Research Article

OMCL: Open-vocabulary Monte Carlo Localization

Evgenii Kruzhkov[1]

1. University of Bonn, Bonn, Germany

Robust robot localization is an important prerequisite for navigation, but it becomes challenging when the map and robot measurements are obtained from different sensors. Prior methods are often tailored to specific environments, relying on closed-set semantics or fine-tuned features. In this work, we extend Monte Carlo Localization with vision-language features, allowing OMCL to robustly compute the likelihood of visual observations given a camera pose and a 3D map created from posed RGB-D images or aligned point clouds These open-vocabulary features enable us to associate observations and map elements from different modalities, and to natively initialize global localization through natural language descriptions of nearby objects. We evaluate our approach using Matterport3D and Replica for indoor scenes and demonstrate generalization on SemanticKITTI for outdoor scenes. The code is accessible at https://github.com/AIS-Bonn/omcl.

Corresponding author: Evgenii Kruzhkov, ekruzhkov@ais.uni-bonn.de

I. Introduction

Localization is a fundamental problem in robotics, allowing robots to estimate their position and orientation within an environment. Traditional approaches utilize single or fused sensor modalities [1][2][3]. Cross-modal localization [4] further increases flexibility for heterogeneous systems and facilitates map reuse across platforms, while incorporating semantic awareness enhances robot autonomy [5][6]. The semantics can be incorporated through object detection [7], semantic segmentation [8], or vision-language features [9][10]. This letter introduces the Open-vocabulary Monte Carlo Localization (OMCL) framework, which extends Monte Carlo Localization (MCL) with vision-language features [10]. These abstract features enable affordable camera-only localization in 3D maps created from different sensors,

 doi.org/10.32388/I8VL7Y

like RGB-D cameras or LiDAR. Fig. 1 demonstrates the language map and localization process of the proposed framework.

Our approach stores features learned by contrastive language-image pretraining (CLIP [10]) in a 3D map. The positions of features are represented within a spatial language map, while RGB input is processed to extract the features that describe the current observation. Ray tracing is employed to estimate the correlation between the observations and the map allowing for pose estimation in 3D environments. We propose a technique to convert pre-existing maps to the compatible representation for our localization framework, compare it to state-of-the-art methods, and conduct a comprehensive ablation study. Our contributions include the following:

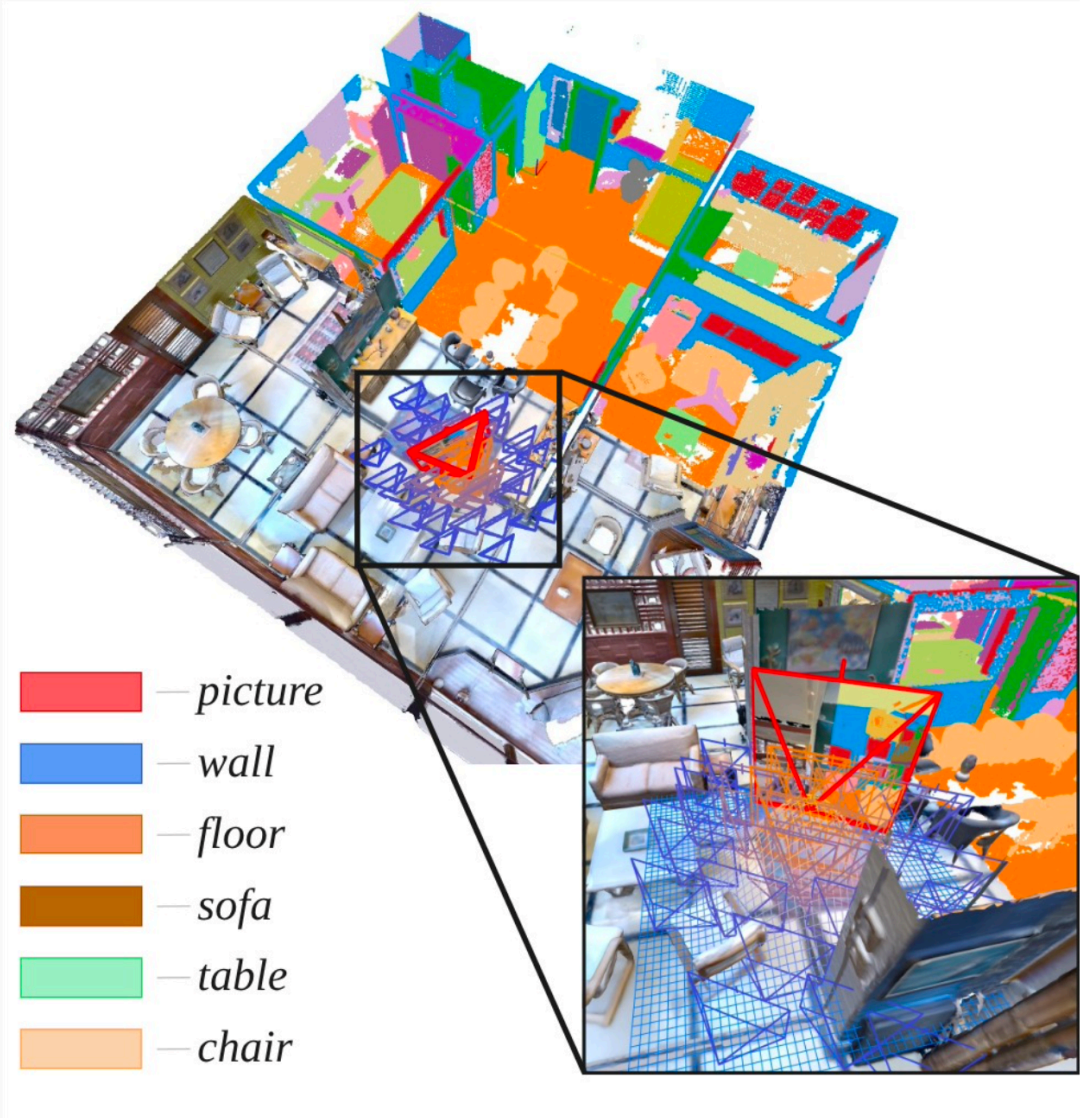


Figure 1. OMCL particles are sampled on a Language Map storing open-vocabulary features. Each particle represents a candidate camera pose. Particles are weighted according to how well the VLM-processed RGB input matches the ray-traced map value at the relevant location. The red particle denotes the estimated pose (weighted mean). The Language Map is colored by similarity to the prompted labels.

- **Language-grounded localization.** We present OMCL, a novel localization framework that grounds pose estimation in language features and accelerates global localization via open-vocabulary prompts.
- **Cross-modal sensor usage.** OMCL includes a mapping module that constructs a unified, sparse language map, enabling sensors of different modalities to be used for mapping and localization (e.g., RGB-D or point clouds for mapping and RGB for localization).

doi.org/10.32388/I8VL7Y

- **Generalization.** OMCL is compatible with independently constructed point clouds, enabling reuse of existing maps, and it generalizes across indoor and outdoor environments.

OMCL offers an effective, language-feature guided solution for localization. It achieves state-of-the-art performance compared to existing baselines and opens up new possibilities for natural language-guided and cross-modal localization. A robot can be localized by non-expert users without the necessity of comprehending the underlying map representation through the use of natural language descriptions in OMCL.

II. Related Works

Semantic Localization. We focus on visual continuous localization within 3D maps constructed from cross-modal sensors [11][12]. Unlike Loc-NeRF [13], which relies on rendered visual consistency for pose tracking, we achieve sensor cross-modality through semantic consistency. Prior methods, SeDAR [14] and SIMP [7], use MCL for long-term indoor semantic localization on floor plans; in contrast, we propose a more general map representation and a semantic-consistency measurement model. We evaluate against SeDAR [14] and SIMP [7] by adapting our map representation to their floor-plan views.

SemLoc [15] and Zhang *et al.* [16] explore semantic consistency for localization in outdoor environments. Unlike these methods, we rely purely on semantic consistency in a multi-modal feature space, without additional geometry-based optimization terms or custom semantic segmentation models. CMRNext [4] is a state-of-the-art camera-to-LiDAR map-matching baseline for outdoor localization, which, in contrast to our approach, relies on a fine-tuned neural network model.

Unlike our approach, place recognition (PR) methods [17] estimate the pose from a single image, whereas we handle continuous image streams and pose tracking, potentially requiring multiple observations before convergence. However, MCL allows using the same map for both global localization and subsequent pose tracking [18]. While PR can be used for initialization, it requires additional PR-specific infrastructure; in contrast, we propose a map-native, prompt-augmented initialization.

Semantic Mapping. Many recent works focus on open-set semantic mapping of indoor environments followed by 3D semantic segmentation. SAM3D [19] creates 3D scene masks from SAM [20]. OpenScene [5] proposes a method to directly predict point-wise CLIP [10] features for input point clouds. ConceptGraphs [21] constructs a graph-based representation with incorporated semantic feature vectors.

ConceptFusion [22], implemented on top of ∇SLAM [23], produces unordered multi-modal maps. LiLMaps [24] investigates sequential visual–language mapping for implicit representations, reducing the memory footprint of such dense maps. Unlike these mapping-focused works, we provide options for constructing the map from different sensing modalities and couple it with a ray-tracing-based, semantic-aware probabilistic localization module.

The recent OVO [25] mapping approach uses Gaussian-SLAM [26] and ORB-SLAM2 [27] but does not exploit the semantic information added to the map to improve pose estimation. Concurrent to our work, RayFronts [28] similarly constructs an ordered map by averaging visual–language features from multiple directions. However, they focus on multi-modal open-set querying and beyond-range semantic classification. Similar to the aforementioned works, we construct our map representation from externally estimated poses, while tight integration with simultaneous localization and mapping (SLAM) [29][30] is a possible future improvement.

Vision Language Models. We employ contrastive Vision-Language Models (VLMs) [5][10][28][31][32], for the localization task. VLMs and large language models (LLMs) are widely used for related navigation [6] [33] and scene understanding [34] tasks. We use pixel-wise visual-language features (such as CLIP [10]) produced by the image encoder as open semantic representations in the maps. By default, our framework employs LSeg [31] and OpenScene [5] encoders, but we demonstrate flexibility with respect to the choice of visual–language backbone by switching to X-Decoder [32] and NARADIO [28].

We don't compete with the aforementioned approaches; instead, our method creates a unified visual-language map from cross-modal sensors and evaluates its downstream impact on localization.

III. Method

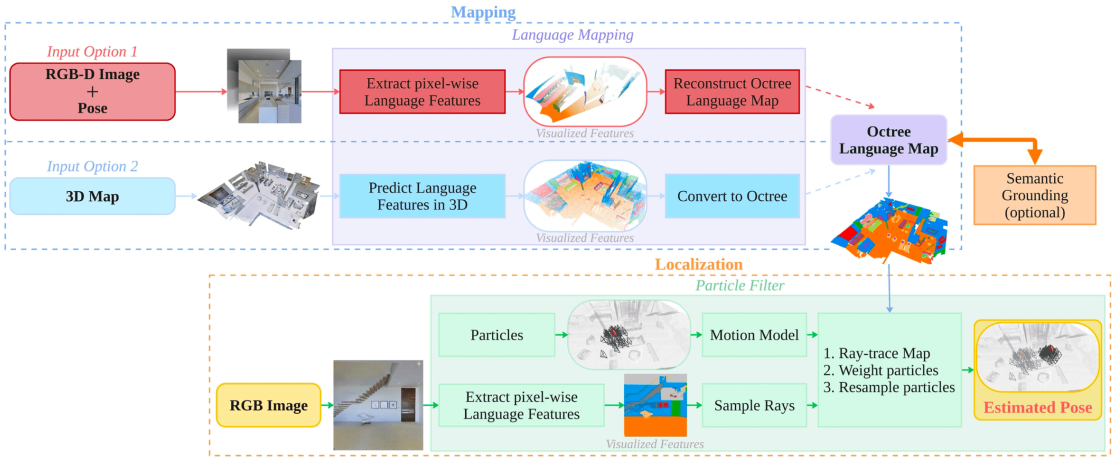


Figure 2. Mapping: We propose two options to create Octree Language Maps. *Input Option 1*: OMCL derives language features from RGB images and reconstructs a 3D map from them using the corresponding volumetric data (depth images, LiDAR measurements, etc.). *Input Option 2*: Language features are directly predicted on precomputed 3D point clouds for each point and subsequently converted into the octree representation. **Localization:** A particle filter uses an RGB image as the only input, weighting particles by the discrepancy between language features extracted from the input image and those ray-traced from the Octree Language Map. Our stratified ray sampling strategy compensates for the imbalance between different object instance sizes in the image. All features are colored for visualization purpose only.

The proposed pipeline is illustrated in Fig. 2. The map representation constructed and used by our framework is the Octree Language Map. We focus on storing visual–language features [5][10][31] in the map to investigate their potential for advancing perception systems. Section III-A covers Octree Language Map implementation details and, after that, describes its construction from different inputs based on pixel-wise visual encoders for RGB-D input and OpenScene [5] for point clouds.

In Section III-B we address the visual-only localization in 3D maps. Although multimodal sensor configurations (e.g., RGB-D cameras, LiDAR) can be employed for mapping, the subsequent localization requires only visual data and map-scaled odometry (Motion Model), under the assumption that the features in the map are consistent with those extracted from the input RGB image.

We focus on mapped visual–language features that are semantically grounded using open-set semantic prompts.

A. Mapping

Throughout the mapping process, we generate a volumetric language map (Octree Language Map), a sparse octree-based structure, in which each map voxel is linked to the corresponding F-dimensional visual-language feature [5][10][31]. The Octree Language Map offers compact storage at fine-grained resolution and provides efficient ray-tracing functionality [35]. It retains only features that are mutually different by at least a cosine distance threshold τ. Each voxel in the map stores the index of its corresponding feature in the database $\text{Features}_{\text{DB}}$:

$$\begin{aligned} &\text{Features}_{\text{DB}} = \{f_i \in \mathbb{R}^n, |d(f_i, f_j) > \tau; i, j \in \mathbb{N}\}, \\ &d(A, B) = 1 - \frac{A \cdot B}{\|A\|\|B\|}, \text{where} A, B \in \mathbb{R}^n. \end{aligned} \tag{1}$$

We provide two mapping approaches: one for posed RGB-D images and another for fused point clouds (Fig. 2). The produced Octree Language Map is unified.

Input Option 1. Unless stated otherwise, we use the LSeg [31] visual–language model to extract pixel-wise features and project them onto the Octree Language Map using depth measurements and known camera poses. The mapping is sequential, and features projected into the same voxel are cumulatively averaged with the existing ones. Once all data have been processed, the mapped features form $\text{Features}_{\text{DB}}$ according to Eq. 1. The first features that satisfy Eq. 1 are added to $\text{Features}_{\text{DB}}$, while subsequent ones are replaced by their cosine-closest counterparts from $\text{Features}_{\text{DB}}$.

Input Option 2. We process the aggregated point cloud map with the OpenScene 3D distillation model [5] to predict visual–language features for each point in a feed-forward manner. The Octree Language Map is then formed by averaging features within voxels and constructing $\text{Features}_{\text{DB}}$ as input option 1. The described approach is suitable for converting existing 3D maps into our Octree Language Map representation.

Semantic Grounding. Each element of the constructed $\text{Features}_{\text{DB}}$ can be viewed as an automatically created semantic class. We propose to ground $\text{Features}_{\text{DB}}$ using user-defined open-set semantic prompts. In practice, we employ the text encoder corresponding to the mapping model (CLIP [10] for LSeg [31] and OpenScene [5]) and redefine $\text{Features}_{\text{DB}}$ using the features of the provided open-set semantic classes. The feature indices stored in the voxels of the Octree Language Map are remapped based on the highest cosine similarity between their original associated features and the redefined $\text{Features}_{\text{DB}}$. Old features that are too far in cosine distance from the provided open-set semantic classes are discarded along with

their corresponding map voxels. Grounding reduces the memory footprint by decreasing the number of stored features. It increases the discriminability of the remaining features and allows the user to specify which semantic classes are represented in the map for downstream localization.

B. Localization

We use RGB images and odometry (Motion Model) to localize within the language maps employing MCL. We initially assign uniformly distributed weights to each particle and sample particles with random poses around the probable starting position. When a new RGB image is received, the motion model is applied to all particles to predict their new poses. A coarse odometry from any built-in sensor can be employed. The image is processed using the language-driven semantic segmentation model [31] to extract per-pixel features. To form the measurement model of our MCL and assign new weights for the particles, we evaluate the consistency between the extracted features and the map (Fig. 2 bottom).

Observation Likelihood. If the camera is exactly at a particle's pose, the features extracted from the image should match those stored in the Octree Language Map where viewing rays hit a surface. We employ camera intrinsics and each particle's predicted pose to form the rays in world coordinates. Ray-tracing [35] on the map finds the first voxel hit by each ray, from which we retrieve the stored language feature. Let γ and φ denote the voxel features and pixel features corresponding to the same rays, respectively. The weight w_i^t of particle i at time t is then estimated as:

$$w_i^t = \frac{w_i^{t-1}\max\left(\mathcal{L}_i, 0\right)}{\sum_i \left(w_i^{t-1}\max\left(\mathcal{L}_i, 0\right)\right)}, \mathcal{L}_i = \frac{1}{N}\sum_{j=1}^{N}\frac{\varphi_j \cdot \gamma_j}{\|\varphi_j\|\|\gamma_j\|}, \tag{2}$$

where N represents the total number of rays per particle.

$\mathcal{L}_i$ is, in effect, the averaged cosine similarity between image and map features. The more similar the observations and the map are, the higher the value of $\mathcal{L}_i$ and, consequently, the higher the corresponding particle weight w_i^t. The final camera pose is then estimated as the average of particles that are close to the most likely particle, using their corresponding weights w_i^t.

Stratified Ray Sampling. We reduce computational costs by sampling and using only N pixels and their corresponding features from the image. However, large surfaces (e.g., walls, floor) occupy most pixels, while smaller but distinctive objects cover far fewer pixels, causing many objects to be underrepresented or missed when sampled uniformly. Since accurate pose estimation relies on preserving information from all objects, we introduce sampling masks for the image, where each mask corresponds to a cluster

of pixels with similar visual-language features. From each such feature-consistent cluster, we uniformly sample an equal number of pixels, discarding duplicates (Fig. 3). The same samples are employed for all particles. The proposed stratified per-cluster sampling improves pose estimation accuracy, as confirmed by our ablation study in Section IV-D.

To form the clusters, we use the map features $\text{Features}_{\text{DB}}$ as centroids because, according to Eq. 2, localization assumes that map and image features are correlated and $\text{Features}_{\text{DB}}$ already stores only distinct features (see Eq. 1). The clusters are then formed from the extracted image features assigned to the centroids with which they produce the highest cosine similarity. Unless stated otherwise, we use full $\text{Features}_{\text{DB}}$ for clustering, though random subsets of centroids may be used to limit the number of clusters and reduce computation. Both $\text{Features}_{\text{DB}}$ and the clusters are created automatically, requiring no manual labeling for mapping or localization.

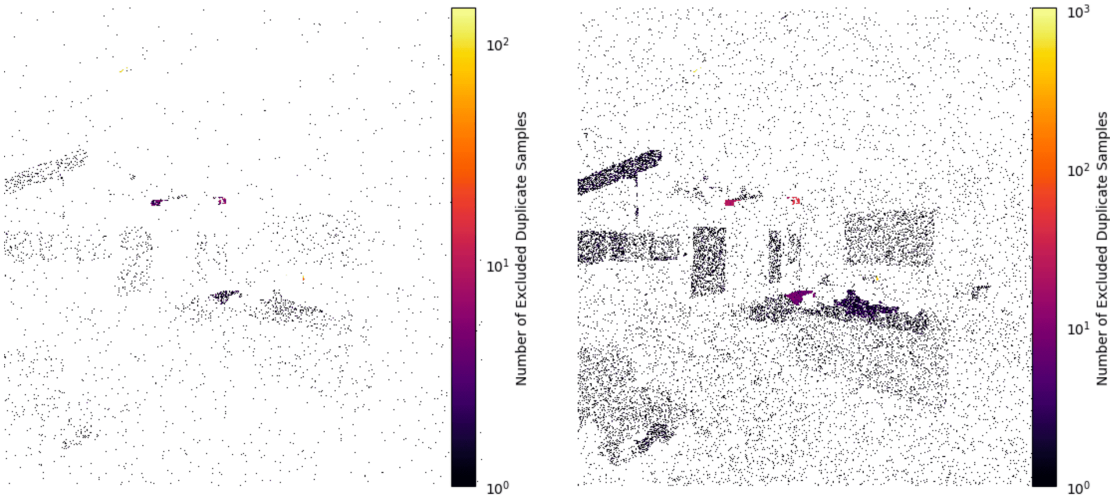


Figure 3. Sampled pixels for images of resolution 540×540. Both images employ the same sampling masks. The left image corresponds to 2^8 samples per cluster and the right one to 2^{11}. Small clusters have a higher sampling density, but the total number of samples is less for them because the duplicates are discarded.

C. Prompt-augmented Initialization

Standard approaches for global localization either require geometric coordinates as input for initialization or sample particles across the entire environment. However, sampling across the whole environment is inefficient, while geometric coordinates are difficult for non-expert users to interpret or provide, and localization may diverge if the initialization is incorrect. Moreover, global localization that is

doi.org/10.32388/I8VL7Y

decoupled from the subsequent pose tracking step may require storing additional information within the map itself.

We propose using textual prompts θ to describe probable initial positions. As demonstrated in Fig. 4, the prompt is an open set of natural language words that describe the surrounding environment. There is no restrictions on word content, as each word is encoded into a feature using a text encoder (CLIP [10] for LSeg [31] and OpenScene [5]).

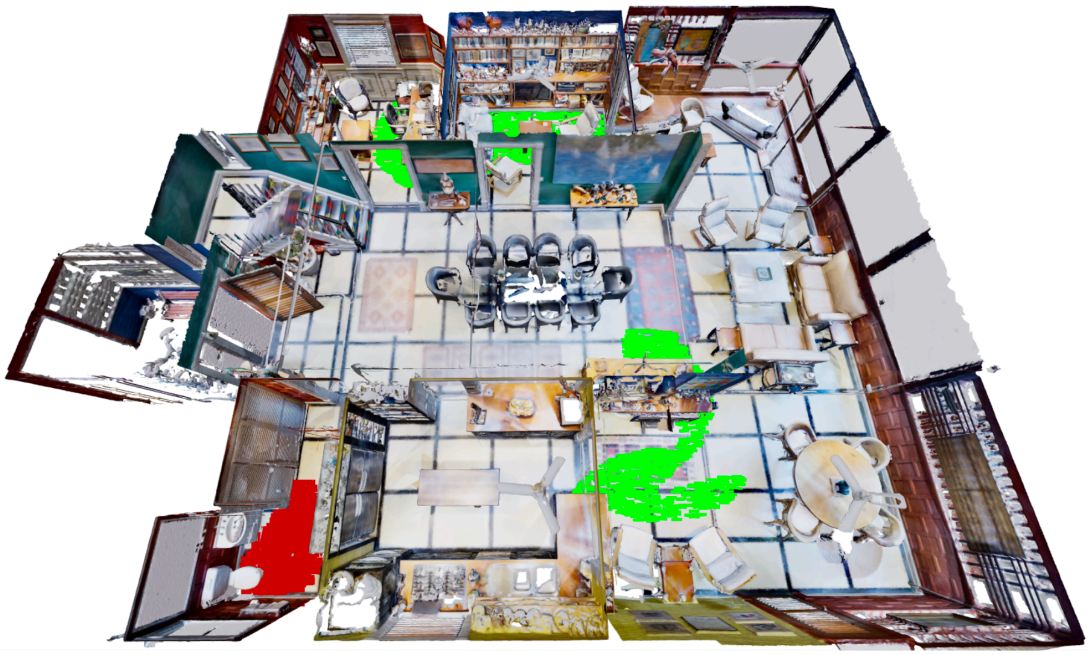


Figure 4. An example of initial locations for the global localization based on the user prompt. The red spots correspond to the prompt **(toilet, mirror, towel, sink)** and the green one to **(table, chair, picture, door, tv monitor)**. OMCL particles can be initialized nearby the prompt matching spots instead of the random locations.

First, the proposed initialization accepts natural language words as input, taking a step toward full natural language-based localization. Second, since the input is text, it is naturally compatible with the output of large language models, enabling initialization with AI agents. Finally, instead of relying on a single initial location, we initialize particles by sampling them uniformly over all locations that match the given prompt. Through direct integration with MCL, the initialization subsequently converges to a precise pose, as described in Section III-B, based on multiple consecutive observations, and then MCL

continues to track the pose. The same Octree Language Map is employed for both global localization and following pose tracking.

We segment the scene into two classes: floor and surroundings. We obtain the floor voxels by comparing all stored features with the feature corresponding to the word “floor”, extracted by the text encoder. For each floor voxel, we then find surrounding features $\bar{\gamma}$ within a radius R. Next, for each word in the prompt θ, we count the number of these surrounding features whose cosine similarity with that word exceeds the threshold ρ:

$$\mathcal{V}_m = \sum_n (\text{CosineSimilarity}(\bar{\gamma}_{n\times F}, \bar{\theta}_{m\times F}) > \rho), \tag{3}$$

where n is the number of neighbor surroundings, m is the number of words in the prompt, and F is the feature dimensionality.

Finally, we estimate the floor voxel–prompt alignment ratio s as a measure of how well the floor voxel matches the prompt, by requiring that at least k of its surrounding voxels correspond to each word:

$$s = \frac{1}{m}\sum_m (\mathcal{V}_m \geq k), 0 \leq k \leq m, \tag{4}$$

where s is equal to 1 for floor voxels that match all words. The particles are uniformly sampled above “floor” voxels that match all words.

IV. Experiments

		RMSE [m]	STD [m]	Mean [m]	Median [m]	min RMSE [m]	max RMSE [m]	SSE [m^2]	Completed Scens
2D	***OMCL***$_{\mathrm{GT}}$	*0.11*	*0.07*	*0.09*	*0.08*	*0.08*	*0.18*	*17.38*	*10/10*
	OMCL$_{\mathrm{LSeg}}$	**0.15**	**0.09**	**0.13**	**0.12**	**0.11**	**0.28**	**33.07**	**10/10**
	OMCL$^{\mathrm{auto}}_{\mathrm{LSeg}}$	0.24	0.15	0.18	0.15	0.14	0.52	95.0	10/10
	OMCL $_{\mathrm{Open\backslash LSeg}}$	0.36	0.24	0.27	0.22	0.15	0.62	164.07	8/10
	SeDAR $^{*}_{\mathrm{GT}}$ [14]	0.79	0.46	0.64	0.54	0.24	2.33	1891.2	6/10
	SIMP$^{*}_{\mathrm{GT}}$ [7]	1.39	1.03	0.79	0.94	0.45	2.81	2898.15	4/10
3D	***OMCL***$_{\mathrm{GT}}$	*0.15*	*0.09*	*0.12*	*0.1*	*0.1*	*0.3*	*30.45*	*10/10*
	OMCL$_{\mathrm{LSeg}}$	**0.2**	**0.1**	**0.18**	**0.17**	**0.17**	**0.41**	**59.53**	**10/10**
	OMCL $_{\mathrm{Open\backslash LSeg}}$	0.42	0.25	0.31	0.25	0.22	0.75	223.32	8/10
	RTAB-Map [36]	5.23	2.97	4.30	3.6	0.31	8.14	25781.93	10/10

Table I. Absolute Trajectory Error (APE) for Localization on Matterport3D Dataset.

We provide complete evaluation of our localization in medium-size indoor environments on Matterport3D dataset [37], comparing against similar approaches and using publicly available sequences provided by VLMaps [6]. We further analyze how localization performance depends on measurement–map consistency.

Using the Replica dataset [38] for smaller, room-sized environments, we obtain localization accuracy aligned with that observed on medium-sized environments and additionally assess the semantic

coverage of our Octree Language Map relative to open-vocabulary baselines. For the large-scale evaluation, we employ the SemanticKITTI urban driving dataset [39], demonstrating generalization capabilities of the proposed approach.

Unless otherwise noted, we use the parameters summarized in Table II. We use the lower index notation $\text{OMCL}_{\text{Mapping}\backslash\text{Localization}}$ to indicate the specific model used for mapping and localization, respectively. For instance, OMCL_{GT} means that ground truth semantic images are used for both mapping and localization; while $\text{OMCL}_{\text{Open}\backslash\text{LSeg}}$ means OpenScene [5] is used for mapping and LSeg [31] for localization. OMCL always operates in the continuous feature-embedding space, processing GT labels with a corresponding text encoder.

By default, we ground $\text{Features}_{\text{DB}}$ as described in Section III-A with a prompt consisting of the datasets' semantic class labels. For reference purposes, we provide evaluation for the ungrounded OMCL $^{\text{auto}}$ version. All results are obtained using an Nvidia RTX 3090 GPU equipped with 24GB of memory.

Parameter	Value	Parameter	Value
Discrepancy threshold τ	0.02	Language feature size F	512
Number of particles	1024	Surrounding radius R	2 [m]
Number of rays	2048	Similarity threshold ρ	0.9
Map resolution	0.02 [m]	Matching criterion k	500

Table II. OMCL Parameters.

A. Long-Term Indoor Localization

Localization Evaluation: Table I. We compare our approach with SeDAR [14] and SIMP [7] which, similar to us, employ MCL and measure semantic consistency for long-term indoor localization (Section II). For fair comparison, all approaches are evaluated on the same Octree Language Map which was created from ground truth data (GT) and projected to a floor plan map (2D) by averaging the data along the vertical dimension. We denote baselines adapted to the common map as SeDAR* and SIMP*. The same noisy odometry is applied to all methods: ground-truth poses are perturbed with zero-mean Gaussian noise with standard deviations of 0.10 m (translation) and 6 ° (rotation). For reference, the average per-step

ground truth motion is 0.13 m in translation and 8 ° in rotation. Particles are initialized around the starting pose with standard deviations of 0.3 m and 17 °. RTAB-Map [36] is employed as the evaluation baseline of our full pose estimation on the Octree Language Map in 3D. We use the RGB-D variant of RTAB-Map to avoid the degeneracies of monocular SLAM in low-parallax, in-place rotation segments widely present in the dataset.

	Accuracy [%]	**Precision [%]**	**Recall [%]**	**IoU [%]**
Matterport3D [37]				
$OMCL_{\mathrm{GT}}$	*86.75*	*78.96*	*78.95*	*67.46*
$\mathbf{OMCL}_{\mathrm{LSeg}}$	**82.74**	**59.74**	**54.30**	**43.35**
$\mathrm{OMCL}^{\mathrm{auto}}_{\mathrm{LSeg}}$	80.99	59.13	51.24	41.28
$\mathrm{OMCL}_{\mathrm{Open\backslash LSeg}}$	62.89	38.25	42.91	26.95
Replica [38]				
$\mathrm{OMCL}_{\mathrm{NAR}}$	96.53	23.87	46.8	17.1

Table III. Measurements-Map Consistency Evaluation.

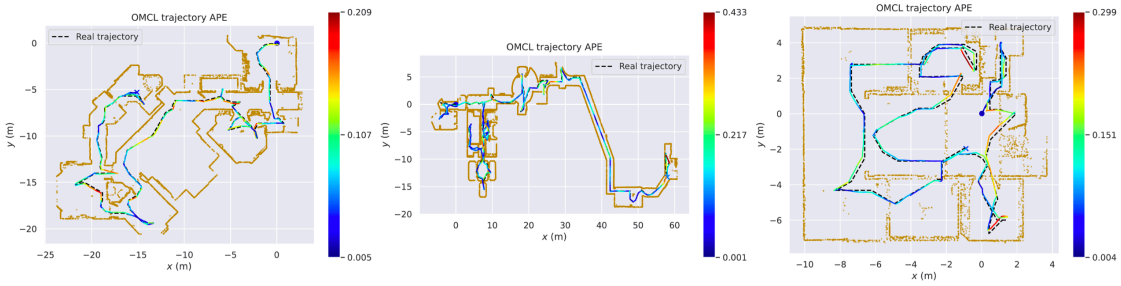


Figure 5. Example trajectories performed by OMCL on the Matterport3D dataset, with APE indicated by color for each segment. Map projections are shown in brown. The plots demonstrate the performance in scenarios that involve loopy paths, corridors, and long monotonic trajectories.

For completeness, we report the full APE statistics [40] in Table I, averaged over all sequences, and provide the corresponding measurement–map consistency evaluation in Table III. We evaluate both OMCL configurations shown in (Fig. 2). Under "Input Option 1", we use $\mathrm{OMCL}_{\mathrm{GT}}$ and $\mathrm{OMCL}_{\mathrm{LSeg}}$. Under "Input Option 2", $\mathrm{OMCL}_{\mathrm{Open\backslash LSeg}}$ employs the OpenScene [5] 3D distillation model for the mapping and, because OpenScene does not accept image input, we use LSeg for localization. This is feasible because both models operate in the same CLIP feature space.

Measurements-Map Consistency Evaluation: Table III. We report accuracy, precision, recall and IoU between the map and the observed features during localization, averaged over all poses and sequences for different datasets. To compute these metrics, we treat each feature stored in the Octree Language Map (i.e., each element of $\mathrm{Features}_{\mathrm{DB}}$) as an independent semantic class. For each per-pixel image feature, we identify its corresponding map feature via ray casting and count a correct correspondence if that map feature attains the highest cosine similarity among all features in $\mathrm{Features}_{\mathrm{DB}}$.

The evaluation quantifies a lower bound on consistency between the map and measurements during localization and assesses both the map's ability to preserve information and the image encoder's consistency across viewing directions. It is a lower-bound because correspondences are discretized as correct/incorrect, whereas localization operates in a continuous feature-similarity space. Note that localization depends solely on this consistency and not on the absolute correctness of the map's semantics.

Results and Comparison: Tables I and III. The metrics reported in Table III align with the localization results in Table I. The metrics of $\mathrm{OMCL}_{\mathrm{LSeg}}$ are close to those of $\mathrm{OMCL}_{\mathrm{GT}}$ and both outperform SeDAR $^{*}_{\mathrm{GT}}$ and $\mathrm{SIMP}^{*}_{\mathrm{GT}}$. $\mathrm{OMCL}_{\mathrm{Open\backslash LSeg}}$ lacks sufficient mapping consistency to complete all test sequences; nevertheless, it still outperforms the baselines and successfully executes most sequences. Moreover, $\mathrm{OMCL}_{\mathrm{Open\backslash LSeg}}$ can create the Octree Language Map directly from existing point clouds. The ungrounded $\mathrm{OMCL}^{\mathrm{auto}}_{\mathrm{LSeg}}$ achieves metrics comparable to $\mathrm{OMCL}_{\mathrm{LSeg}}$.

Larger gaps between the Mean and RMSE in Table I indicate more significant deviations from the real pose. The geometry-based RTAB-Map [36] loses tracking in narrow, feature-poor areas. Although it subsequently relocalizes, the accumulated errors prevent accurate recovery of the full trajectory. Fig. 5 illustrates the performance of OMCL in a variety of scenarios, including loopy paths, corridors, and monotonic trajectories.

	All		Tail		Camera pose / ATE RMSE [cm]
	mIoU	mAcc	mIoU	mAcc	
OpenScene [5]	15.9	24.6	1.5	6.7	N/A
ConceptGraphs [21]	16.7	33.7	4.4	26.8	N/A
RayFronts [28]	27.7	54.5	17.6	41.3	N/A
OVO [25]	27.1	39.1	12.1	19.6	**0.6 – 1.9**
OMCL$_{\mathrm{NAR}}$ (ours)	**32.1**	**56.2**	**21.4**	**43.9**	35

Table IV. Replica 3D Semantic And Localization Evaluation.

B. Datasets Generalization

Replica. The Replica dataset [38] contains photo-realistic 3D indoor scene reconstructions at room and building scale. Following the OVO [25] evaluation procedure, we estimate the semantic map classification quality and report localization accuracy on the resulting estimated map in Section IV-A. We employ the NARADIO [28] image and text encoder for mapping and localization. Ground-truth odometry is perturbed with zero-mean Gaussian noise at 20% of the per-step motion. The classification accuracies are reported for all classes (All) and for the most challenging, low-frequency one-third subset (Tail [25]). The camera poses in OVO [25] are estimated using Gaussian-SLAM [26] and ORB-SLAM2 [27], with ATE RMSEs of 0.6 cm and 1.9 cm, respectively.

OMCL$_{\mathrm{NAR}}$ achieves the best semantic mapping quality due to its high-resolution mapping, while the small improvement over similar RayFronts is likely due to implementation details and design choices. The largest improvement is observed in the less frequent classes (Tail). Although the measured semantic quality is high, the subsequent localization on the map is stable but not very accurate (about 35 cm). This correlates with the low measurement–map consistency metric estimated for the used dataset–encoder (Replica-NARADIO) pair in Table III. Replica is smaller but also semantically denser than the Matterport3D dataset, which makes it more challenging to perceive visual features on the map equally well from all viewing directions. Precision and IoU in Table III correlate with localization accuracy across the Matterport3D and Replica datasets.

	Translation [m]		Rotation [°]		Fine-tuning data
	Mean	Std	Mean	Std	
Stereo DSO [41]	7.23	5.0	1.99	0.86	**no training**
SemLoc [15]	1.49	–	–	–	other urban-driving
Zhang *et al.* [16]	0.58	–	–	–	KITTI
Pi-Long [42]	9.88	5.39	6.24	3.61	**no fine-tuning**
OMCL$_{\text{X-Dec}}$ (our)	0.52	0.26	1.77	0.26	**no fine-tuning**
CMRNext [4]	**0.11**	**0.06**	**0.25**	**0.15**	KITTI + other urban

Table V. ATE on Sequence 00 of the Kitti Dataset.

SemanticKITTI. SemanticKITTI [39] contains semantically annotated 3D LiDAR from urban driving. As in SemLoc, we rely on the reconstructed semantic point cloud and evaluate localization on existing maps. OMCL$_{\text{x-dec}}$ employs X-Decoder [32] as the visual localization model and language encoder, with odometry provided by Stereo DSO. We use 512 particles, 4096 rays per image, and an Octree Language map resolution of 0.05 m. Following CMRNext [4], we report the averaged localization metrics on Sequence 00 in Section IV-B.

OMCL$_{\text{X-Dec}}$ outperforms other semantic-consistency-based localization methods in accuracy, while requiring no fine-tuning, leading to a significant improvement in the estimated odometry. CMRNext is the state-of-the-art camera-to-LiDAR map matching approach, but it requires fine-tuning to the data and was trained on the remaining KITTI sequences. We also compare against the feed-forward Pi-Long approach [42] with loop closure, which, similar to ours, performs monocular localization and does not require data-specific fine-tuning.

C. Prompt-augmented Initialization

In order to assess the proposed initialization, we randomly distribute particles across the map and report the average number of OMCL steps required to achieve the designated localization accuracy in Table VI.

The single OMCL step corresponds to the weighing of the particles and then resampling. The prompts are randomly generated from the lists of probable words for each starting location, with each prompt containing three to five words. We evaluate on the sequences of Matterport3D dataset, and localization can be considered sufficient once it reaches 0.2 m, since this matches the accuracy of the subsequent pose tracking (Table I).

Table VI shows that the maximum number of steps needed for localization is 44, with fast convergence once 1 m accuracy is achieved. On average, localization requires 29 steps to determine the camera pose, and a minimum of 24 steps is needed to reach 0.1 m accuracy.

Enhancing global localization with the user prompt according to Section III-C allows us to accelerate localization convergence. $\mathrm{OMCL}_{\mathrm{LSeg}}^{0.1\,\mathrm{m}\ /\ \mathrm{prompt}}$ used 14 steps on average to achieve an accuracy of 0.1 m with the fastest case taking 3 steps, accelerating localization convergence by factors of 2.7 and 8, respectively.

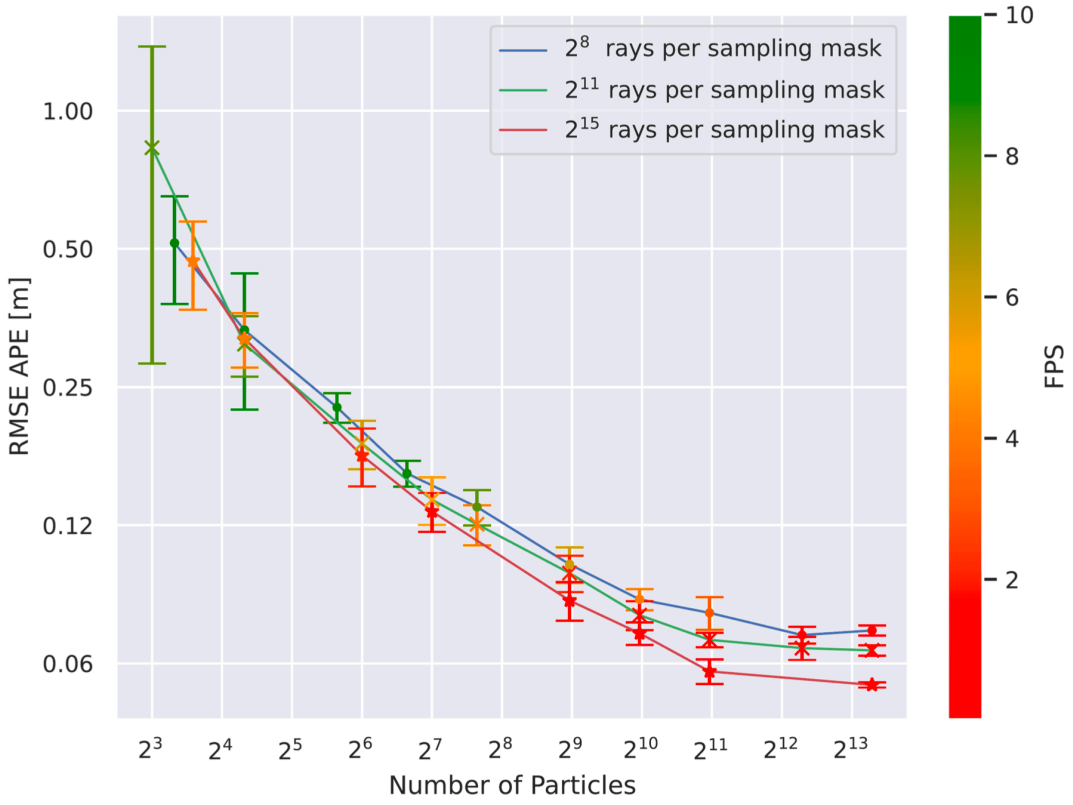


Figure 6. Impact of the number of particles on localization accuracy and FPS for varying numbers of sampled rays.

doi.org/10.32388/I8VL7Y

D. Ablation Studies

We demonstrate the impacts of different parameters and ray sampling strategies in Figs. 6 and 7. The plots are constructed by running one trajectory ten times with $\text{OMCL}_{\text{LSeg}}$. According to Fig. 7, our stratified ray sampling strategy (Section III-B) outperforms the one with an equal distribution of the rays among the observable categories in both the APE metric and the FPS for the high number of rays. Moreover, our implementation consumes less GPU memory, allowing it to handle a larger number of rays. Compared with uniform sampling over the full image, our stratified approach can work even with a small number of sampled rays because it tries to cover all different observable object instances. It tends to cover the full image without duplicated samples when increasing the number of rays. The APE variance does not decrease gradually with an increase in rays. Keeping the number of particles constant, using more rays improves the APE by about 10% compared to 2^8 rays.

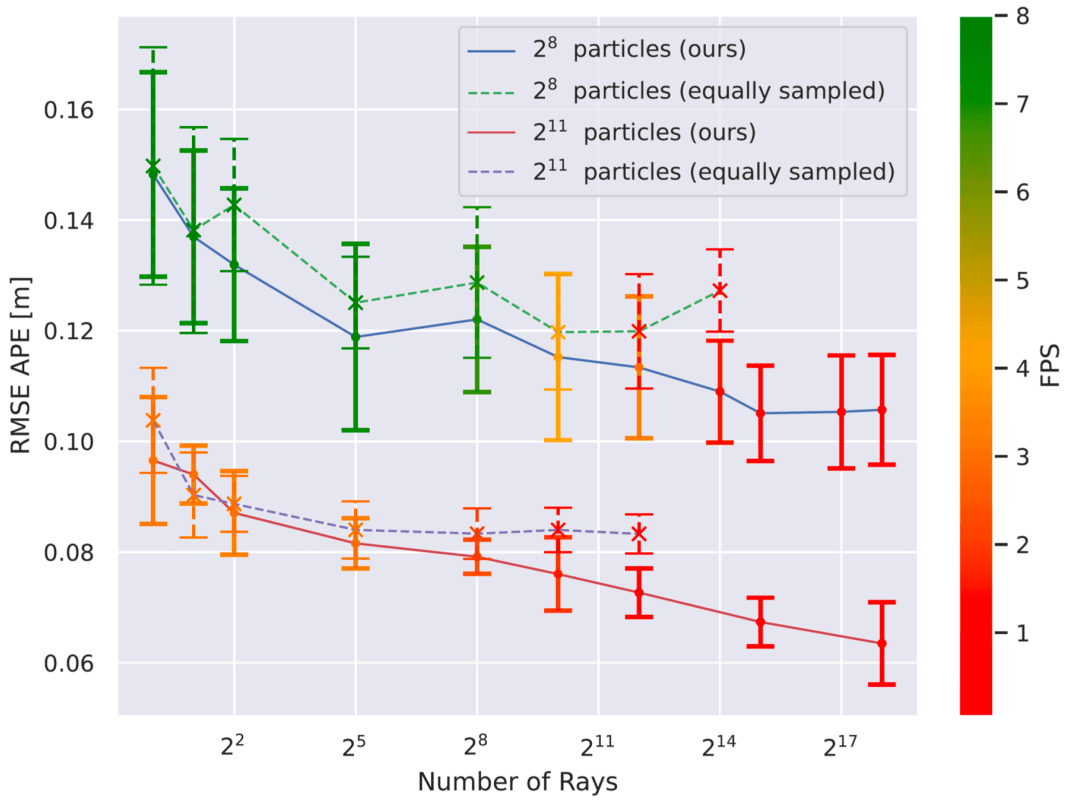


Figure 7. Comparison between our stratified ray sampling strategy and equal distribution of rays among observable categories, for different particle counts.

doi.org/10.32388/I8VL7Y

	mean [st.]	std [st.]	min [st.]	max [st.]
$\mathbf{OMCL}^{\mathbf{2\backslash,m}}_{\mathrm{LSeg}}$	19	8	11	39
$\mathbf{OMCL}^{\mathbf{1\backslash,m}}_{\mathrm{LSeg}}$	22	8	16	41
$\mathbf{OMCL}^{\mathbf{0.5\backslash,m}}_{\mathrm{LSeg}}$	25	8	16	42
$\mathbf{OMCL}^{\mathbf{0.2\backslash,m}}_{\mathrm{LSeg}}$	29	8	17	44
$\mathbf{OMCL}^{\mathbf{0.1\backslash,m}}_{\mathrm{LSeg}}$	38	7	24	45
$\mathbf{OMCL}^{\mathbf{0.1\backslash,m\ /\ prompt}}_{\mathrm{LSeg}}$	**14**	**5**	**3**	**23**

Table VI. Global Localization Steps Before Convergence.

The steps indicate the number of OMCL iterations needed to reach the specified localization accuracy.

Fig. 6 demonstrates that the increase in the number of particles reduces the variance of APE. It is noticeable that even with a small number of particles, OMCL can still perform pose estimation. In the beginning, the particles have a higher impact on APE compared to the rays (Fig. 7), but the rays have a smaller initial APE error. Increasing the number of particles beyond 2^{11} leads to a 10% improvement in APE, but the system may become less stable to the track loss. Together, Figs. 7 and 6 help to select parameters with corresponding APE and processed frames per second (FPS) to satisfy the application needs. We do not include the timing of the used visual model in the presented FPS values because it varies for different models and can be parallelized.

V. Conclusion

In this paper, we introduced OMCL, a framework for vision-based Monte Carlo localization in open-vocabulary 3D semantic maps. The maps are created from posed RGB-D images or 3D point clouds and store CLIP features in an octree, enabling cross-modal sensor setups for mapping and localization. A textual prompt can be used to initialize localization. OMCL computes the likelihood of localization hypotheses based on the consistency of open-vocabulary features extracted from the current input image and the corresponding map features that are retrieved by ray casting. OMCL is flexible with respect to the

doi.org/10.32388/I8VL7Y

choice of the feature extractor and benefits from stratified ray sampling. Our method generalizes across room-scale indoor, large-scale indoor, and outdoor environments. We provide a measurements-map consistency analysis and ablation studies of the proposed framework.

OMCL lacks online map updates and requires map-scaled odometry. Future work on tighter integration with a SLAM backend could potentially address both issues.

Acknowledgments

This work has been partially funded by the Federal Ministry of Research, Technology and Space of Germany (BMFTR) under grant no. 01IS22094A WestAI and within the Robotics Institute Germany, grant no. 16ME0999.

References

1. ^*Krombach N, Droeschel D, Houben S, Behnke S (2018). "Feature-based Visual Odometry Prior for Real-time Semi-dense Stereo SLAM." Robot Auton Syst. **109**:38–58.*
2. ^*Droeschel D, Behnke S (2018). "Efficient Continuous-Time SLAM for 3D Lidar-Based Online Mapping."*
3. ^*Cao H, Behnke S (2024). "SLCF-Net: Sequential LiDAR-Camera Fusion for Semantic Scene Completion usin g a 3D Recurrent U-Net."*
4. a, b, c, d*Cattaneo D, Valada A (2025). "CMRNext: Camera to LiDAR Matching in the Wild for Localization an d Extrinsic Calibration." IEEE Trans Robot. **41**:1995–2013. doi:10.1109/TRO.2025.3546784.*
5. a, b, c, d, e, f, g, h, i, j, k, l, m*Peng S, Genova K, others (2023). "OpenScene: 3D scene understanding with open vo cabularies."*
6. a, b, c*Huang C, Mees O, Zeng A, Burgard W (2023). "Visual language maps for robot navigation."*
7. a, b, c, d, e*Zimmerman N, Sodano M, Marks E, Behley J, Stachniss C (2023). "Constructing Metric-Semantic M aps using Floor Plan Priors for Long-Term Indoor Localization."*
8. ^*Yu C, Liu Z, Liu X, Xie F, Yang Y, others (2018). "DS-SLAM: A Semantic Visual SLAM towards Dynamic Envir onments."*
9. ^*Muhammad N, Paxton C, Pinto L, Chintala S, Szlam A (2023). "CLIP-Fields: Weakly Supervised Semantic Fi elds for Robotic Memory."*
10. a, b, c, d, e, f, g, h, i, j*Radford A, Kim JW, Hallacy C, Ramesh A, others (2021). "Learning Transferable Visual Mo dels From Natural Language Supervision."*

11. ^Yudin E, Karpyshev P, others (2023). "CloudVision: DNN-based Visual Localization of Autonomous Robots using Prebuilt LiDAR Point Cloud."
12. ^Zhang L, Tao Y, Lin J, Zhang F, Fallon M (2024). "Visual Localization in 3D Maps: Comparing Point Cloud, Mesh, and NeRF Representations." arXiv:2408.11966.
13. ^Maggio D, Abate M, Shi J, Mario C, Carlone L (2023). "Loc-NeRF: Monte Carlo Localization using Neural Radiance Fields."
14. a, b, c, dMendez O, Hadfield S, Pugeault N, Bowden R (2018). "SeDAR-Semantic Detection and Ranging: Humans can localise without LiDAR, can robots?."
15. a, bLiang S, Zhang Y, Tian R, Zhu D, Yang L, Cao Z (2022). "SemLoc: Accurate and Robust Visual Localization with Semantic and Structural Constraints from Prior Maps."
16. a, bZhang C, Zhao H, Wang C, others (2023). "Cross-modal monocular localization in prior LiDAR maps utilizing semantic consistency."
17. ^Zhao B, Yang L, Mao M, Bao H, Cui Z (2024). "PNeRFLoc: Visual Localization with Point-based Neural Radiance Fields."
18. ^Atanasov N, Zhu M, Daniilidis K, Pappas G (2016). "Localization from semantic observations via the matrix permanent." Int J Robot Res. **35**(1-3):73–99.
19. ^Yang Y, Wu X, He T, Zhao H, Liu X (2023). "SAM3D: Segment Anything in 3D Scenes." arXiv:2306.03908.
20. ^Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, others (2023). "Segment Anything."
21. a, bGu Q, Kuwajerwala A, Morin S, others (2024). "ConceptGraphs: Open-Vocabulary 3D Scene graphs for Perception and Planning."
22. ^Jatavallabhula KM, Kuwajerwala A, Gu Q, Omama M, Iyer G, Saryazdi S, Chen T, others (2023). "ConceptFusion: Open-set multimodal 3D mapping."
23. ^Jatavallabhula K, Iyer G, Paull L (2020). "gradSLAM: Dense SLAM meets Automatic Differentiation."
24. ^Kruzhkov E, Behnke S (2025). "LiLMaps: Learnable Implicit Language Maps."
25. a, b, c, d, eMartins T, Oswald M, Civera J (2025). "Open-Vocabulary Online Semantic Mapping for SLAM." IEEE Robot Autom Lett. **10**(11):11745–11752.
26. a, bYugay V, Li Y, Gevers T, Oswald M (2023). "Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting." arXiv:2312.10070.
27. a, bMur-Artal R, Tardós J (2017). "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras." IEEE Trans Robot. **33**(5):1255–1262.

28. a, b, c, d, e *Alama O, Bhattacharya A, He H, Kim S, Qiu Y, Wang W, Ho C, Keetha N, others (2025). "RayFronts: Open-Set Semantic Ray Frontiers for Online Scene Understanding and Exploration." arXiv:2504.06994.*

29. ^ *McCormac J, Clark R, Bloesch M, Davison A, Leutenegger S (2018). "Fusion++: Volumetric Object-Level SLAM."*

30. ^ *Stückler J, Biresev N, Behnke S (2012). "Semantic Mapping Using Object-Class Segmentation of RGB-D Images."*

31. a, b, c, d, e, f, g, h, i *Li B, Weinberger KQ, Belongie SJ, Koltun V, Ranftl R (2022). "Language-driven Semantic Segmentation."*

32. a, b, c *Zou X, Dou Z, Yang J, Gan Z, Li L, Li C, Dai X, Behl H, Wang J, Yuan L, others (2022). "Generalized Decoding for Pixel, Image, and Language."*

33. ^ *Chen B, Xia F, Ichter B, Rao K, others (2023). "Open-vocabulary Queryable Scene Representations for Real World Planning."*

34. ^ *Ha H, Song S (2022). "Semantic Abstraction: Open-World 3D Scene Understanding from 2D Vision-Language Models."*

35. a, b *Jatavallabhula KM, Smith EJ, Lafleche J, Tsang CF, Rozantsev A, others (2019). "Kaolin: A PyTorch Library for Accelerating 3D Deep Learning Research." CoRR. abs/1911.05063.*

36. a, b, c *Labbé M, Michaud F (2019). "RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation." J Field Robot. **36**(2):416–446.*

37. a, b *Chang A, Dai A, Funkhouser T, Halber M, Niessner M, others (2017). "Matterport3D: Learning from RGB-D Data in Indoor Environments." Int Conf 3D Vis.*

38. a, b, c *Straub J, Whelan T, Ma L, Chen Y, Wijmans E, Green S, Engel J, Mur-Artal R, Ren C, Verma S, others (2019). "The Replica Dataset: A Digital Replica of Indoor Spaces." arXiv:1906.05797.*

39. a, b *Behley J, Garbade M, Milioto A, Quenzel J, Behnke S, Gall J, Stachniss C (2021). "Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset." Int J Robot Res. **40**(8-9).*

40. ^ *Grupp M (2017). "evo: Python package for the evaluation of odometry and SLAM."*

41. ^ *Wang R, Schworer M, Cremers D (2017). "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras."*

42. a, b *Deng K, Ti Z, Xu J, Yang J, Xie J (2025). "VGGT-Long: Chunk it, Loop it, Align it – Pushing VGGT's Limits on Kilometer-scale Long RGB Sequences." https://arxiv.org/abs/2507.16443.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.