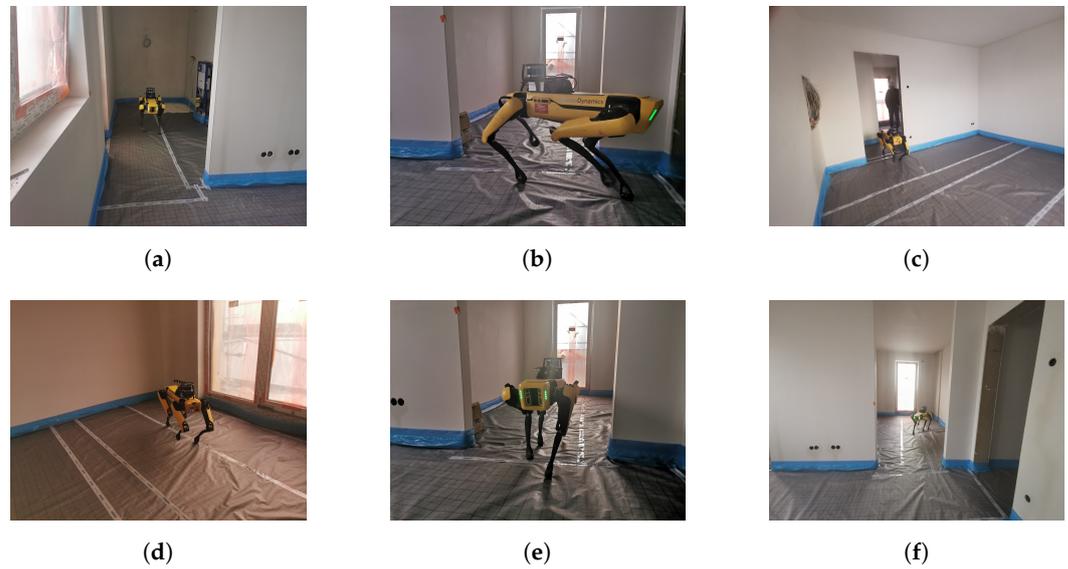


Figure 11. Snapshots of testing of our algorithm in the real-world construction site with a legged robot. (a–c) show the robot being tested in environment *BM-1* and (d–f) show the robot being tested in environment *BM-2* as described in Table 4.

Table 4. Point cloud alignment error [m] on the real datasets. The best results are boldfaced, second best are underlined.

Method	Alignment Error [m] ↓		
	Datasets		
	BM-1	BM-2	BM-3
AMCL [11]	0.98	–	–
UKFL [15]	<u>0.43</u>	<u>0.27</u>	<u>1.03</u>
<i>S-Graph Localization (ours)</i>	0.285	0.25	0.99

4.2.3. Ablation Study

Table 5 presents the convergence rate of our algorithm in simulated datasets without the topological factor. The advantage of adding the topological factor is evident as the convergence time significantly increases in D_3 , D_4 , D_5 , & D_6 which have high symmetry.

Table 5. Convergence Time [s] for our *S-Graph Localization* in simulated environments without the Topological factor. The convergence time increases without the topological factor.

Method	Time [s] ↓					
	Datasets					
	D_1	D_2	D_3	D_4	D_5	D_6
<i>S-Graph Localization (ours)</i>	27	33	97	93	115	103

Moreover, removing the topological factor also results in a reduction of the convergence rate of our algorithm. With the removal of the topological factor, our algorithm only uses plane detections to localize which results in a high convergence rate because of the ambiguity. It can be seen in Table 6 that the convergence rate is reduced by almost 50% in most of the datasets.

Table 6. Localization success rate of our method without topological factor in our simulated datasets. Every experiment was performed 10 times for each dataset.

Method	Convergence Rate [%]					
	Datasets					
	D_1	D_2	D_3	D_4	D_5	D_6
<i>S-Graph Localization (ours)</i>	50	50	30	30	20	40

We also inquired about the effect of removing the topological factor on the absolute pose error, estimated by our algorithm. By looking at Table 7 we can claim that removing the topological factor had little to no effect on the pose estimated by our algorithm.

Table 7. Absolute Pose Error (APE) [m] without topological factor in our simulated datasets.

Method	APE [m] ↓					
	Datasets					
	D_1	D_2	D_3	D_4	D_5	D_6
<i>S-Graph Localization (ours)</i>	0.30	0.16	0.20	0.24	0.31	0.23

4.2.4. Limitations

Our proposed localization method shows encouraging results on both simulated as well as real-world data when compared to the state-of-the-art. However, the current version of our approach does have some limitations. First, our approach struggles if the environment has irregularly shaped rooms. Currently, our *S-Graphs* is capable of only detecting rooms with four walls i.e., a rectangular shape. In the future, we plan to incorporate rooms of different shapes. Second, our approach also finds it difficult to localize if the starting orientation of the robot is different than the orientation of the Revit frame. This is because there is symmetry in the environment, our method finds it difficult to localize if the starting orientation is different than the Revit frame. Unlike AMCL which uses an entire occupancy grid map of the environment to localize, our method only uses plane normals and distance values of rooms which makes it difficult to counter the orientation errors. We plan to address this issue in future work. It is to be noted that if two identical rooms are present in the environment or the environment is highly symmetrical, the algorithm faces difficulty in localization. In this case, the robot needs to further explore the environment. Moreover, if there are objects obstructing the walls, the robot will not be able to detect the wall and consequently the room. This will mean that the algorithm will not be able to take advantage of the topological factor and will eventually perform as a basic particle filter localization algorithm.

5. Conclusions

In this paper, we present our work for the localization of mobile robots in construction environments integrating BIM with scene graphs. We extract useful geometric, semantic as well as topological information from the readily available BIM plans to integrate it within a three-layered hierarchical (*S-Graphs*) [2], by generating the metric-semantic and topological layer for a given environment. As the robot navigates in the environment with an on-board 3D LiDAR, our novel implementation of the particle filter utilizing geometric, semantic, and topological information rapidly estimates the initial guess of the robot pose in the environment, after which the *S-Graphs* starts creating the robot tracking layer appropriately connecting it with the previously generated metric-semantic and topological layers. We tested our approach in both simulated and real construction environments comparing it with traditional metric-based localization algorithms and achieving state-of-the-art results. In the future, we plan to improve the information extraction from the BIM as well as localize the robot irrespective of whether it starts in a room.

Author Contributions: Conceptualization of this work was done by M.S. and H.B. The methodology was designed by M.S. and H.B. Software was made by M.S. and H.B. Validation was done by M.S. A formal analysis was done by M.S. The investigation was done by M.S. The initial draft of the paper was made by M.S. which was edited and reviewed by H.B. The work was supervised by J.L.S.-L. and H.V. who also administrated the project. Funding was acquired by H.V. and J.L.S.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Fonds National de la Recherche of Luxembourg (FNR), under the projects C19/IS/13713801/5G-Sky, by a partnership between the Interdisciplinary Center for Security Reliability and Trust (SnT) of the University of Luxembourg and Stugalux Construction S.A.

Data Availability Statement: According to the non-disclosure agreement signed between our funding partner and us, the data provided by the company is private and cannot be shared due to privacy and ethical reasons.

Conflicts of Interest: BIM models, and access to corresponding real construction sites for experimentation, were provided by our funding partner *Stugalux Construction SA*.

References

1. Azhar, S.; Hein, M.; Sketo, B. Building information modeling (BIM): Benefits, risks and challenges. In Proceedings of the 44th ASC Annual Conference, Halifax, NS, Canada, 22–26 September 2008; pp. 2–5.
2. Bavle, H.; Sanchez-Lopez, J.L.; Shaheer, M.; Civera, J.; Voos, H. Situational Graphs for Robot Navigation in Structured Indoor Environments. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9107–9114. [[CrossRef](#)]
3. Desrochers, B.; Lacroix, S.; Jaulin, L. Set-membership approach to the kidnapped robot problem. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3715–3720.
4. Kendall, A.; Cipolla, R. Modelling uncertainty in deep learning for camera relocalization. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4762–4769.
5. Lowry, S.; Sünderhauf, N.; Newman, P.; Leonard, J.J.; Cox, D.; Corke, P.; Milford, M.J. Visual place recognition: A survey. *IEEE Trans. Robot.* **2015**, *32*, 1–19. [[CrossRef](#)]
6. Miller, I.D.; Cowley, A.; Konkimalla, R.; Shivakumar, S.S.; Nguyen, T.; Smith, T.; Taylor, C.J.; Kumar, V. Any way you look at it: Semantic crossview localization and mapping with lidar. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2397–2404. [[CrossRef](#)]
7. Leung, K.Y.K.; Clark, C.M.; Huissoon, J.P. Localization in urban environments by matching ground level video images with an aerial image. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 551–556.
8. Zuo, X.; Ye, W.; Yang, Y.; Zheng, R.; Vidal-Calleja, T.; Huang, G.; Liu, Y. Multimodal localization: Stereo over LiDAR map. *J. Field Robot.* **2020**, *37*, 1003–1026. [[CrossRef](#)]
9. Fox, D.; Burgard, W.; Thrun, S. Markov Localization for Reliable Robot Navigation and People Detection. *J. Artif. Intell. Res.* **1999**, *11*, 391–427. [[CrossRef](#)]
10. Dellaert, F.; Fox, D.; Thrun, S.; Burgard, W. Monte Carlo Localization for Mobile Robots. 1999. Available online: https://www.ri.cmu.edu/pub_files/pub1/dellaert_frank_1999_2/dellaert_frank_1999_2.pdf (accessed on 28 February 2023).
11. Fox, D. KLD-sampling: Adaptive particle filters. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December 2001; Volume 14.

12. Röwekämper, J.; Sprunk, C.; Tipaldi, G.D.; Stachniss, C.; Pfaff, P.; Burgard, W. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 1999.
13. Thrun, S.; Fox, D.; Burgard, W. Monte Carlo localization with mixture proposal distribution. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, Austin, TX, USA, 30 July–3 August 2000; pp. 859–865.
14. Saarinen, J.; Andreasson, H.; Stoyanov, T.; Lilienthal, A.J. Normal Distributions Transform Monte-Carlo Localization (NDT-MCL). In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013.
15. Koide, K.; Miura, J.; Menegatti, E. A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419841532. [[CrossRef](#)]
16. Rusu, R.; Blodow, N.; Beetz, M. Fast point feature histograms (fpfh) for 3d registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.
17. Tombari, F.; Salti, S.; Stefano, L.D. Unique signatures of histograms for local surface description. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010.
18. Steder, B.; Ruhnke, M.; Grzonka, S.; Burgard, W. Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011.
19. Rhling, T.; Mack, J.; Schulz, D. Fast histogram-based similarity measure for detecting loop closures in 3-d lidar data. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015.
20. He, L.; Wang, X.; Zhang, H. M2dp: A novel 3d point cloud descriptor and its application in loop closure detection. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016.
21. Dubé, R.; Cramariuc, A.; Dugas, D.; Sommer, H.; Dymczyk, M.; Nieto, J.; Siegwart, R.; Cadena, C. SegMap: Segment-based mapping and localization using data-driven descriptors. *Int. J. Robot. Res.* **2020**, *39*, 339–355. [[CrossRef](#)]
22. Park, Y.S.; Kim, J.; Kim, A. Radar Localization and Mapping for Indoor Disaster Environments via Multi-modal Registration to Prior LiDAR Map. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1307–1314. [[CrossRef](#)]
23. Li, Z.; Ang, M.H.; Rus, D. Online Localization with Imprecise Floor Space Maps using Stochastic Gradient Descent. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 8571–8578. [[CrossRef](#)]
24. Boniardi, F.; Valada, A.; Mohan, R.; Caselitz, T.; Burgard, W. Robot Localization in Floor Plans Using a Room Layout Edge Extraction Network. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019. [[CrossRef](#)]
25. Cui, L.; Rong, C.; Huang, J.; Rosendo, A.; Kneip, L. Monte-Carlo Localization in Underground Parking Lots using Parking Slot Numbers. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 2267–2274. [[CrossRef](#)]
26. Zimmerman, N.; Wiesmann, L.; Guadagnino, T.; Läbe, T.; Behley, J.; Stachniss, C. Robust Onboard Localization in Changing Environments Exploiting Text Spotting. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022. [[CrossRef](#)]
27. Siemiatkowska, B.; Harasymowicz-Boggio, B.; Przybylski, M.; Rozanska-Walczyk, M.; Wisniowski, M.; Kowalski, M. BIM Based Indoor Navigation System of Hermes Mobile Robot. In *Proceedings of the 19th CISM-Iftomm Symposium*; Springer: Berlin/Heidelberg, Germany, 2013.
28. Park, J.; Cho, Y.K.; Martinez, D. A BIM and UWB integrated mobile robot navigation system for indoor position tracking applications. *J. Constr. Eng. Proj. Manag.* **2016**, *6*, 30–39. [[CrossRef](#)]
29. Xu, M.; Wei, S.; Zlatanova, S.; Zhang, R. BIM-based indoor path planning considering obstacles. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 417. [[CrossRef](#)]
30. Asadi, K.; Ramshankar, H.; Noghabaei, M.; Han, K. Real-time image localization and registration with BIM using perspective alignment for indoor monitoring of construction. *J. Comput. Civ. Eng.* **2019**, *33*, 04019031. [[CrossRef](#)]
31. Moura, M.S.; Rizzo, C.; Serrano, D. BIM-based Localization and Mapping for Mobile Robots in Construction. In Proceedings of the 2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Santa Maria da Feira, Portugal, 28–29 April 2021; pp. 12–18. [[CrossRef](#)]
32. Yin, H.; Liew, J.M.; Lee, W.L.; Ang, M.H.; Yeoh, K.W.-J. Towards BIM-based robot localization: A real-world case study. In Proceedings of the 39th International Symposium on Automation and Robotics in Construction, Bogota, Colombia, 12–15 July 2022. [[CrossRef](#)]
33. Hendriks, R.W.M.; Pauwels, P.; Torta, E.; Bruyninckx, H.J.; van de Molengraft, M.J.G. Connecting Semantic Building Information Models and Robotics: An application to 2D LiDAR-based localization. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11654–11660. [[CrossRef](#)]

34. Kim, U.; Park, J.; Song, T.; Kim, J. 3-D Scene Graph: A Sparse and Semantic Representation of Physical Environments for Intelligent Agents. *IEEE Trans. Cybern.* **2019**, *50*, 4921–4933. [[CrossRef](#)] [[PubMed](#)]
35. Armeni, I.; He, Z.Y.; Gwak, J.; Zamir, A.R.; Fischer, M.; Malik, J.; Savarese, S. 3D Scene Graph: A structure for unified semantics, 3D space, and camera. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5664–5673.
36. Wu, S.C.; Wald, J.; Tateno, K.; Navab, N.; Tombari, F. SceneGraphFusion: Incremental 3D Scene Graph Prediction from RGB-D Sequences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021. [[CrossRef](#)]
37. Rosinol, A.; Gupta, A.; Abate, M.; Shi, J.; Carlone, L. 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv* **2020**, arXiv:2002.06289.
38. Hughes, N.; Chang, Y.; Carlone, L. Hydra: A Real-time Spatial Perception Engine for 3D Scene Graph Construction and Optimization. *arXiv* **2022**, arXiv:2201.13360.
39. Qin, C.; Zhang, Y.; Liu, Y.; Lv, G. Semantic loop closure detection based on graph matching in multi-objects scenes. *J. Vis. Commun. Image Represent.* **2021**, *76*, 103072. [[CrossRef](#)]
40. Gawel, A.; Del Don, C.; Siegwart, R.; Nieto, J.; Cadena, C. X-View: Graph-Based Semantic Multi-View Localization. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1687–1694. [[CrossRef](#)]
41. Guo, X.; Hu, J.; Chen, J.; Deng, F.; Lam, T.L. Semantic Histogram Based Graph Matching for Real-Time Multi-Robot Global Localization in Large Scale Environment. *IEEE Robot. Autom. Lett.* **2021**, *6*, 8349–8356. [[CrossRef](#)]
42. Wang, F.; Zhang, C.; Tang, F.; Jiang, H.; Wu, Y.; Liu, Y. Lightweight Object-level Topological Semantic Mapping and Long-term Global Localization based on Graph Matching. *arXiv* **2022**, arXiv:2201.05977. [[CrossRef](#)]
43. Liu, Y.; Petillot, Y.; Lane, D.; Wang, S. Global Localization with Object-Level Semantics and Topology. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4909–4915. [[CrossRef](#)]
44. Thein, V. Industry foundation classes (IFC). In *BIM Interoperability through a Vendor-Independent File Format*; Bentley: Crewe, UK, 2011; p. 152.
45. Zhou, L.; Koppel, D.; Kaess, M. LiDAR SLAM With Plane Adjustment for Indoor Environment. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7073–7080. [[CrossRef](#)]
46. Geneva, P.; Eckenhoff, K.; Yang, Y.; Huang, G. LIPS: LiDAR-Inertial 3D Plane SLAM. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 123–130. [[CrossRef](#)]
47. Quigley, M. ROS: An open-source Robot Operating System. In Proceedings of the International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.