IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Long-Term Localization using Semantic Cues in Floor Plan Maps

Nicky Zimmerman Tiziano Guadagnino Xieyuanli Chen Jens Behley Cyrill Stachniss

***Abstract*—Lifelong localization in a given map is an essential capability for autonomous service robots. In this paper, we consider the task of long-term localization in a changing indoor environment given sparse CAD floor plans. The commonly used pre-built maps from the robot sensors may increase the cost and time of deployment. Furthermore, their detailed nature requires that they are updated when significant changes occur. We address the difficulty of localization when the correspondence between the map and the observations is low due to the sparsity of the CAD map and the changing environment. To overcome both challenges, we propose to exploit semantic cues that are commonly present in human-oriented spaces. These semantic cues can be detected using RGB cameras by utilizing object detection, and are matched against an easy-to-update, abstract semantic map. The semantic information is integrated into a Monte Carlo localization framework using a particle filter that operates on 2D LiDAR scans and camera data. We provide a long-term localization solution and a semantic map format, for environments that undergo changes to their interior structure and detailed geometric maps are not available. We evaluate our localization framework on multiple challenging indoor scenarios in an office environment, taken weeks apart. The experiments suggest that our approach is robust to structural changes and can run on an onboard computer. We released the open source implementation[1] of our approach written in C++ together with a ROS wrapper.**

***Index Terms*—Localization, Semantic Scene Understanding**

I. Introduction

To operate autonomously in indoor environments, such as factories or offices, mobile robots must be able to determine their pose. For localization in a given map, there are two challenges: the changing nature of human-occupied environment and the quality of available maps. Precise, highly-detailed maps are an accurate representation of the environment only at the time they were captured, and they become outdated in the presence of "quasi-static" changes such as moving furniture, clutter, opening and closing doors. We describe "quasi-static" changes as long-lasting alterations (hours, days, weeks) that cause deviation between sensor observations and the given map, in contrast to dynamics such as humans and fast-moving objects. The availability of feature-rich, dense maps is not guaranteed and construction of such maps can be costly. Therefore, autonomous robots benefit from localizing in sparse maps such as floor plans or hand-crafted room layouts as they are seldom affected by changes. Architectural drawings are familiar to inexpert users and can be easily updated with CAD software. As they capture persistent structures, they typically do not require updates. However, using these sparse maps is challenging due to the paramount discrepancies between the robot's observations of the environment and the information depicted in the maps. Additionally, floor plans lack geometric information necessary to localize in a highly repetitive indoor environment, as can be seen in Fig. 1.

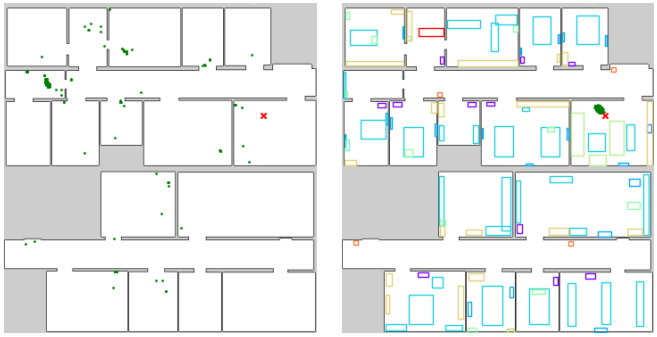

Fig. 1: Floor plan maps include high degree of symmetry and low similarity to actual LiDAR measurements. This leads to multiple hypotheses that cannot be resolved correctly. We propose integrating semantic cues from a high level, abstract semantic map to assist with global localization. The red cross indicates the ground truth pose and the green dots are the particles. Left: 2D LiDAR MCL with multiple hypotheses. Right: Convergence to a single hypothesis when exploiting semantic cues, in an abstract semantic maps including various objects (colored rectangles).

Additional sources of information can be used to overcome the challenges of global localization, and such cues have been frequently used by researchers to improve robot localization. For example, WiFi, an extremely prevalent utility, can aid in pose estimation by considering the signal strength [14]. Textual information, constantly used by humans to navigate, is readily available in human-occupied environments. However, very few works consider textual cues for localization [7][28][43].

Another avenue is exploiting semantic information. The last decade was marked by significant advances in object detection [2][41] and semantic segmentation [12][32], where semantic cues can be efficiently inferred from images (with some fine-tuning). The most commonly used map representation for robotics is an occupancy grid map [24]. However,

Manuscript received: July 6, 2022; Revised: Oct 5, 2022; Accepted: Nov 2, 2022. This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments.

This work has partially been funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony).

All authors are with the University of Bonn, Germany. Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK. Corresponding author: zimmerman@igg.uni-bonn.de

Digital Object Identifier (DOI): see top of this page.

[1]https://github.com/PRBonn/hsmcl

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

human environments tend to be object-centric, and humans do not require precise metric information in order to navigate them [21][39]. Rather, humans rely on a small number of specific landmarks, and associate places with the objects present there. For this reason, we consider localization in a sparse, approximate map, that does not require an elaborate map acquisition process. No work on semantic localization in sparse maps with abstract and hierarchical semantic information exists to our knowledge.

The main contributions of this paper is a global localization system in floor plan maps that integrates semantic cues. We propose to leverage semantic cues to break the symmetry and distinguish between locations that appear similar or identical in the nondescript maps. Semantic information is commonly available in the form of furniture, machinery and textual cues and can be used to distinguish between spaces with similar layout. To avoid the complexity of building a 3D map from scans and to enable easy updates to semantic information, we present a 2D, high level semantic map. Thus, we present a format for abstract semantic map with an editing application and a sensor model for semantic information that complements LiDAR-based observation models. Additionally, we provide a way to incorporate hierarchical semantic information. Unlike most modern semantic-based SLAM approaches [6][20][31][37][38], our approach does not require a GPU and can run online on an onboard computer. Like semantic visual SLAM methods, we also rely on semantic information, but while SLAM approaches construct a map online, we focus on localization in a given map. In our experiments, we show that our approach is able to: (i) localize in sparse floor plan-like map with high symmetry using semantic cues, (ii) localize long-term without updating the map, (iii) localize in previously unseen environment. (iv) localize the robot online using an onboard computer. These claims are backed up by the paper and our experimental evaluation.

II. Related Work

Localization in 2D maps has been thoroughly researched [5][35][36][40]. Among the most robust and commonly-used approaches, are the probabilistic methods for pose estimation, including Markov localization by Fox et al. [11], the extended Kalman filter (EKF) [16] and particle filters, also known as Monte Carlo localization (MCL) by Dellaert et al. [8]. These works laid the foundation for localization using range sensors and cameras.

Localization in detailed, feature-rich maps, usually constructed by range sensors, is extensively-studied [23], but few works address the problem of localization in sparse, floor plan-like maps, despite their benefits. Floor plans are readily-available in many facilities, and therefore do not depend on prior mapping. As they only include information on permanent structures, they do not require frequent updates when objects, such as furniture, are relocated. Their main drawback comes from their sparse nature, and the lack of detailed geometric information can results in global localization failures when multiple rooms look alike. Another concern is the possible mismatch between the floor plans and the constructed building [3]. Li et al. [17] address the scale difference between constructed structure and floor plans by introducing a new state variable. Boniardi et al. [4] uses cameras to infer the room layout via edge extraction and match it against the floor plan. In the evaluation, the authors initialized the pose within 10 cm and 15° from the ground-truth pose, and did not evaluate global localization. We speculate that edge extraction of the walls is not sufficient in a highly repetitive indoor environment where many rooms have the same size. Both approaches provide tracking capabilities, but not global localization.

Recent works in extracting semantic information with deep learning models showed significant improvement in performance for both text spotting [18][33] and object detection [2][41]. The use of textual cues for localization is surprisingly uncommon, with notable works by Cui et al. [7] and Zimmerman et al. [43]. Both works considered using textual information within an MCL framework, but used different approaches to integrate it. In our approach, we expand our previous work [43] to consider semantic cues via object detection, not only textual ones.

The use of semantic information for localization and place recognition is applied to a variety of sensors, including 2D and 3D LiDARs, RGB and RGB-D cameras. Rottmann et al. [30] use AdaBoost features from 2D LiDAR scans to infer semantic labels such as office, corridor and kitchen. They combine the semantic information with occupancy grid map in an MCL framework. Unlike our approach, their method requires a detailed map and manually assigning a semantic label to every grid cell. Hendrikx et al. [13] utilize available building information model to extract both geometric and semantic information, and localize by matching 2D LiDAR-based features corresponding to walls, corners and columns. While the automatic extraction of semantic and geometric maps from a BIM is promising, the approach is not suitable for global localization as it cannot overcome the challenges of a repetitively-structured environment.

Atanasov et al. [1] treat semantic objects as landmarks that include their 3D pose, semantic label and possible shape priors. They detect objects using a deformable part model [9], and use their semantic observation model in an MCL framework. The results they report do not outperform LiDAR-based localization. An alternative representation for semantic information is a constellation model, as suggested by Ranganathan et al. [29]. In their approach, they use stereo cameras, exploiting depth information. They rely on hand-crafted features including SIFT [19] to detect objects. Places are associated with constellations of objects, where every object has shape and appearance distribution and a relative transformation to the base location. Unlike these two approaches, our approach does not require exact poses for the semantic objects. A more flexible representation is proposed by Yi et al. [39], who use topological-semantic graphs to represent the environment. They extract topological nodes from an occupancy grid map, and characterize each node by the semantic objects in its vicinity. It suffers when objects are far from the camera and can easily diverge when objects cannot be detected, while our approach is more robust as it relied additionally on LiDAR observations and textual cues. Similarly to the above mentioned approaches, we also use

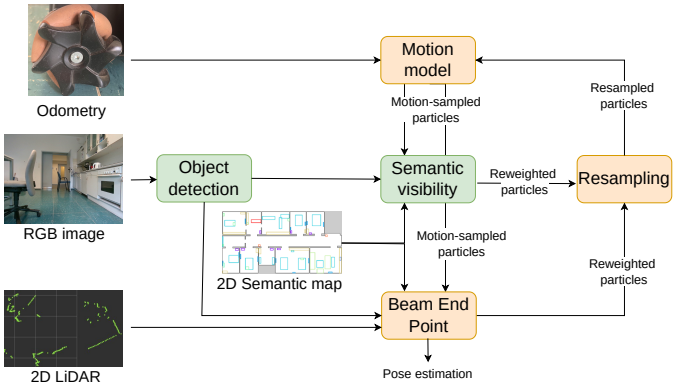

Fig. 2: A simplified overivew of the online localization approach. Given RGB images, 2D LiDAR scans an odoemtry input, we integrate semantic cues into an MCL framework.

sparse representation for semantic objects. However, by using deep learning to detect objects, we are able to detect a larger variety of objects with greater confidence, and localize in previously unseen places.

Sünderhauf et al. [34] construct semantic maps from camera by assigning a place category to each occupancy grid cell. They use the Places205 ConvNet [42] to recognize places, and rely on a LiDAR-based SLAM for building the occupancy grid map. The limitation of their approach is in the high level of semantic abstraction. As their work relied on coarse room categorization, it might not be sufficient for global localization in highly repetitive environments.

III. Our Approach

Our goal is to globally localize in an indoor environment represented by a nondescript floor plan and a high level semantic map. As sensors for localization, we use 2D LiDAR, cameras and wheel odometry. We build our localization approach on the Monte Carlo localization (MCL) framework [8]. To distinguish between locations that appear similar or identical in the sparse maps, we introduce imprecise, high-level semantic maps in Sec. III-B and a sensor model for semantic similarity in Sec. III-D. The integration of the semantic information in the MCL framework is introduced in Sec. III-E. In addition, we perform an analysis to determine the stability of semantic classes as discussed in Sec. III-G and utilize the semantic information to discard LiDAR measurements resulting from dynamic objects. Furthermore, in Sec. III-G we explore a hierarchical semantic approach for inferring the room type based on objection detection, and exploit this information to initialize the particle filter. An overview of the approach is illustrated in Fig. 2.

A. Monte Carlo Localization

Monte Carlo localization [8] is a particle filter-based approach for state estimation given a map m and sensor readings z_t at time t. As we localize in floor plan maps, the robot's state x_t is defined by the 2D coordinates $(x, y)^\top$ and the orientation $\theta \in [0, 2\pi)$. The map m is represented by an occupancy grid map [24] or an abstract semantic map, see Sec. III-B, and

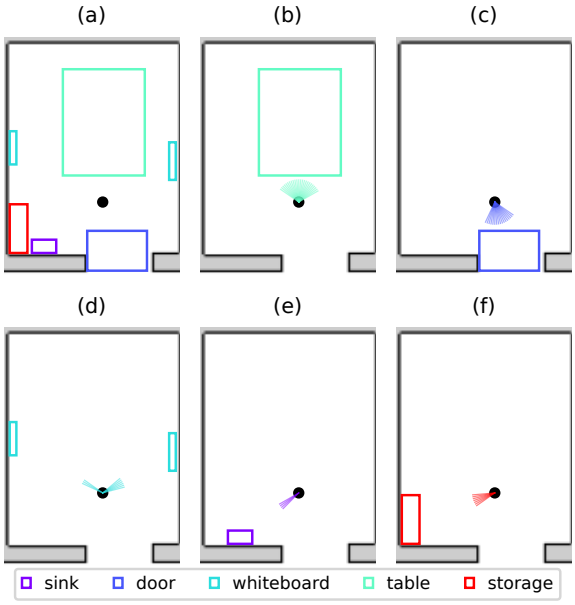

Fig. 3: A visualization of the semantic visibility concept. (a) A semantic map of a single room, with a query point (black dot). (b)-(f) The bearings in which each semantic class objects are visible from the query point.

the observation z is composed of K elements z_k. We apply a recursive Bayesian update to a set of particles $\mathcal{S}_t$, which represent the belief about the robot's pose, $p(x_t \mid z_{1:t}, m)$. Every particle is represented by a state $x_t^{(i)}$ and a weight $w_t^{(i)}$. The proposal distribution $p(x_t \mid x_{t-1}, u_t)$ is sampled when a new motion prior u_t is available, using a motion model for holonomic robots with odometry noise $\sigma_{\text{odom}} \in \mathbb{R}^3$. By computing the likelihood of an observation z_t given a robot's state x_t using the observation model $p(z_t \mid x_t^{(i)}, m)$, an individual importance weight $w_t^{(i)}$ is assigned to each particle. In the resampling step, we use low-variance resampling [35].

B. High-Level Semantic Maps

We represent our prior information about semantics with a 2D high level semantic map, where semantic objects are represented by a semantic class label l and a rectangle overlying the occupancy grid map. see Fig. 3. The size of the rectangle does not have to be very accurate and the location where it is marked can be a rough estimate of its actual placement. In our abstract map, objects differed from their actual size by 62.5%, or up to 1.25 m. This imprecise representation of semantic information is both generic enough to address variety of objects and simple enough to allow editing by end users. Each room can be assigned a name, corresponding to a text sign, and a room category representing a higher level of semantic understanding compared to basic object detection. The semantic maps can be easily created and edited using the GUI application MAPhisto[2].

[2]https://github.com/FullMetalNicky/Maphisto

C. Beam End Point Model

The beam end point model [35] $p_L(z_t \mid x_t, m)$ is an observation model for range sensors

$$p_L(z_t^k \mid x_t, m_L) = \frac{1}{\sqrt{2\pi\sigma_{\text{obs}}}} \exp\left(-\frac{\text{edt}(\hat{z}_t^k)^2}{2\sigma^2}\right) \quad (1)$$

Where $\hat{z}_t^k$ is the end point of the LiDAR beam in the occupancy grid map m_L and edt is the Euclidean distance transform [10], in which each cell is labeled with the distance to an occupied cell in the occupancy grid map. The edt was truncated at r_{max}, a predefined maximal range.

D. Semantic Visibility Model

The last decade's progress in semantic interpretation allows us to use deep learning models for text spotting [18][33] and object detection [2][41].

Object detection is the task of detecting instances of semantic objects in images and videos. In our approach, the required output from an object detection model is a semantic label, a bounding box and a confidence score for every detected object. For each bounding box in the prediction, we transform it to a 3D cone $\hat{\boldsymbol{x}}$ in the robot coordinate system, see Fig. 4. We take the pixel coordinates of the right and left boundaries of a bounding box, (bb_r, bb_l) and project them to 3D rays by using the camera's intrinsics and extrinsics matrices. For a pixel $v = (x, y, 1)^T$, we define the associated 3D ray $V(\lambda)$ as follows:

$$V(\lambda) = O + \lambda R^{-1} K^{-1} v, \quad (2)$$

where $K \in \mathbb{R}^{3\times3}$ is the camera intrinsics, $R \in \mathbb{R}^{3\times3}$ is the camera rotation and $O \in \mathbb{R}^3$ is camera center.

From the high-level semantic information, we construct visibility maps for the semantic classes. For each valid, free space cell c in the occupancy grid map, we compute the visibility of semantic objects. A semantic object o is visible from a grid cell c if we can ray-trace it without crossing a non-valid, i.e., occupied or unknown, cell. For each cell c, we maintain a list of all visible semantic classes. For each semantic class l, we store the set of bearing vectors, $\mathcal{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}, \|\boldsymbol{b}_i\| = 1$, in which objects of class l are visible. This process of constructing the visibility maps is performed once, when the algorithm is launched, and is illustrated in Fig. 3.

A semantic observation y_t includes the set of detected objects. For every object we store its semantic label l, its confidence score f and the center of its cone as the bearing $\hat{\boldsymbol{b}}$. For each particle $s_t^{(i)}$ with pose $x_t^{(i)} = (x, y, \theta)^\top$, we transform the bearing $\hat{\boldsymbol{b}}$ into the world coordinate system. We query the pre-built semantic visibility maps for cell c corresponding to the pose of particle $s_t^{(i)}$, and compare it with the observation. If an object is observed with confidence score f which is lower than a threshold τ, we ignore the observation. Otherwise, if an object with semantic label l is visible from cell c, we compare the observed bearing $\hat{\boldsymbol{b}}$ to the set of possible bearings $\mathcal{B}$ by using the cosine similarity:

$$\text{sim}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\|\boldsymbol{v}_1\|\|\boldsymbol{v}_2\|}. \quad (3)$$

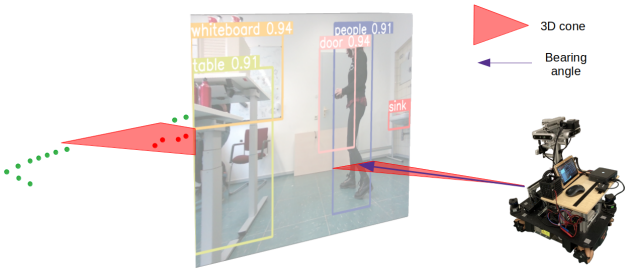

Fig. 4: The bounding box detecting a dynamic class (person) is projected to 3D and used to mask the LiDAR beams that fall within the cone.

where $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^2$. To compare the observed bearing $\hat{\boldsymbol{b}}$ with all possible visible bearings $\boldsymbol{b}_i \in \mathcal{B}$ and select the best match according to the distance d, defined as

$$d = 1 - \max_{\boldsymbol{b}_i \in \mathcal{B}}(\text{sim}(\boldsymbol{b}_i, \hat{\boldsymbol{b}})). \quad (4)$$

For a set of detected objects z_t^S, our observation model is given by

$$p_S(z_t^k \mid x_t, m_S) = \exp(-d) \qquad z_t^k \in z_t^S, \quad (5)$$

where z_t^k is the k^{th} confidently observed object in the set z_t^S, and m_S is the abstract semantic map.

E. Integrating Different Modalities in the MCL Framework

We handle all information sources asynchronously – the motion model is sampled when odometry input is available, and the particles are re-weighted when an observation arrives. We integrate the 2D LiDAR measurements and the object detections using two different observation models. For a 2D LiDAR observation, z_t^L, we use the beam-end model $p_L(z_t^L \mid x_t, m_L)$ described in Sec. III-C. When object detection information arrives, z_t^S, we use the semantic visibility model $p_S(z_t^S \mid x_t, m_S)$, detailed in Sec. III-F.

The product of likelihood model assumes elements of each observation, e.g scan points in a LiDAR scan, are independent of each other. With the high angular resolution of our LiDAR this assumption does not hold. Similarly, for the semantic visibility model, detected objects are not entirely independent of each other as they often belong to the same context. The traditional product of likelihood model tends to be overconfident in such circumstances, leading us to choose the product of experts model [22], which uses geometric mean to compute the weight of each particle

$$p(z_t \mid x_t, m) = \prod_{k=0}^{K} p(z_t^k \mid x_t, m)^{\frac{1}{K}}, \quad (6)$$

where z_t^k is a single component of an observation z_t, be it a LiDAR scan or a set of detected objects. The beam-end model is triggered only when the robot moves more than d_{xy} or rotates more than d_θ, while the semantic observation model is always updated. Based on the semantic stability analysis, we detected semantic classes that tend to move frequently, which we refer to as dynamics. In addition to excluding these classes from the semantic map, we also use these detections to filter out LiDAR measurements that are the result of dynamics, as seen in Fig. 4

F. Semantic Stability Analysis

To decide which semantic classes would benefit localization, we estimated how likely they are to move around. We prepared a semantic map for all detectable classes, and examined the training-dedicated recordings T1-T5 spanning over multiple weeks. As our dataset includes the ground truth pose of the robot using an external reference system, we were able to conclude whether the position of detected objects corresponded to their position in the map. Using Eq. (4), we consider a detected object to correspond to the semantic map if $d < \tau_s$. We calculate the ratio of map-consistent detection per semantic class, and deem a semantic class stable if the ratio was above 0.6. The ratio is computed by dividing the number of map-consistent detected objects of class l, by the total number of detections. Unstable classes are excluded from the semantic visibility model, and then stability scores are given in Tab. I.

G. Hierarchical Semantic Localization

In big indoor environments, a very large number of particles is required to sufficiently cover the area in the initialization phase of global localization, which result in great computational costs. It is possible to reduce the number of used particles and achieve global localization by considering a hierarchy of semantic information. We propose to infer the room category (office, corridor, kitchen, reception) based on the predictions from the object detection. We use a nearest-neighbor classifier [25] to learn a relationship between the detected objects and the room category. We encode the semantic information as a feature vector $r \in \mathbb{R}^M$, where M is the number of classes we are able to detect. Each vector element r_l represents the number of detected objects from a specific semantic label l. We used our initial semantic observations to infer the room category, and initialize the particle filter accordingly, so that particles are only initialized in rooms of the same category. The information about the category of each room is stored in the high-level semantic map (Sec. III-B).

IV. Experimental Evaluation

The focus of this work is to provide an efficient, robust localization approach that exploits semantic information for long-term operation in sparse floor plans. We conducted our experiments to support our claims and show that our approach is able to: (i) localize in sparse floor plan-like map with high symmetry using semantic cues, (ii) localize long-term without updating the map, (iii) localize in previously unseen environment, (iv) localize the robot online using an onboard computer.

A. Experimental Setup

To evaluate the performance of our approach, we recorded a dataset on the first and second floors of our building. Our mobile sensing platform consisted of a Kuka YouBot platform with 2 Hokuyo UTM-30LX LiDAR sensors, wheel encoders, 4 cameras covering jointly a 360° field-of-view, and an upward-looking camera that is used only for evaluation purposes, see Fig. 5. The recordings span across several weeks, capturing different scenarios including moving furniture, opening and closing of doors and humans passing by.

By using precisely localized AprilTags [26], which are densely placed (approx. 1 tag/m^2) on the ceiling of every room and corridor on the second floor, we are able to extract the ground truth pose of the robot from the upwards-looking camera. The camera is used to detect the AprilTags, which allows us to accurately localize the robot even when the environment undergoes changes. The upward-looking camera captured frames at 25 fps, and due to its wide-angle lens, we were able to detect multiple AprilTags in every frame. The pose was extracted in a least-squares fashion using multiple detections. The locations of the AprilTags were obtained using a high resolution terrestrial FARO laser scanner, and were aligned to the floor plan of the second floor. By enforcing a shared coordinate system, we are able to compare the pose estimation to our ground truth poses.

Recording R1-R11 are captured in the second floor of our building and include ground truth poses. Recording Q1-Q3 were recorded in the first floor of the building and do not include ground truth information, and are used for qualitatively evaluation on previously unseen environment. Each sequence was evaluated multiple times to account for the inherent stochasticity of the MCL framework.

In our implementation, we used YOLOv5 [15], which is a family of object detection models of varying size and performance. YOLOv5 models are capable of real-time inference on CPU-only platforms, thus making them well-suited for mobile robots. We trained a small model, YOLOv5s, on 581 images from the second floor of our building using the default training script provided in the YOLOv5 repository. The map used for localization is joint map of two CAD floor plan drawings, of the first and second floor side-by-side, illustrated in Fig. 1. The semantic information was integration using our GUI application MAPhisto[3] based on our recollection of location of semantic objects. For all experiments, we use a map resolution of 0.05 m by 0.05 m per cell and the algorithm parameters specified in Tab. II.

As baseline, we compare against AMCL [27], which is a publicly available and highly-used ROS package for MCL-based localization, our own MCL implementation without using semantic cues and a text-enhanced MCL [43], which we refer to as TMCL. Our method, exploiting both semantic information from object detection and hierarchical semantic knowledge discussed in Sec. III-G, is referred to as HSMCL. For the tracking experiments, we considered SMCL, a variation of our approach that uses only semantic cues through the semantic visibility model, without hierarchical semantic localization. All experiments were executed with 10,000 particles in the filter unless mentioned otherwise.

We consider two metrics, the success rate and absolute trajectory error (ATE) after convergence. In our definition, convergence occurs when the estimated position is within a radius of 0.3 m from the ground truth pose and the estimated orientation is within $\frac{\pi}{4}$ radians. Tracking of the pose is considered unreliable if the pose estimate diverges for more than

[3]https://github.com/FullMetalNicky/Maphisto

TABLE I: Semantic stability scores for different detected object classes computed on sequences T1-T5.

Class	sink	door	oven	whiteboard	table	cardboard	plant	drawers	sofa	storage	chair	extinguisher	person	desk
Score	0.97	0.96	0.90	0.91	0.95	0.46	0.88	0.86	0.99	0.96	0.58	0.84	0.11	1.00

TABLE II: Algorithm parameters

Method	σ_{odom}	σ_{obs}	$r_{\max}$	τ_s	ρ	d_{xy}	d_θ
MCL	(0.15 m, 0.15 m, 0.15 rad)	6.0	15.0 m	-	-	0.1 m	0.03 rad
TMCL	(0.15 m, 0.15 m, 0.15 rad)	6.0	15.0 m	-	0.5	0.1 m	0.03 rad
HSMCL	(0.15 m, 0.15 m, 0.15 rad)	6.0	15.0 m	0.6	-	0.1 m	0.03 rad

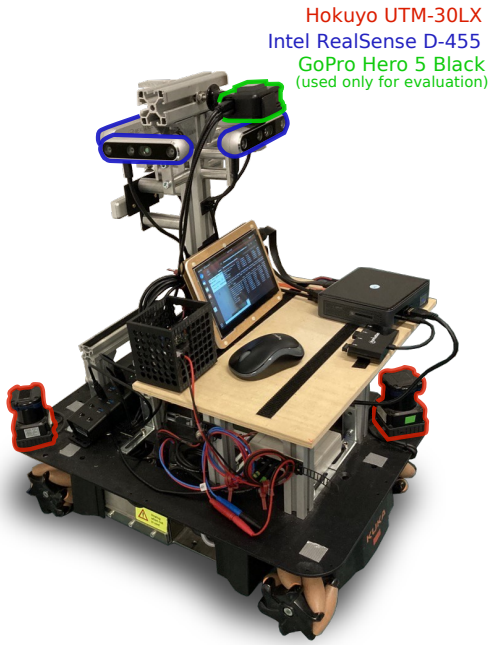

Fig. 5: The data collection platform, an omnidirectional Kuka YouBot, with 2D LiDAR scanners (marked by a red outline) and with 4 cameras (marked by a blue outline) providing 360° coverage. The up-ward facing camera (marked by a green outline) is only used for generating the ground truth via AprilTag detections.

1% of the time. If convergence did not occur within the first 95% of the sequence, or if the pose is not reliably tracked from convergence moment until the end of the sequence, we consider it a failure.

B. Long-Term Localization in CAD Floor Plans

The first experiment evaluates the performance of our approach and supports the claim that we are capable of long-term localization in sparse, floor-plan-like maps. Sequences R1-R11 are recorded in April-June 2022, and traverse all the rooms in the second floor. The given map had been constructed in 2021. All sequences include humans walking around, opening and closing of doors, moving furniture and large amount of clutter. We repeat the evaluation of each sequence 5 times, computing the success rate, ATE and convergence time over all 5 runs, and compare against the baselines. As can be seen in Tab. III the semantically-enhanced methods have superior performance over the baselines. AMCL and MCL are

TABLE III: Success rate for 11 sequences recorded all across the second floor in the span of several weeks. A run was considered successful if the algorithm converged to the ground truth in the first 95% of the recording and remained localized until the end of the sequence.

Method	R1	R2	R3	R4	R5	R6
AMCL	0%	0%	0%	0%	0%	0%
MCL	40%	20%	60%	40%	20%	0%
TMCL	80%	0%	100%	80%	60%	40%
HSMCL	100%	100%	100%	100%	100%	100%

Method	R7	R8	R9	R10	R11	AVG
AMCL	0%	0%	0%	0%	0%	0%
MCL	0%	40%	20%	60%	0%	27%
TMCL	100%	100%	100%	80%	100%	76%
HSMCL	100%	100%	100%	100%	100%	100%

mostly used with detailed maps constructed using range-sensor measurements, and we can attribute their poor performance to the sparse nature of the floor plans. This highlights the impact of semantic information when localizing in nondescript, sparse maps, especially in face of high geometric symmetry.

As reported in Tab. IV, upon successful convergence, HSMCL achieves accuracy of 0.23 m and negligible angular error. HSMCL successfully converges, on average over all sequences, after 25 s.

We further provide pose tracking experiments. A similar approach to ours, Boniardi et al. [4], tracked the pose of a robot by inferring the room layout from camera images, reporting RMSE of approx. 0.23 m and approx. 0.04 rad with adaptive particle number ranging between 1,500-5,000. However, they did not provide open source code. Our office environment is similar to the Freiburg one where Boniardi et al. [4] evaluated their method. For the tracking experiments we used SMCL, which integrates semantic cues from object detection, without hierarchical information. We report our tracking results with fixed 1500 particles in Tab. V, achieving an ATE of 0.2 m and 0.05 rad. This suggest that integrating semantic cues, and specifically, our SMCL approach, are beneficial also for tracking purposes and not only for global localization.

C. Localization in a Previously Unseen Environment

To support our claim that we are able to localize in a previously unseen environment, we qualitatively evaluate our method on sequences Q1-Q3 recorded on the first floor of our building. The object detection model and the room category classifier were not trained or validated on data from this floor. While the first floor is not entirely dissimilar to the second one, it does include different furniture and rooms that serve different purposes such as a classroom and a robotics lab. The pose estimated by SMCL sequences is shown in Fig. 6. Our approach correctly predicts that the robot is located in the first floor and identifies the correct room and maintaining a trajectory that is consistent with floor plan map. We manually

TABLE IV: ATE for global localization on consistently stable (100% success rate) sequences recorded all across the second floor in the span of several weeks. Angular error in radians / translational error in meters.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	AVG
AMCL	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
MCL	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
TMCL	-/-	-/-	0.048/0.16	-/-	-/-	-/-	0.034/0.22	0.043/0.18	0.050/0.21	-/-	0.034/0.18	0.042/0.19
HSMCL	0.054/0.15	0.064/0.24	0.069/0.25	0.205/0.23	0.100/0.34	0.064/0.23	0.069/0.23	0.049/0.18	0.090/0.26	0.052/0.16	0.052/0.25	0.079/0.23

TABLE V: ATE for tracking on a subset of sequences recorded all across the second floor in the span of several weeks. The particle filter was set to adaptive 1,500-5,000 particles for AMCL and a fixed 1,500 particles for MCL and SMCL. Angular error in radians / translational error in meters.

Method	R3	R4	R6	R7	R8	R10	AVG
AMCL	-/-	-/-	0.047/0.22	-/-	-/-	-/-	0.047/0.22
MCL	0.051/0.17	0.050/0.21	0.051/0.29	0.064/0.23	0.039/0.14	0.041/0.15	0.049/0.20
SMCL	0.063/0.21	0.046/0.22	0.068/0.29	0.048/0.19	0.042/0.15	0.044/0.13	0.052/0.20

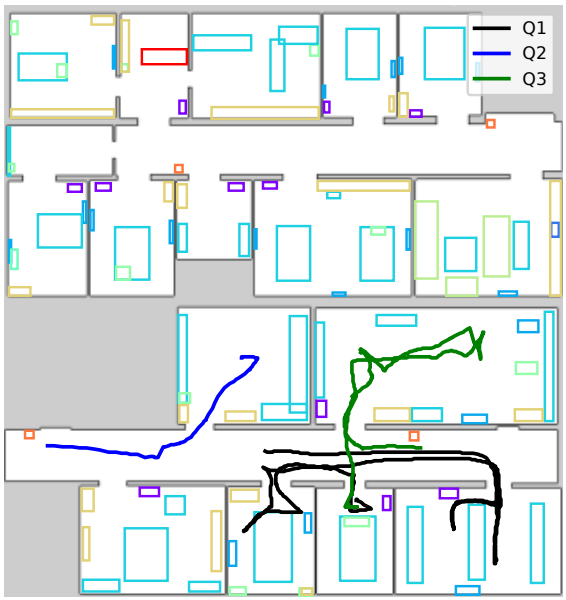

Fig. 6: Examples of pose estimation for localization in previous unseen environment, using SMCL and 10,000 particles.

verified that the robot's estimated trajectory corresponded to the rooms visited using the RGB images from sequences Q1-Q3.

D. Ablation Study

To justify our use of both low-level and hierarchical semantic information, we conducted an ablation study. We analyzed three strategies for integrating semantic knowledge into an MCL framework. SMCL uses only semantic cues through the semantic visibility model. HMCL uses semantic hierarchy, described in Sec. III-G, to initialize the particles only in the rooms corresponding to the observed room category, and then relies solely on the LiDAR information. HSMCL combines both strategies. The ATE was computed only on stable sequences with 100% success rate. As can be seen in Tab. VI, utilizing the two levels of semantic information benefits localization. HSMCL was able to localize stably even on the challenging sequences, where other methods failed. The ATE for HSMCL is on par with the other methods, and

TABLE VI: Performance on 11 sequences recorded all across the second floor in the span of several weeks. A run was considered successful if the algorithm converged to the ground truth in the first 95% of the recording and remained localized until the end of the sequence. Angular error in radians / translational error in meters.

Method	Hierarchy	Semantics	Success	ATE (# of stable sequences)	ATE (# of successful runs)
MCL			27%	- (0)	0.046/0.20 (15)
HMCL	✓		61%	0.046/0.21 (3)	0.044/0.19 (34)
SMCL		✓	81%	0.055/0.23 (7)	0.066/0.24 (45)
HSMCL	✓	✓	100%	0.079/0.23 (11)	0.079/0.23 (55)

TABLE VII: Runtime for HSMCL, with 10,000 particles. The Yolov5s results are for inference on a single camera.

Platform	Sem. Visibility	Beam-End	Yolov5s (640x480)	Yolov5s (320x240)
NUC10i7FNK	55 ms	24 ms	223 ms	57 ms
Dell Precision-3640-Tower	19 ms	14 ms	10 ms	6.8 ms

the slightly larger error can be attributed to including more challenging sequences and runs in the computation of the ATE for HSMCL, sequences and runs where other methods failed to localize entirely.

E. Runtime

We evaluate the runtime performance of our approach in support of our fourth claim, that we are able to operate onboard and allow real-time localization. We tested our approach on a Dell Precision-3640-Tower (with NVidia GeForce RTX 2080) and once on an Intel NUC10i7FNK, which we have on our robot. The Dell PC has 64 GB of RAM and runs at 3.70 GHz. The Intel NUC has 16 GB of RAM and runs at 1.10 GHz. The measurements are reported in Tab. VII. Since we are using 4 cameras simultaneously for object detection, we used an optimized ONNX export of YOLOv5s, and run inference on 320 by 240 images. Qualitative online tests indicates that reducing the resolution does not impact the detection accuracy significantly. These runtime results suggest that our approach is suitable for online localization, and utilizes semantic information without requiring a GPU onboard.

V. Conclusion

Our approach incorporates semantic information, from low-level object detection to higher understanding of room categories, to assist navigation in human-oriented environments. This enables us to successfully localize in sparse floor plans under high geometric symmetry and changing environments. We demonstrate that using sparse and abstract map representation benefits long-term localization, and reduces the need to update the map. We also provide a tool for updating the semantic map, when critical changes occur. For our evaluation, we recorded a dataset spanning across weeks, introducing a

variety of elements that are not represented in the floor plan, and the changes a human-occupied environment undergoes. We compared our performance to other existing methods, supporting all of our claims. The results of our experiments imply mobile localization systems can benefits greatly from exploiting ever-present semantic cues.

ACKNOWLEDGMENTS

We thank Lior Rozin for his contribution to the GUI application MAPhsito. We are also grateful to segments.ai for supporting our research with free and full access to all features. We also thank Matteo Sodano, Louis Wiesmann and Thomas Läbe for their assistance with the data collection.

REFERENCES

[1] N. Atanasov, M. Zhu, K. Daniilidis, and G.J. Pappas. Localization from semantic observations via the matrix permanent. *Intl. Journal of Robotics Research (IJRR)*, 35(1-3):73–99, 2016.

[2] A. Bochkovskiy, C.Y. Wang, and H.Y.M. Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint*, 2004.10934, 2020.

[3] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard. Robust LiDAR-based localization in architectural floor plans. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[4] F. Boniardi, A. Valada, R. Mohan, T. Caselitz, and W. Burgard. Robot localization in floor plans using a room layout edge extraction network. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Trans. on Robotics (TRO)*, 32(6):1309–1332, 2016.

[6] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[7] L. Cui, C. Rong, J. Huang, A. Rosendo, and L. Kneip. Monte-Carlo Localization in Underground Parking Lots Using Parking Slot Numbers. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[8] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 1999.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Visual Object Detection with Deformable Part Models. *Communications of the ACM*, 56(9):97–105, 2013.

[10] P.F. Felzenszwalb and D.P. Huttenlocher. Distance Transforms of Sampled Functions. *Theory of Computing*, 8(1):415–428, 2012.

[11] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)*, 11:391–427, 1999.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2017.

[13] R. Hendrikx, P. Pauwels, E. Torta, H. Bruyninckx, and M. van de Molengraft. Connecting Semantic Building Information Models and Robotics: An application to 2D LiDAR-based localization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[14] S. Ito, F. Endres, M. Kuderer, G. Tipaldi, C. Stachniss, and W. Burgard. W-RGB-D: Floor-Plan-Based Indoor Global Localization Using a Depth Camera and WiFi. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[15] G. Jocher. ultralytics/yolov5: v3.1. https://github.com/ultralytics/yolov5, 2020.

[16] J. Leonard and H. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382, 1991.

[17] L. Li, B. Yang, M. Liang, W. Zeng, and M. Ren. End-to-end Contextual Perception and Prediction with Interaction Transformer. *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[18] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai. Real-time Scene Text Detection with Differentiable Binarization. *arXiv preprint*, 1911.08947, 2019.

[19] D. Lowe. Object recognition from local scale-invariant features. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 1999.

[20] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric Object-Level SLAM. In *Proc. of the Intl. Conf. on 3D Vision*, 2018.

[21] O. Mendez, S. Hadfield, N. Pugeault, and R. Bowden. Sedar-semantic detection and ranging: Humans can localise without lidar, can robots? In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[22] R. Miyagusuku, A. Yamashita, and H. Asama. Data Information Fusion From Multiple Access Points for WiFi-Based Self-localization. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):269–276, 2019.

[23] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 1985.

[24] H.P. Moravec. Sensor Fusion in Certainty Grids for Mobile Robots. In *Sensor Devices and Systems for Robotics (SDSR)*, 1989.

[25] A. Mucherino, P.J. Papajorgji, and P.M. Pardalos. *k-Nearest Neighbor Classification*. Springer Verlag, 2009.

[26] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[27] P. Pfaff, W. Burgard, and D. Fox. Robust Monte-Carlo Localization Using Adaptive Likelihood Models. In *STAR Springer Tracts in Advanced Robotics*, 2006.

[28] N. Radwan, G. Tipaldi, L. Spinello, and W. Burgard. Do You See the Bakery? Leveraging Geo-Referenced Texts for Global Localization in Public Maps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[29] A. Ranganathan and F. Dellaert. Semantic modeling of places using objects. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.

[30] A. Rottmann, O. Martínez-Mozos, C. Stachniss, and W. Burgard. Place Classification of Indoor Environments with Mobile Robots using Boosting. In *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, pages 1306–1311, 2005.

[31] M. Runz, M. Buffier, and L. Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2018.

[32] D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld, and H.M. Gross. Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[33] B. Shi, X. Bai, and C. Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *arXiv preprint*, 1507.05717, 2015.

[34] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford. Place categorization and semantic mapping on a mobile robot. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[35] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[36] P. Trahanias, W. Burgard, A. Argyros, D. Hähnel, H. Baltzakis, P. Pfaff, and C. Stachniss. TOURBOT and WebFAIR: Web-Operated Mobile Robots for Tele-Presence in Populated Exhibitions. *IEEE Robotics and Automation Magazine (RAM)*, 12(2):77–89, 2005.

[37] K. Wada, E. Sucar, S. James, D. Lenton, and A.J. Davison. MoreFusion: Multi-object Reasoning for 6D Pose Estimation from Volumetric Fusion. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger. MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[39] C. Yi, I.H. Suh, G.H. Lim, and B.U. Choi. Bayesian robot localization using spatial object contexts. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[40] F. Zafari, A. Gkelias, and K.K. Leung. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys Tutorials (CST)*, 21(3):2568–2599, 2019.

[41] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L.M. Ni, and H.Y. Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv preprint*, 2203.03605, 2022.

[42] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. In *Conference on Neural Information Processing Systems (NIPS)*, 2014.

[43] N. Zimmerman, L. Wiesmann, T. Guadagnino, T. Läbe, J. Behley, and C. Stachniss. Robust Onboard Localization in Changing Environments Exploiting Text Spotting. *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.