2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

CLIP-Loc: Multi-modal Landmark Association for Global Localization in Object-based Maps

Shigemichi Matsuzaki, Takuma Sugino, Kazuhito Tanaka, Zijun Sha,
Shintaro Nakaoka, Shintaro Yoshizawa, and Kazuhiro Shintani

***Abstract*— This paper describes a multi-modal data association method for global localization using object-based maps and camera images. In global localization, or relocalization, using object-based maps, existing methods typically resort to matching all possible combinations of detected objects and landmarks with the same object category, followed by inlier extraction using RANSAC or brute-force search. This approach becomes infeasible as the number of landmarks increases due to the exponential growth of correspondence candidates. In this paper, we propose labeling landmarks with natural language descriptions and extracting correspondences based on conceptual similarity with image observations using a Vision Language Model (VLM). By leveraging detailed text information, our approach efficiently extracts correspondences compared to methods using only object categories. Through experiments, we demonstrate that the proposed method enables more accurate global localization with fewer iterations compared to baseline methods, exhibiting its efficiency.**

I. Introduction

Global localization is a fundamental ability of mobile robots where a pose of a robot is estimated given a sensor observation without prior information of its ego-location. It is used, e.g., for wake-up in a prior map, and recovery from localization failures. Monocular cameras are a popular choice as a sensor for their nature of low cost and light weight, as well as the rich visual information available.

One of the most popular approaches to visual localization is to use a feature map built by visual SLAM (Simultaneous Localization and Mapping) systems, such as ORB-SLAM series [1, 2, 3]. These methods calculate corresponding points between images and maps based on the feature similarity, and then compute the camera pose. Local features capture the distribution of local pixel values viewed from a certain viewpoint; therefore, changes in the viewpoint and camera resolution during map generation and localization lead to inaccurate correspondence matching and localization.

To address these challenges, some studies proposed to use semantic objects as landmarks, and image-based object detection as observation [4, 5]. Particularly, deep learning-based object detectors are usually used for their robustness to viewpoint and resolution changes. Leveraging the characteristics, object-based localization has been shown to be robust to those changes [6, 7]. Object-based maps also provide rich semantic information that can be exploited in, e.g., manipulation [7] and high-level task planning [8].

Authors are with Frontier Research Center, Toyota Motor Corporation, Toyota, Aichi, Japan.
shigemichi_matsuzaki@mail.toyota.co.jp

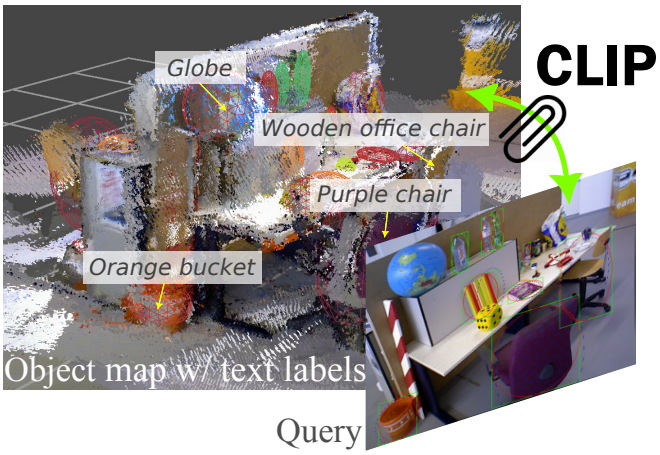


Fig. 1. In conventional global localization in object-based maps, only class information is utilized for constructing a correspondence candidate set, leading to exponential growth of the number as the landmark increases. In this paper, we propose to assign a natural language description as a label to each landmark, and match them with visual observation using a Vision Language Model (VLM). It enables more efficient correspondence matching leveraging the fine-grained information given by the text labels.

Relocalization in object-based maps has not been actively tackled in the existing work. Most methods implement a simple iterative correspondence matching via Random Sample Consensus (RANSAC) [9] where all possible pairs of a landmark and an observation with the same object class are treated as correspondence candidates [6, 10]. This method is not feasible when the number of landmarks is large due to exponential growth of the number of candidates. Moreover, the existing method treats all the candidates equally. In practice, there may be candidates that are almost certainly inliers based on appearance. Those should be favored more in the iterative sampling procedure for more efficient calculation.

With the aforementioned discussions in mind, in this work, we present a novel method of landmark association for global localization in object-based maps. First, we propose a method of multi-modal landmark association leveraging an off-the-shelf vision and language model (VLM), namely CLIP [11]. We assign each landmark a text label describing its appearance and match them with visual observations via the multi-modal embedding of CLIP. Utilizing rich information given by natural language labels, the proposed method is able to establish a correspondence candidate set with only landmark-observation pairs that share similar concepts, leading to fewer outliers and thus more efficient inlier extraction. Second, we introduce an efficient iterative inlier extraction

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

based on Progressive Sample Consensus (PROSAC) [12], where correspondence candidates more likely to be inliers are sampled more frequently. We use the similarity score of the text and vision embeddings from CLIP as a sampling weight for a correspondence candidate.

The main contributions of this work are as follows:

- We propose a method to associate object landmarks with image observations by utilizing natural language labels and a VLM for efficient global localization.
- We introduce an efficient sampling method inspired by PROSAC [12] using text-image similarity of correspondence candidates as a score for efficient computation.

II. Related Work

A. Object-based SLAM and relocalization

Object-based SLAM is a subfield of SLAM where a landmark is represented as an object instance in a form of CAD models [13], or primitive shapes (cubes [5], planes [14], quadrics [4, 6, 10, 15], superquadrics [16], etc.). It has been claimed that the object-based SLAM has some advantages over feature-based visual SLAM methods, such as rich semantic information of objects, and robustness against view point changes [6], Especially, quadric-based object mapping has been actively studied for its compact representation and a feature that projections of quadrics onto an image plane can be efficiently calculated.

Although such object-based SLAM methods have been studied, object-based relocalization is relatively under-explored. As the object-based map lacks discriminative information and only provides the object category, the size, and so on, it has to rely on iterative methods such as RANSAC to identify a set of inlier correspondences from all possible matching candidates assigned with the same category, which is not scalable to larger maps with more landmarks.

OA-SLAM by Zins et al. [6] combines a feature-based SLAM (ORB-SLAM2 [2]) and the object-based landmark representation for relocalization more robust to viewpoint changes. However, the feature information is not exploited in the object-based relocalization and simple RANSAC is used. Zins et al. [10] also proposed a camera pose estimation based on a level-set metric that quantifies the alignment of the ellipses and the projections of corresponding ellipsoids. This, however, also depends on pose initialization by RANSAC. Wu et al. [7] proposed a graph-based scene matching for relocalization in object-centric maps.

Unlike the existing work, we use natural language labeling and rich multi-modal information from CLIP to reinforce the discriminativity of object landmarks and thus to improve the efficiency of object-based global localization.

B. Vision Language Models

Driven by the recent progress of Transformer [17]-based networks, so-called *foundation models* have been proposed in a variety of tasks [18, 19]. Vision Language Models (VLMs) are a type of models pre-trained with large-scale image-text pairs so that they can be applied to visual and linguistic inference tasks. CLIP [11] is a representative VLM that can ground visual concepts to natural language instructions. CLIP has been applied to various applications with open vocabulary scene inference such as visual navigation [20, 21], visual place recognition [22], etc.

The work by Mirjalili et al. [22] is the most related to our work in a way that they use a VLM for visual localization. In [22], a per-image descriptor is constructed exploiting CLIP and GPT-3 and conduct image retrieval from pre-built image database. The proposed method differs in that it is specifically tailored for localization in object-based maps by directly associating the visual observations with the landmarks.

C. Random sampling strategies

Random Sample Consensus (RANSAC) [9] is a commonly used model estimator for robustly estimating model parameters using a dataset that contains a certain percentage of outliers by iterating a procedure of data sampling, parameter estimation, and verification. In the context of correspondence matching, the dataset is a set of correspondence hypotheses. The original RANSAC uniformly samples candidates in each iteration. Progressive Sample Consensus (PROSAC) [12] is a variant of RANSAC where the quality of each datum is measured, and the data are sampled according to the quality score to favor more promising samples. Although some other sampling strategies have been proposed, e.g., MAGSAC++ [23], we adopt PROSAC for its simplicity of implementation and the suitability to our task where correspondence candidates are naturally assigned with a similarity score between image and text embeddings.

III. Proposed Method

A. Problem formulation

We assume an object map represented as ellipsoidal landmarks with text labels in a form of natural language descriptions. Text labels can be given by any methods such as manual input, voice dictation, or image captioning, e.g., [24]. As a query, a single image is given. Bounding boxes of objects in the image are detected using an arbitrary object detector and used as observations.

Formally, a map $\mathcal{M}$ consists of a set of N_m object landmarks $\{\mathbf{Q}_i^*, l_i\}_{i=1}^{N_m}$, where $\mathbf{Q}_i^*$ and l_i denote a dual form of a quadric representing the landmark, and a text label assigned to the landmark, respectively. A query image is denoted as I, and detected object bounding boxes are denoted as $\{\mathbf{b}_j\}_{j=1}^{N_o}$, where N_o is the number of detected objects.

B. Overview of the algorithm

The overview of the method is shown in Fig. 2, and pseudo-code is shown in Algorithm 1. The process of the proposed method consists of three steps: 1) correspondence candidate generation, 2) inlier correspondence extraction, and 3) pose estimation. In correspondence candidate generation, observation-landmark matches are searched for based on the similarity of observations and text-labeled landmarks given by CLIP. Inlier correspondences are then extracted via PROSAC [12]-like weighted sampling, and verification based

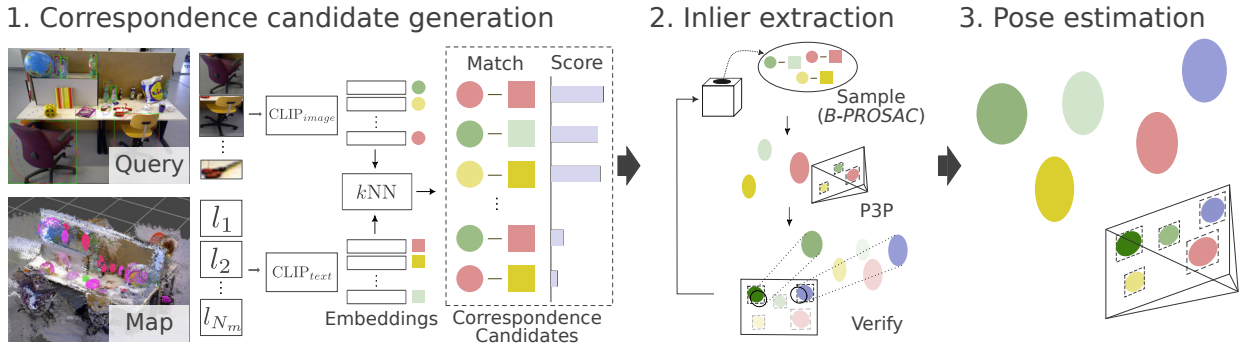


Fig. 2. Overview of the proposed method. We assume that object landmarks in the map are labeled by natural language descriptions. In addition, for a query image, object locations and classes are given by a general object detector (here we use YOLOv8). Given those data, a camera pose is estimated in the following steps: 1) The query objects and the landmark labels are embedded to a common feature space using image and text encoders of CLIP [11], respectively. For each image embedding, k nearest neighboring text embeddings are retrieved, and correspondence candidates are generated. 2) Inlier correspondences are estimated through B-PROSAC described in Sec. III-D. It consists of sampling three candidates considering the matching score, pose estimation via P3P, and pose verification. The final pose is given as the one with the best score after a certain number of iterations. 3) Optionally, the pose can be further refined by methods, e.g., [10].

on overlap of observations and the projected corresponding landmarks. The solution is given as the one with the best score. Finally, the camera pose can be further refined using the best correspondences.

C. CLIP-based correspondence candidate generation

As an offline pre-processing, a text label for each map landmark is encoded by the text encoder of CLIP and stored as a set of embeddings $\mathcal{F}^m = \{f_i^m\}_{i=1}^{N_m}$, where

$$f_i^m = \mathrm{CLIP}_{text}\left(l_i\right). \tag{1}$$

When an image I is given, object regions are extracted by the object detector. A visual embedding f_j for an image cropped by bounding box $\mathbf{b}_j$ is extracted by CLIP image encoder $\mathrm{CLIP}_{image}\left(\cdot\right)$:

$$f_j^o = \mathrm{CLIP}_{image}\left(\mathrm{crop}\left(\mathbf{b}_j, I\right)\right), \tag{2}$$

where $\mathrm{crop}\left(\cdot,\cdot\right)$ denotes a function to return a given image cropped with a given bounding box.

After getting the image embeddings, for each image embedding f_j^o, we retrieve k nearest neighboring text embeddings (kNN) in the embedding space:

$$\mathcal{F}_j^{m*} = k\mathrm{NN}\left(\mathcal{F}^m, f_j^o, k\right). \tag{3}$$

A set of correspondence candidates with observation $\mathbf{b}_j$ is given as a Cartesian product ($\times$):

$$\mathcal{C}_j = \{f_j^o\} \times \mathcal{F}_j^{m*}. \tag{4}$$

The whole set of correspondence candidates is thus given as:

$$\mathcal{C} = \bigcup_{j=1}^{N_o} \mathcal{C}_j. \tag{5}$$

D. Efficient inlier extraction

1) Sampling method: Intuitively, the higher the similarity score between an observation and a landmark, the more likely that they are an inlier correspondence. PROSAC [12] is based on such an intuition. In PROSAC, the correspondence candidates are sorted by some measure of the quality of matching. Correspondence hypotheses are sampled from only promising candidates in the earlier iterations, and the sampling strategy gradually shifts towards the original RANSAC as the iteration progresses. For details, the readers are referred to [12].

In our problem, the matching score can be naturally acquired as a cosine similarity between the image and the text embeddings, and thus PROSAC is suitable to the problem. We, however, found that sorting simply by the cosine similarity led to imbalanced samples biased to detections with larger regions of interest (ROI). We, therefore, sort the candidates such that the observations included in the candidates are balanced. The nearest landmarks for the observations are first sorted in the descending order of the similarity scores. The second-nearest landmarks are sorted next and stored, and so forth (see Fig. 3). After sorting, the inliers are estimated in the same procedure as PROSAC. We refer to the modified version of PROSAC as B-PROSAC (Balanced-PROSAC).

2) Pose verification: In each iteration of PROSAC/B-PROSAC, a camera pose is calculated using the sampled correspondence candidates and verified based on consistency of other observations with potentially corresponding landmarks. For simplicity of sampling, we sample three correspondences and calculate camera poses by the Perspective-3-Point (P3P) problem. In P3P, up to four camera poses are yielded. The most likely pose is chosen by evaluating the overlaps between the observations and the projected ellipses of the corresponding ellipsoid landmarks.

Once the pose is determined, correspondences for observations not in the samples are estimated by projecting potentially corresponding landmarks onto the image from the pose. For an observation $\mathbf{b}_j$, a landmark is considered as a correspondence if the IoU of an ellipsoid fit to the bounding box and the projection of the landmark is larger than a threshold, and that of the ellipsoid and the projection of any other potentially corresponding landmarks. The pose is scored with the sum of the IoUs for the observations.

After the pre-determined number of iterations, the final pose is given as the one with the best score. Optionally, the

Algorithm 1 CLIP-Loc

Input: Map instance set $\mathcal{M} = \{Q_i, l_i\}_{i=1}^{N_m}$, query instance set $\mathcal{Q} = \{q_j\}_{j=1}^{N_q}$, the number of loops N, the number of nearest neighbors k to search for each observation
Output: Camera pose $\mathbf{p}$, correspondence set $\mathcal{C}^*$

Text and image embedding via CLIP
$\mathcal{F}_m, \mathcal{F}_q = \text{CLIP}(\mathcal{M}, \mathcal{Q})$
Correspondence candidate set
$\mathcal{F}_m^* = \text{kNN}(\mathcal{F}_m, \mathcal{F}_q, k)$
$\mathcal{C} = \mathcal{F}_m^* \times \mathcal{F}_q$
Sort $\mathcal{C}$ by the method described in Sec. III-D.1

PROSAC loop
$max\_score = -1$
for i in range(N) **do**
  # Get a sample set $\mathbf{s}$ by PROSAC sampling [12]
  $\mathbf{s} = \text{prosac\_sample}(\mathcal{C})$
  # Calculate a camera pose via P3P
  $\mathbf{p}_{tmp} = \text{P3P}(\mathbf{s})$
  # Calculate $score$, and estimate potential correspondence set $\hat{\mathcal{C}}$
  $score, \hat{\mathcal{C}} = \text{verify}(\mathbf{p}_{tmp}, \mathcal{M}, \mathcal{Q})$
  if $score > max\_score$ **then**
    $\mathcal{C}^* \leftarrow \hat{\mathcal{C}}$
    $max\_score \leftarrow score$
    $\mathbf{p} \leftarrow \mathbf{p}_{tmp}$
  end if
end for
return $\mathbf{p}$, $\mathcal{C}^*$

pose can be further refined by methods, e.g., [10].

Landmarks used for pose verification We found that CLIP prone to wrongly estimate the matching for small observations. Using those erroneous candidates in verification leads to unreliable results. Therefore, we introduce a "hybrid" matching strategy where both CLIP-based correspondences and those based on the object class information are used for pose verification. By this way, we can reduce outlier candidates leveraging CLIP, while still utilizing the class information, which is more reliable for small objects.

IV. Experiments

A. Setup

1) Datasets: We evaluated our method with two datasets. The first one is TUM dataset [25]. We use *fr2/desk* and *fr3/long_office_household* sequences for evaluation. The sequences include 60 and 52 frames, respectively, extracted by sampling every 50 frames from the original image sets. For both datasets, we reconstructed 3D point cloud of the scenes, manually fitted an ellipsoid, and assigned a text label to each object using a self-made GUI tool to build consistent maps.

2) Implementation details: To acquire image observations, we used YOLOv8-x model[1]trained with COCO dataset [26]. We saved the predictions of the model with confidence higher than 0.1 for the test images offline and used them in all experiments for fair comparison among the different algorithms and trials. As an implementation of CLIP [11], we used ViT-L/14 model shared by OpenAI [2].

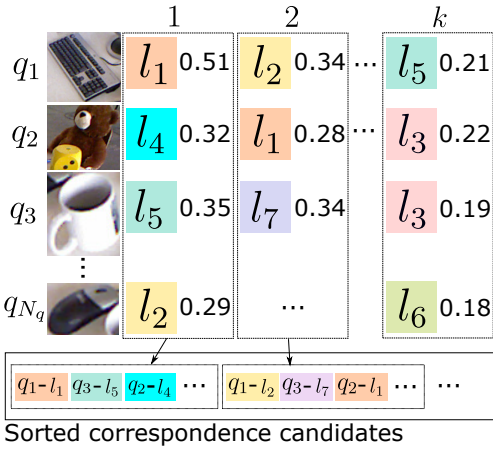


Fig. 3. Sorting in the proposed method. Top 1 nearest landmarks for the observations are first sorted by the score and stored in an ordered correspondence candidate list. Top 2 landmarks are then sorted and concatenated to the list, and so forth.

All algorithms were implemented in Python 3 mainly for integration with official implementations of YOLOv8 and CLIP, and without any parallelization. Although the implementation in pure Python is not optimal from a practical point of view, the algorithm is parallelizable and the computational efficiency can be further improved. Here, we focused on evaluation of the efficiency of the proposed algorithm relative to the baselines. In addition, we report the results without pose refinement using the found correspondences for all methods. The experiments were conducted on a computer with an RTX 4090 and Intel Core i9.

For the proposed correspondence candidate generation, the parameter k (the number of nearest text embeddings searched for each observation) is empirically set to 3.

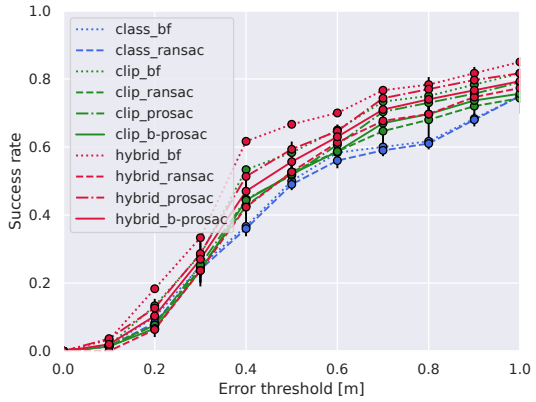
3) Baselines: We compare different combinations of matching types for generating correspondence candidates, and algorithms for inlier extraction.

Matching types We compare three matching types, namely *class*, *clip*, and *hybrid*. *class*: Pairs of an observation and a landmark with the same category are considered correspondence candidates and the same set is used for pose verification. *clip*: Correspondence candidates generated by the method described in Sec. III-C are used for both pose calculation and pose verification. *hybrid*: The same correspondence candidates as *clip* are used for pose calculation, while those of both *clip* and *class* are used for pose verification, as described in Sec. III-D.2.

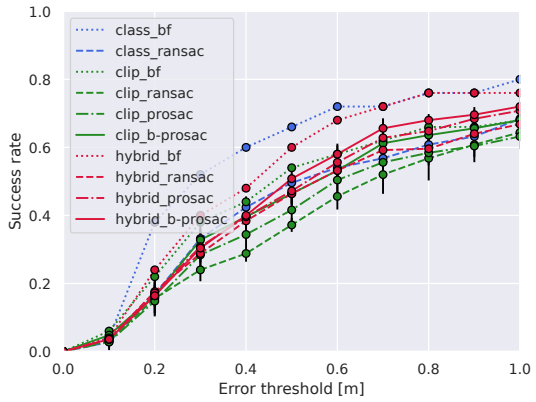
Algorithms We compare four algorithms, namely *bf*, *ransac*, *prosac*, and *b-prosac*. *bf*: Brute force search over all possible combinations of candidates. *ransac*: Ordinary RANSAC [9]. *prosac*: PROSAC [12] considering only the score. *b-prosac*: B-PROSAC described in III-D.1 considering the balance

[1]https://github.com/ultralytics/ultralytics

[2]https://github.com/openai/CLIP

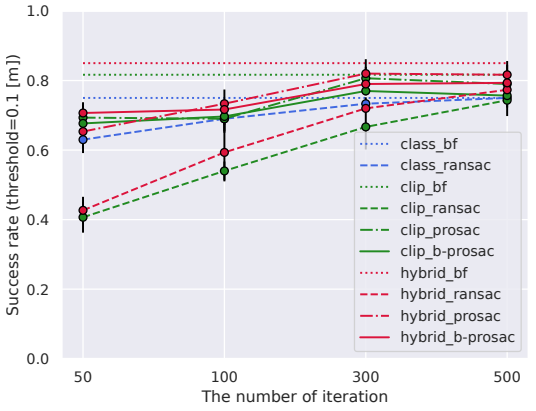


(a) *fr2*

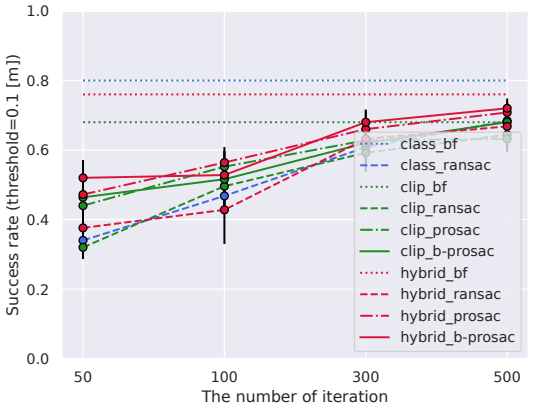


(b) *fr3*

Fig. 4. Success rate with regard to error thresholds when the number of iteration is 500. The proposed methods (*hybrid_prosac*/*hybrid_b-prosac*) outperformed the category-based baseline.



(a) *fr2*



(b) *fr3*

Fig. 5. Success rate with different numbers of iterations. *hybrid_prosac* and *hybrid_b-prosac* consistently outperformed the counterparts relying on category information and RANSAC.

among the observations.

We hereafter denote the methods by a notation of "{matching type}_{algorithm}", such as *hybrid_b-prosac*. Note that *class_bf* is used in the official implementation [3]of OA-SLAM [6], and *class_ransac* is used in major object-based SLAM systems such as [10].

4) Evaluation metrics: We evaluate the results with two metrics: *success rate* calculated as the ratio of test queries for which a pose is estimated with an error less than a threshold against all test queries, *translation error* (TE), and *rotation error* (RE) between the estimated and the ground-truth poses.

B. Comparative studies

The comparative results with the baselines with the number of iterations $N = 500$ are shown in Table I, and Fig. 4 shows the success rate over the sequences with respect to translation error thresholds. We report the average and standard deviation of five trials for each method.

Comparison of matching types Hybrid matching with PROSAC/B-PROSAC resulted in better performance than the object class-based baselines in both datasets. *hybrid_prosac* and *hybrid_b-prosac* even outperformed *class_bf*. This is because *hybrid* focuses only on promising candidates and thus reduces the outlier ratio. The proposed correspondence generation enables efficient correspondence search leveraging the fine-grained information of text labels and the VLM. *clip* resulted in worse performance than *hybrid*, justifying the proposed matching strategy.

In *fr2*, *class_bf* is a reasonable choice because of the moderate number of landmarks, but it soon became infeasible in *fr3* requiring more than 8 minutes on average. On the other hand, the proposed matching strategy makes the brute force algorithm feasible also on *fr3* with only a few seconds.

Comparison of algorithms B-PROSAC or PROSAC in combination with *hybrid* matching type consistently resulted in better performance with more efficient computation compared to RANSAC-based counterparts, effectively exploiting the similarity scores given by the proposed CLIP-based correspondence matching. To take a closer look at the effect of PROSAC / B-PROSAC, we compare the success rates with different numbers of iterations, shown in Fig. 5. *hybrid_b-prosac* resulted in better success rate than other algorithms and matching types even with fewer iterations, exhibiting the efficiency of the proposed method.

Fig. 6 shows qualitative results of the proposed *hybrid_b-prosac* with the iteration number of 500.

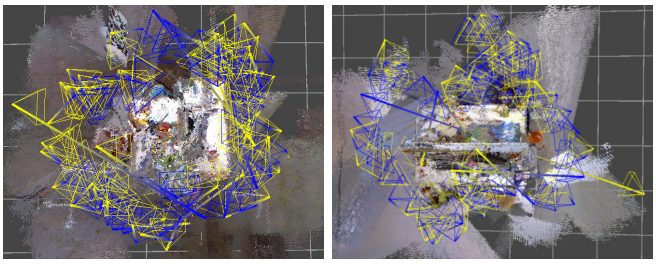
[3]https://gitlab.inria.fr/tangram/oa-slam

TABLE I

COMPARISON WITH THE BASELINES

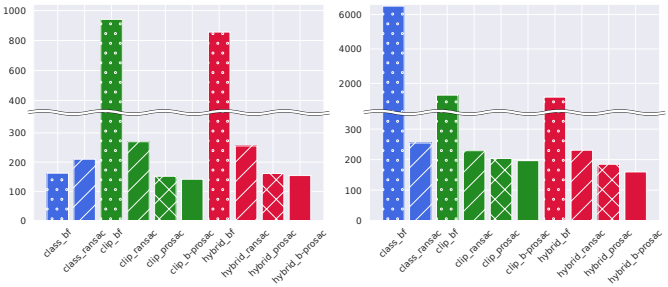
		fr2				*fr3*			
		SR [%]↑	TE [m]↓	RE [rad] ↓	Time [sec] ↓	SR [%]↑	TE [m]↓	RE [rad]↓	Time [sec] ↓
class	bf	75.0 ± 0.0	0.970 ± 0.00	0.726 ± 0.00	0.246 ± 0.20	80.0 ± 0.0	0.749 ± 0.00	0.638 ± 0.00	212 ± 493
	ransac	75.6 ± 2.1	0.987 ± 0.05	0.738 ± 0.04	$\underline{0.289 \pm 0.04}$	64.7 ± 2.5	1.045 ± 0.05	0.826 ± 0.02	0.519 ± 0.16
clip	bf	80.0 ± 0.0	0.778 ± 0.00	0.538 ± 0.00	4.795 ± 1.22	68.0 ± 0.0	1.020 ± 0.00	0.821 ± 0.00	4.933 ± 1.44
	ransac	75.6 ± 2.1	0.987 ± 0.05	0.738 ± 0.04	$\mathbf{0.288 \pm 0.04}$	63.3 ± 0.9	1.126 ± 0.01	0.941 ± 0.04	0.486 ± 0.06
	prosac	75.0 ± 2.4	0.836 ± 0.03	0.566 ± 0.02	0.458 ± 0.04	67.3 ± 4.7	1.053 ± 0.10	0.862 ± 0.09	$\underline{0.470 \pm 0.06}$
	b-prosac	76.7 ± 3.6	0.828 ± 0.08	0.595 ± 0.09	0.455 ± 0.04	64.7 ± 0.9	1.173 ± 0.04	0.869 ± 0.01	$\mathbf{0.469 \pm 0.07}$
hybrid	bf	85.0 ± 0.0	0.609 ± 0.00	0.363 ± 0.00	5.851 ± 1.52	76.0 ± 0.0	0.796 ± 0.00	0.619 ± 0.00	8.113 ± 2.61
	ransac	74.4 ± 2.8	0.824 ± 0.05	0.570 ± 0.04	0.601 ± 0.06	$\underline{71.3 \pm 2.5}$	0.952 ± 0.07	$\mathbf{0.694 \pm 0.07}$	0.873 ± 0.21
	prosac	$\underline{80.0 \pm 0.0}$	$\mathbf{0.75 \pm 0.07}$	$\mathbf{0.491 \pm 0.05}$	0.574 ± 0.06	70.7 ± 1.9	$\mathbf{0.889 \pm 0.09}$	$\underline{0.707 \pm 0.09}$	0.846 ± 0.21
	b-prosac	$\mathbf{81.7 \pm 2.7}$	$\underline{0.789 \pm 0.04}$	$\underline{0.520 \pm 0.01}$	0.577 ± 0.06	$\mathbf{72.7 \pm 5.7}$	$\underline{0.937 \pm 0.16}$	0.724 ± 0.09	0.852 ± 0.21

Bold and underline denote the best and the second-best among the sampling base algorithms (*ransac*, *prosac*, *b-prosac*), respectively.



(a) *fr2* (b) *fr3*

Fig. 6. Top view of the estimation results by *hybrid_b-prosac*. Estimated frames and groundtruth frames are in yellow and blue, respectively, and an estimation and its groundtruth is connected by a line. Best viewed in color.



(a) *fr2* (b) *fr3*

Fig. 7. The average number of iterations required to achieve the final solution when the maximum number of iteration is 500. Using the same matching type, PROSAC or B-PROSAC consistently found the best solution in earlier iterations.

C. Convergence of the algorithms

Fig. 7 shows the average number of iterations required to find the best solution in each method. *hybrid_b-prosac* on average fell into the best solution much earlier than the baselines. B-PROSAC allows slightly faster convergence thanks to the sampling strategy considering the similarity score between an observation and a text label on the landmark. Although the absolute runtime of *hybrid* matching is worse than *class* and *clip* with the same number of iteration, the faster convergence of the proposed method suggests a possibility of further optimization in runtime with appropriate early-stopping criteria.

D. Discussions

While we confirmed the effectiveness of the proposed method, there are some factors to be further examined.

Limitation in the accuracy of CLIP The accuracy of CLIP depends on image quality, and in some cases, it completely fails to establish correct correspondences. It is ideal to train a more accurate model. In practice, however, we should consider careful selection of observations for correspondence matching to increase the robustness of the method as errors in estimation of machine learning models is inevitable.

Dealing with larger scale maps The current framework cannot handle cases where there more than k objects that are either identical or closely resembling each other. To address this issue, a more flexible approach is required, such as adapting the value of k based on the landmark distribution.

Optimality of parameters In the experiments, we empirically set the parameter $k = 3$ as it yielded the best results overall. However, how to set the optimal parameter is not trivial and not investigated well. An adaptive way of parameter setting is required.

V. CONCLUSION AND FUTURE WORK

We presented a method of landmark association for global localization in an object map. It leverages landmarks with fine-grained information about the landmarks given as natural language labels, and the multi-modal capability of CLIP. By effectively associating the text-labeled landmarks with visual observations, the proposed method improves the performance of global localization. In addition, we introduced a sampling strategy inspired by PROSAC for boosting the efficiency of sampling-based inlier extraction. As a result, we confirmed that the proposed method enables more accurate and more efficient camera global localization compared to the baseline based on object classes and RANSAC.

As future work, we are looking to apply the proposed method to loop closing capability of an object SLAM system.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[4] L. Nicholson, M. Milford, and S. Niko, "QuadricSLAM : Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2019.

[5] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[6] M. Zins, G. Simon, and M.-O. Berger, "OA-SLAM: Leveraging Objects for Camera Relocalization in Visual SLAM." In *Proc. of the IEEE International Symposium on Mixed and Augmented Reality*, IEEE, 2022, pp. 720–728.

[7] Y. Wu, Y. Zhang, D. Zhu, Z. Deng, W. Sun, X. Chen, and J. Zhang, "An Object SLAM Framework for Association, Mapping, and High-Level Tasks," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2912–2932, 2023.

[8] Y. Kawasaki, S. Mochizuki, and M. Takahashi, "ASTRON: Action-Based Spatio-Temporal Robot Navigation," *IEEE Access*, vol. 9, pp. 141 709–141 724, 2021.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] M. Zins, G. Simon, and M.-O. Berger, "Level Set-Based Camera Pose Estimation From Multiple 2D/3D Ellipse-Ellipsoid Correspondences." In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2022, pp. 939–946.

[11] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision." In *Proc. of the International Conference on Machine Learning*, PMLR, 2021, pp. 8748–8763.

[12] O. Chum and J. Matas, "Matching with PROSAC — Progressive Sample Consensus." In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2005, pp. 220–226.

[13] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects." In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, pp. 1352–1359.

[14] S. Yang and S. Scherer, "Monocular Object and Plane SLAM in Structured Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, 2019.

[15] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang, and W. Wang, "SO-SLAM: Semantic Object SLAM With Scale Proportional and Symmetrical Texture Constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4008–4015, 2022.

[16] X. Han and L. Yang, "SQ-SLAM: Monocular Semantic SLAM Based on Superquadric Object Representation," Tech. Rep., 2022. arXiv: 2209.10817. [Online]. Available: http://arxiv.org/abs/2209.10817.

[17] A. Dosovitskiy *et al.*, "An Image Is Worth 16 x 16 Words : Transformers for Image Recognition at Scale." In *Proc. of the International Conference on Learning Representations*, 2021.

[18] T. B. Brown *et al.*, "Language models are few-shot learners." In *Proc. of the International Conference on Neural Information Processing Systems*, vol. 2020-Decem, Curran Associates, Inc., 2020, pp. 1877–1901.

[19] A. Brohan *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," Tech. Rep., 2023, pp. 1–26. arXiv: 2307.15818. [Online]. Available: http://arxiv.org/abs/2307.15818.

[20] D. Shah, B. Osinski, B. Ichter, and S. Levine, "LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action." In *Proc. of the Conference on Robot Learning*, PMLR, 2022.

[21] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation." In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2023.

[22] R. Mirjalili, M. Krawez, and W. Burgard, "FM-Loc: Using Foundation Models for Improved Vision-based Localization," Tech. Rep., 2023. arXiv: 2304.07058. [Online]. Available: http://arxiv.org/abs/2304.07058.

[23] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "MAGSAC++, a Fast, Reliable and Accurate Robust Estimator." In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2020, pp. 1301–1309.

[24] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework." In *Proc. of the International Conference on Machine Learning*, 2022, pp. 23 318–23 340.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems." In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 573–580.

[26] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context." In *Proc. of the European Conference on Computer Vision*, 2014, pp. 740–755.