

# SOS-SLAM: Segmentation for Open-Set SLAM in Unstructured Environments

Jouko Kinnari<sup>1\*</sup>, Annika Thomas<sup>2\*</sup>, Parker Lusk<sup>2</sup>, Kota Kondo<sup>2</sup>, and Jonathan P. How<sup>2</sup>

**Abstract**—We present a novel framework for open-set Simultaneous Localization and Mapping (SLAM) in unstructured environments that uses segmentation to create a map of objects and geometric relationships between objects for localization. Our system consists of 1) a front-end mapping pipeline using a zero-shot segmentation model to extract object masks from images and track them across frames to generate an object-based map and 2) a frame alignment pipeline that uses the geometric consistency of objects to efficiently localize within maps taken in a variety of conditions. This approach is shown to be more robust to changes in lighting and appearance than traditional feature-based SLAM systems or global descriptor methods. This is established by evaluating SOS-SLAM on the Båtvik seasonal dataset which includes drone flights collected over a coastal plot of southern Finland during different seasons and lighting conditions. Across flights during varying environmental conditions, our approach achieves higher recall than benchmark methods with precision of 1.0. SOS-SLAM localizes within a reference map up to 14x faster than other feature-based approaches and has a map size less than 0.4% the size of the most compact other maps. When considering localization performance from varying viewpoints, our approach outperforms all benchmarks from the same viewpoint and most benchmarks from different viewpoints. SOS-SLAM is a promising new approach for SLAM in unstructured environments that is robust to changes in lighting and appearance and is more computationally efficient than other approaches. We release our code and datasets: <https://acl.mit.edu/SOS-SLAM/>.

## I. INTRODUCTION

The capability of a robot to localize itself with respect to an environment is a fundamental requirement in mobile robotics. Various approaches exist for achieving this, including infrastructure-based methods, map-based methods, and Simultaneous Localization and Mapping (SLAM).

Infrastructure-based methods such as Global Navigation Satellite System (GNSS) directly provide estimates of location in a known coordinate system, but are subject to interference by malicious actors [1] and limited in availability (*e.g.*, only work outdoors). SLAM-based approaches [2] do not depend on availability of localization infrastructure, and they correct for odometry measurement errors by detecting loop closures. In the single-agent SLAM setting, knowledge of odometry measurements' uncertainty bounds can be used for constraining the search of matches between current environment observation and the SLAM map. In the multi-agent



Fig. 1: SOS-SLAM tracks object masks produced with no pre-training or fine-tuning across sequential posed camera frames to build sparse object-based maps. It robustly associates object masks using their geometric relationship with each other, enabling correspondence detections between traverses over highly ambiguous natural terrains.

SLAM case, prior knowledge of the agents' poses with respect to each other may not be available, which means that the search of correspondences must be done between the maps of agents in an exhaustive manner to perform frame alignment. In the case of localizing within another agent's previously traversed path after time has passed, an agent must search the entire traverse of previous agent and be robust to lighting or environmental variations that occur over time. Localizing onboard necessitates compact map formulation and computationally efficient methods in searching for correspondences. Object-based maps for SLAM address these problems, but current approaches such as [3] rely on assumptions about the set of objects an agent may encounter which may not be possible in unstructured settings.

We propose that for unstructured environments, an observed geometric structure of object masks produced by a foundational segmentation model, without semantic information or object descriptors, is a robust map formulation for localization and loop closure detection. This approach allows for very compact map representations, thus reducing requirements for storage, correspondence search time and communication. This is an especially suitable approach for drone localization over unstructured terrain, as we demonstrate through experiments with localization and loop closure detection in drone flights collected across different seasons.

\* Equal Contribution

<sup>1</sup>J. Kinnari is with Saab Finland Oy, Salomonkatu 17B, 00100 Helsinki, Finland [jouko.kinnari@saabgroup.com](mailto:jouko.kinnari@saabgroup.com)

<sup>2</sup>A. Thomas, K. Kondo and J. How are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. [{annikat, plusk, kkondo, jhow}@mit.edu](mailto:{annikat, plusk, kkondo, jhow}@mit.edu).

We present SOS-SLAM, an open-set mapping pipeline that makes no prior assumptions about the content of the environment to extract and map objects from visually ambiguous unstructured settings. Utilizing the Segment Anything Model (SAM) [4] for front-end segment detection, our pipeline tracks object masks across frames and prunes spurious detections to construct a map of consistent objects. We use a robust graph-theoretic data association method [5] to associate object locations within object maps, leveraging their size to prune unreasonable associations. Since many SLAM algorithms are expensive to run on platforms with limited computing resources, we formulate a windowed correspondence search that can trade off accuracy for computational cost.

The association search problem with no prior information is challenging, since in general, a very large number of associations are possible. Given a very high number of putative associations, even modern approaches approximating robust data association backends [5] have a runtime that scales quadratically with the number of putative associations, restricting feasibility of considering all possible matches to only small scenarios. In our approach, we relax these requirements in two ways: 1) our mapping pipeline provides an efficient (*i.e.*, small but discriminative) set of objects in comparison to feature-based approaches, enabling efficient storage, communication and correspondence search, and 2) we note that objects observed spatially close to each other in the frame of one robot should be spatially close to each other in the frame of the second robot for objects that correspond with each other. This approach enables compact map representations for efficient storage and faster runtime by leveraging geometric consistency in the correspondence search.

In summary, the contributions of this work include:

- A front end capable of reconstructing vehicle maps made of segmented object masks that are less than 0.4% the size of other benchmark maps, relying on no prior assumptions of the operating environment.
- A method for relating vehicle maps using a geometric correspondence search with a windowed approach that is up to 14x faster than feature-based data association approaches.
- SOS-SLAM achieves higher recall compared to classical feature-based methods and a state-of-the-art visual place recognition approach evaluated on real-world flights across varied seasonal and illumination conditions.
- We release the Båtvik seasonal dataset containing long traverses with an Unmanned Aerial Vehicle (UAV) across diverse lighting conditions and seasonal appearance change to promote novel contributions towards localization in unstructured environments.

## II. RELATED WORK

One key design choice for a SLAM system is how to relate the current *environment measurements*, *i.e.*, sensor inputs in the vicinity of the robot, to previous environment measurements. We review related work in this domain with focus

on applications for deep learning in visual navigation and approaches to localization in unstructured environments.

### A. Environment Representation

Descriptive and efficient environment representation lays the framework for robust localization. In visual SLAM, common environment representations are feature-based or object-based. In feature-based SLAM, the environment is described by consistently detectable features within images that are distinct from one another like ORB [6], SIFT [7], or SURF [8] features. Several SLAM systems utilize these features for mapping, as shown in [9]. While feature-based SLAM is widely used, this approach poses a significant data handling challenge due to the substantial data volume it entails. In object-based SLAM methods, object detectors like YOLOv3 [10] can be used to extract a set vocabulary of objects in urban environments. Using object-based methods coupled with semantic labels can be advantageous in gathering contextual information about the environment while maintaining a compact map, as demonstrated in earlier work [3]. While the sparse nature of object-based mapping is promising for computational efficiency in large-scale operations, object classifiers are restricted to the types of objects that can be detected, limiting their use in unstructured environments.

### B. Segmentation

Segmentation partitions an image into meaningful regions or objects. In computer vision, segmentation is widely used for object detection and classification with classifiers such as YOLOv3 [10] or semantic classifications with CLIP [11]. Segmentation can be used as a step in methods for path planning and object detection [12]. Segment Anything [4], uses a trained model to perform segmentation in any environment without assumptions or fine tuning and has recently been used for coordinate frame alignment in multi-agent trajectory deconfliction [13].

### C. Deep Learning in Visual Navigation

Some recent developments in deep learning for visual navigation leverage foundational models [14], which are models trained in a self-supervised way and can accomplish many tasks without fine-tuning or additional training. Deep learning has been used to learn features in methods such as SuperPoint [15] and D2-Net [16], but these techniques require environment-specific training for optimal performance in different environments, limiting their use in open-set scenarios. One of the largest advances in visual navigation using deep learning has been in the field of visual place recognition where global descriptors facilitate robust scene recognition across viewpoints and differing illuminations. AnyLoc [17] is a technique for visual place recognition that uses DINoV2 [18], a self-supervised vision transformer model in combination with unsupervised feature aggregation using VLAD [19], which surpasses other VPR approaches in open-set environments, but degrades in cases where contents are similar across frames. Segment Anything uses deep learning to segment environments without prior assumptions [4].

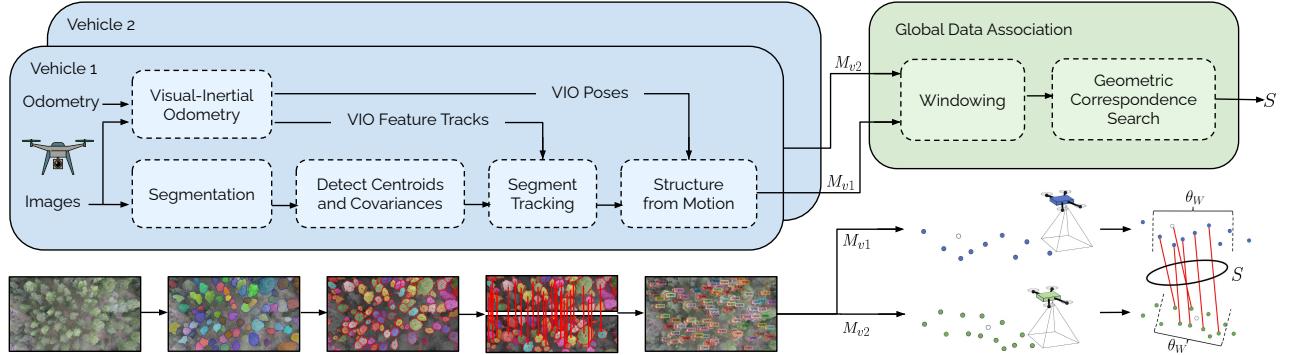


Fig. 2: SOS-SLAM incorporates two novel components. The front end mapping pipeline utilizes the vehicle odometry sensor along with camera images to perform SLAM and generate vehicle maps. The frame alignment pipeline offsets windows and uses our data association algorithm to filter the most likely correspondences.

#### D. Unstructured Environments

Unstructured environments are a challenging setting for many visual SLAM systems that assume urban-centric information such as the presence of lane markings or buildings. In unstructured environments, it is important to adequately extract meaningful features or recognize objects to aid in place recognition. Some approaches address road roughness and limited distinctive features in these settings by integrating a range of sensor modalities and strategies like using wheel odometry with visual tracking [20], integrating topological maps [21], and utilizing lidar point clouds [22]. These methods exhibit limitations tied to external hardware requirements, point cloud size constraints, susceptibility to structural changes, and reliance on the assumption of well-defined off-road trails. A recent work globally localizes agents in unstructured environments but is limited to the presence of specific object classes [3].

#### E. Placement of This Work

SOS-SLAM utilizes a pre-trained foundation model for segmentation to construct object-based maps without any prior assumptions about the environment such as presence of objects of specific classes. The generality of the structure of this framework allows it to localize successfully in unstructured environments with extreme illumination and structural changes while keeping map sizes compact enough to be shared between multiple agents.

### III. METHOD

Our method consists of two main parts; mapping and frame alignment. A block diagram of our method is illustrated in Figure 2 where two vehicles generate object-based maps then perform global data association.

#### A. Mapping

Our mapping approach consists of running camera images through a foundational image segmentation model [4], identifying tracks (*i.e.*, finding correspondence between object masks across a sequence of images), and reconstructing the positions of the centroids of the segmented areas with a

Structure-from-Motion (SfM)-style approach using camera poses estimated by visual-inertial odometry (VIO).

To enable balancing computational requirements, we perform segment detection only after movement of  $\theta_T$  meters after most recent keyframe, estimated with VIO.

Given an image  $\mathcal{I}(t)$ , acquired at time  $t$ , we use SAM [4] to extract binary object masks  $I_k(t)$ , indexed by  $k \in \{0, 1, \dots, K(t)\}$ . For each  $I_k(t)$  larger than 2 pixels, we compute the centroid  $m_k(t)$  and covariance  $\Sigma_k(t)$  of the pixel coordinates of each mask. Further, we extract SIFT features [23] from the region of the mask and store them as set  $A_k(t)$ . We emphasize that we only use SIFT features for inter-frame tracking and don't keep a record of them in the map. We also compute a size descriptor  $h_k(t) = \sqrt{\lambda_k(t)}$ , where  $\lambda_k(t)$  is the largest eigenvalue of  $\Sigma_k(t)$ .

For inter-frame tracking, we assume a generic VIO implementation is available for tracking camera poses  $T(t) \in SE(3)$  as well as for tracking movement of visual feature points between images  $\mathcal{I}(t_i)$  and  $\mathcal{I}(t_j)$ . We compute the amount of movement of visual feature points tracked by VIO in pixels and compute mean  $\mu_p$  and standard deviation  $\sigma_p$  of movement of feature points.

We evaluate putative correspondences for each track whose latest observation is at most  $\theta_t$  keyframes old; this provides some robustness against intermittent temporal inconsistencies in detection of segments by SAM.

To associate detections of object masks across time, we require that 1) the putative match satisfies the epipolar constraint [24] with a margin of  $a$  pixels and 2) the change in pixel coordinates of the centroid across frames is less than a specified limit when comparing to movement of VIO feature points, *i.e.*,

$$\frac{|m_i(t-1) - m_j(t)| - \mu_p}{\sigma_p} < \theta_v, \quad (1)$$

when considering the association  $(i, j)$ , and that 3) the putative association satisfies a similarity constraint  $q > \theta_q$ , where scoring function  $q = \sqrt{q_s q_f}$  is the geometric mean of a size scoring function  $q_s$  and feature scoring function  $q_f$ .

For comparing size descriptors, we use a handcrafted scoring function for scoring a putative association of areas with size descriptors  $h_i$  and  $h_j$ :

$$q_s(h_i, h_j) = \begin{cases} 1 + \cos(\frac{\pi}{\theta_h} r(h_i, h_j)) & \text{if } r(h_i, h_j) < \theta_h, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where we measure relative size difference of masks  $i$  and  $j$  using

$$r(h_i, h_j) = \frac{2|h_i - h_j|}{h_i + h_j}. \quad (3)$$

We define the feature scoring function  $q_f$  as the fraction of SIFT features in sets  $A_i(t-1)$  and  $A_j(t)$  that are not eliminated by Lowe's ratio test [23].

The three requirements listed above may provide more than one possible associations between object masks observed in latest keyframe to tracks. To provide an unambiguous mapping between latest object masks and history of masks, we use an implementation of the Hungarian algorithm [25] using weights from (2). We initialize new tracks for observations that cannot be matched.

Finally, for tracks with more than  $\theta_n$  observations, indexed by  $n \in \{0, 1, \dots, N\}$ , we build a small SFM-style factor graph [26] for each track separately. We specify poses based on odometry and projection factors for the mean positions of object masks, and use GTSAM [27] for finding a minimal cost solution to the factor graph. We record the mean positions of each object, indexed by  $n$  in frame of robot  $i$ , in odometry frame,  $l_{i,n}$ . We discard tracks that do not converge to a solution as they are often a result of tracking errors. We further compute a size descriptor for each object, which scales the size of the object with estimated depth, to improve invariance to view distance.

The end result of the mapping pipeline is a vehicle map  $\mathcal{M}_{v,i}$ , which contains estimated positions of objects corresponding to segmented masks, expressed in the odometry frame of robot  $i$ , and size descriptors for the object masks.

### B. Finding correspondences between vehicle maps

With perfect knowledge of correspondences of objects, any alignment errors could be mitigated to the level determined by measurement errors of environment measurements. For robot  $i$  that has observed  $n$  successfully tracked object masks in  $\mathcal{M}_{v,i}$ , we thus focus on finding the correspondences between objects within its own map  $\mathcal{M}_{v,i}$  and another map, communicated by a peer or collected at an earlier time instant,  $\mathcal{M}_{v,j}$ .

Assuming no further prior information on the correspondences, the number of possible associations grows quadratically as the number of objects increases, leading to an infeasible search time for any reasonably large map. To utilize the notion that objects spatially close to each other in  $\mathcal{M}_{v,i}$  should be spatially close to each other in  $\mathcal{M}_{v,j}$ , we implement a windowed search approach, where we define a window length of  $\theta_{WL}$  objects, and search for correspondences between the frames, moving forward the window by a stride length of  $\theta_{SL}$  objects after each comparison.

We denote  $\mathcal{M}_{v,i} = \{l_{i,n}\}$  for robot  $i$ , where  $n \in \{0, 1, \dots, N_i\}$ . The windowed search thus attempts to find correspondences between subsets  $\mathcal{M}_{v,i}[a_i \cdot \theta_{SL}, \dots, a_i \cdot \theta_{SL} + \theta_{WL}]$  and  $\mathcal{M}_{v,j}[a_j \cdot \theta_{SL}, \dots, a_j \cdot \theta_{SL} + \theta_{WL}]$  across all values of  $a_i$  and  $a_j$ , where the  $G[\cdot]$  notation corresponds to taking a subset of  $G$  using items with indices  $[\cdot]$ .

In finding correspondences, we first exclude hypothetical pairs of objects where pairwise difference in distance of the objects in each map is  $\varepsilon > \theta_\varepsilon$  or where size difference of objects is significant, i.e.,  $r(h_i, h_j) > \theta_r$ . We weigh putative associations using (2). We use a robust geometric data association framework [5] to approximate a set  $S$  of associations (object pairs). As a final step, we estimate with [28] the relative translation and rotation between  $\mathcal{M}_{v,i}$  and  $\mathcal{M}_{v,j}$ , assuming correspondences defined by  $S$  and discard hypotheses that would result in more than  $\theta_\alpha$  angular difference in roll or pitch. This is motivated by that odometry frames' roll and tilt can be estimated with an IMU due to excitation from gravity. We use the number of associations returned by the framework,  $|S|$  as criteria for accepting or rejecting the hypothesis. We accept the hypothesis if  $|S| > \theta_S$ . By varying threshold  $\theta_S$  for acceptance, we can balance precision and recall of our solution.

## IV. EXPERIMENTS

We evaluate the performance of SOS-SLAM at varying levels of appearance change of the environment, comparing precision, recall, average F1 score, search time and map size with respect to three reference methods. In each experiment, we use a dataset collected for this task. We perform ablation studies to evaluate various design choices of our algorithm.

### A. Datasets

We introduce the Båtvik seasonal dataset which includes six drone flights that travel a distance of approximately 3.5

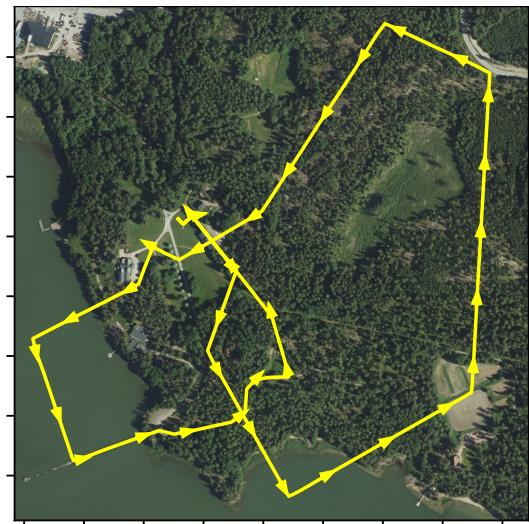


Fig. 3: UAV trajectory passes mostly natural forests and sea areas. Figure shows an area of 860 m by 860 m. Orthophoto © National Land Survey of Finland

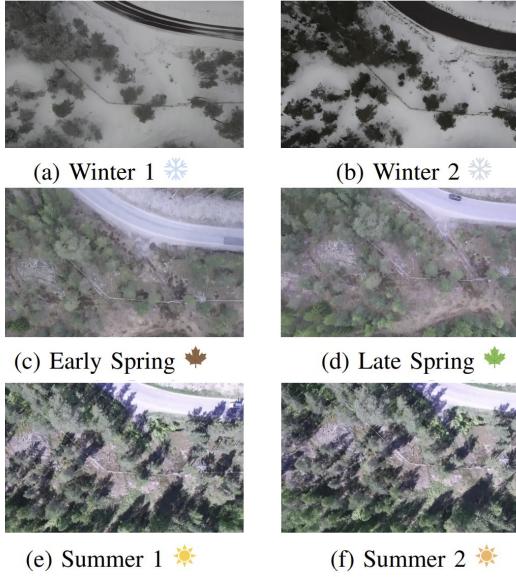


Fig. 4: Example images from Båtvik seasonal dataset, including variation in snow coverage, deciduous tree foliage and sharpness of shadows across different seasons.

km over a coastal plot in southern Finland at an altitude approximately 100 m over ground, each following the same trajectory plan as shown in Figure 3. We release this dataset for public use. The flights consist of drone images collected with a nadir-pointing camera as well as Inertial Measurement Unit (IMU) measurements, and we record autopilot output along with other telemetry data from an Ardupilot-based [29] drone flight controller. The flight takes place over an area that contains only a few buildings, and a large part of the trajectory takes place over a forest region, as well as above sea. This dataset represents flight of an UAV over a terrain that has naturally high ambiguity. We record this flight six times over many seasonal conditions, as illustrated in Figure 4 and outlined in Table I.

### B. Baseline approaches

Several visual SLAM approaches [9] use image features such as ORB or SIFT as front end, for detecting and describing feature points, and use random sample consensus (RANSAC) [30] to prune outliers. We implement two baseline methods that detect and describe image features with each approach using keyframes taken every 2 m of travel of the flight sequence. Next, for evaluating correspondence, we use RANSAC to find correspondences that are consistent in terms of the fundamental matrix, and set a limit for the number of required associations to consider the keyframes a match. We extract 500 SIFT or ORB features, select 20% of best matches in terms of descriptors, and use a reprojection threshold of 5.0 pixels in fundamental matrix filtering. We run RANSAC for a maximum of 2000 iterations with confidence level 0.995.

In addition to dense feature-based approaches, we compare our method against a modern Visual Place Recogni-

TABLE I: Description of flight trajectories in Båtvik dataset.

Name	Icon	Time of flight	Description of appearance
Winter 1	❄️	2022-03-30 12:51	Snow coverage
Winter 2	❄️	2022-03-31 11:39	Snow coverage
Early Spring	🍁	2022-05-05 14:10	Some leaves
Late Spring	🌿	2022-05-25 12:33	Leaves in deciduous plants
Summer 1	☀️	2022-06-09 12:05	Full leaves, hard shadows
Summer 2	☀️	2022-06-09 12:28	Full leaves, hard shadows

tion (VPR) approach that uses global descriptors for images, AnyLoc [17]. AnyLoc outperforms universal place recognition pipelines NetVLAD [31], CosPlace [32] and MixVPR [33] in almost every evaluation, making it an appropriate benchmark that authors claim works across very different objects and lighting conditions. AnyLoc uses DINOv2 features aggregated by VLAD to generate global descriptors for images. In our implementation, we define a DINOv2 extractor following AnyLoc’s parameters at layer 31 with facet value and 32 clusters. We train a VLAD vocabulary of 32 cluster centers on database images, generate global descriptors for each image in the query set, then compute cosine similarity of the global descriptors of each image pair in each sequence.

### C. Performance measures

We compare correspondence search results by first computing what region of the ground would be visible from each keyframe camera pose if the ground under the image acquisition position was flat. For this, we use ground truth camera poses recorded from the extended Kalman filter (EKF) output from an flight controller and a terrain elevation map of the area. By comparing the area of overlap to the area of intersection of each keyframe pairwise, we compute the intersection over union (IoU) of every pair of keyframes. In evaluating recall, we assume that each keyframe pair for which IoU is more than 0.333, the matching algorithm should return a match indication. In evaluating precision, we assume that an algorithm may provide a correspondence between frames if the IoU from ground truth is more than 0.01; for smaller IoUs, we assume a returned correspondence is a false positive. In mapping, for purposes of evaluation, we use relative poses between ground truth poses of flight controller EKF in SFM. We use 3.0 as pixel measurement noise standard deviation. For other parameters, we choose  $\theta_T = 2.0$  m,  $\theta_v = 4.0$ ,  $\theta_q = 0.2$ ,  $\theta_h = 0.2$ ,  $\theta_n = 5$ ,  $\theta_\alpha = 22.5^\circ$ ,  $\theta_\varepsilon = 2.0$ , and  $\theta_r = 0.2$ .

We produce precision-recall results by varying the acceptance limit (threshold for number of detected correspondences for our approach, ORB and SIFT-based methods and required level of cosine similarity for AnyLoc).

The computational time evaluations are made with an 2x8 core Intel Xeon 6134 @ 3.2 GHz cluster computer from which we reserve 16 GB RAM, and in segmentation we use a Tesla V100 GPU.

TABLE II: Localization performance across flights with increasing visual discrepancy between flights.

Case	Implementation	Precision	Recall	Mean search time [std] (s)	Map size (Mb)
$\Delta_T = 23 \text{ min}$	Ours	<b>1.00</b>	<b>0.60</b>	5.34 [0.13]	<b>0.05</b>
	ORB+RANSAC	<b>1.00</b>	0.53	49.11 [0.25]	25.43
	SIFT+RANSAC	<b>0.99</b>	0.55	<b>76.82 [0.52]</b>	<b>406.95</b>
	Any-Loc	0.92	<b>0.01</b>	<b>0.11 [0.01]</b>	310.05
$\Delta_T = 1 \text{ day}$	Ours	<b>1.00</b>	<b>0.37</b>	4.35 [0.04]	<b>0.07</b>
	ORB+RANSAC	<b>0.99</b>	0.13	27.23 [0.48]	21.94
	SIFT+RANSAC	<b>1.00</b>	0.08	<b>40.28 [0.67]</b>	<b>345.89</b>
	Any-Loc	0.55	<b>0.00</b>	<b>0.12 [0.01]</b>	310.05
$\Delta_T = 20 \text{ days}$	Ours	<b>0.99</b>	<b>0.28</b>	9.29 [0.12]	<b>0.10</b>
	ORB+RANSAC	<b>1.00</b>	0.19	38.35 [0.52]	22.07
	SIFT+RANSAC	<b>1.00</b>	0.27	<b>49.75 [0.67]</b>	<b>337.47</b>
	Any-Loc	0.97	<b>0.00</b>	<b>0.12 [0.01]</b>	310.05
$\Delta_T = 15 \text{ days}$	Ours	<b>1.00</b>	<b>0.17</b>	8.06 [0.14]	<b>0.05</b>
	ORB+RANSAC	<b>1.00</b>	0.01	40.38 [0.44]	25.43
	SIFT+RANSAC	<b>1.00</b>	0.02	<b>64.75 [0.81]</b>	<b>407.20</b>
	Any-Loc	<b>1.00</b>	<b>0.00</b>	<b>0.12 [0.01]</b>	310.05

#### D. Precision and recall: Full Traverse

First, we evaluate the performance of SOS-SLAM when an agent localizes within a previously collected map from another agent after time has passed. We include sections of the flight in Figure 3 that do not involve flying over water, as visual navigation-based systems do not perform well in environments with no distinctive features. Thus, to evaluate the performance of our pipeline over unstructured terrain with a variety of visual features, we consider the flight as a whole in addition to flights from the same viewpoint and different viewpoints. Differentiating into these test cases allows us to evaluate the performance of our method when localizing from the same viewpoint and different viewpoints.

In Table II, we show precision and recall of each comparison method in the cross-seasonal localization case increasing visual discrepancy between flights from the top to bottom of the table. We tabulate the recall returned by each method for the case where precision is 0.99 or above; if the method is not capable of this, we report the recall at highest returned precision. We tune this way to reflect the importance of accurate loop closures in the SLAM setting.

Our method provides better localization performance than the reference methods when tuned for 0.99 precision or above. The case of Summer 1 versus Summer 2 shows that we outperform reference methods even with very minor appearance change between two flights. In the Winter 1 versus Winter 2 case, snow accumulation causes the other methods to degrade considerably while our method can still recall many objects between the two maps. The Early Spring versus Late Spring and Late Spring versus Summer cases represent large visual ambiguities between the two maps, and while our method outperforms baseline methods, shadows and changes in appearance of objects degrade performance.

The smaller map sizes in our approach allow for communication of maps between multiple agents without the need to transmit large amounts of information. The shorter mean search time of our method versus ORB+RANSAC and SIFT+RANSAC applies to use cases with real-time imple-

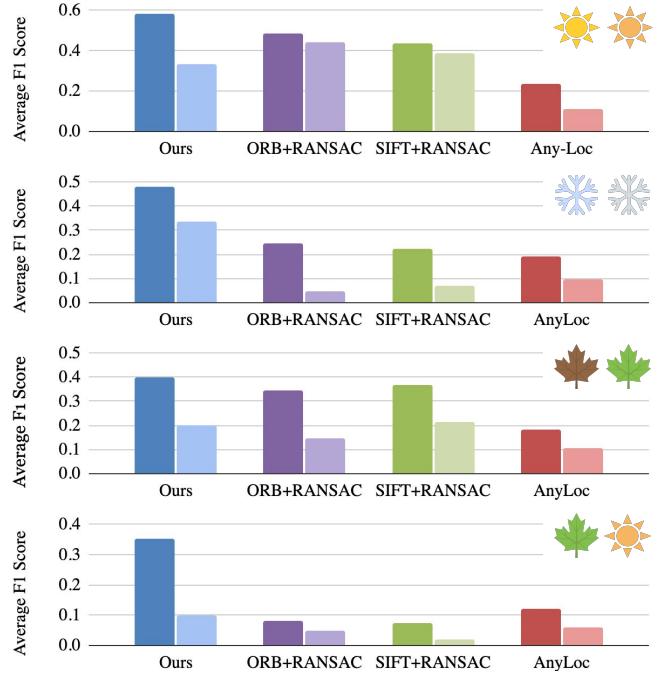


Fig. 5: Average F1 scores of different cross-season cases. Bars indicate the performance from the same viewpoint (left) and from different viewpoints (right).

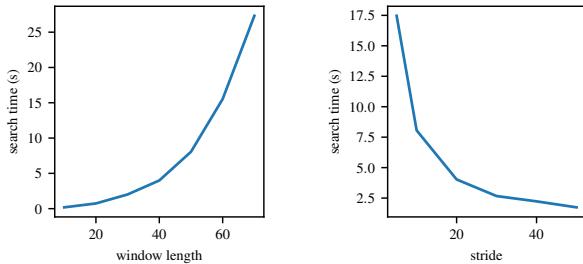
mentation. We also note that our method uses no knowledge of odometry in the data association step, so sufficient prior knowledge of odometry would limit this search time to even shorter as it would not be necessary to observe the full arrangement of objects. While AnyLoc’s search time is shorter than our method, its performance metrics reflect that it is not suited to cases that necessitate high precision.

#### E. F1 Score: Performance Analysis by Viewpoint

To further analyze the performance of our method by viewpoint, we consider the cases in which an agent localizes within the map of another agent from the same viewpoint and when an agent passes over a place previously seen by another agent from a separate viewpoint. We calculate the average F1 score of different flights separated into the same viewpoint and different viewpoint, as shown in Figure 5.

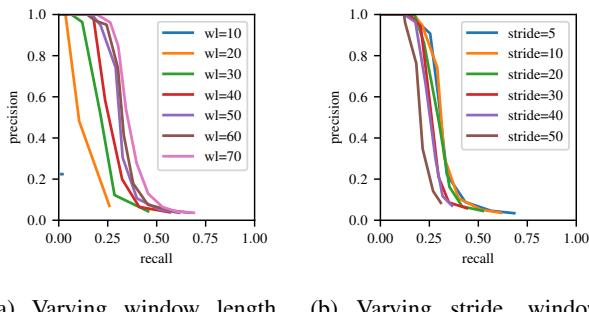
In this evaluation, we take the average F1 score calculated as the average value between when precision is tuned to at least 0.99 and when recall is tuned to at least 0.99. If recall cannot be tuned to at least 0.99, we start at the highest value. Thus, these results demonstrate performance in cases tuning between precision and recall.

In Figure 5, we see that all methods have a lower average F1 score in the different viewpoint setting than in the same viewpoint setting. We note that our method does not perform as well as reference methods in the summer case from multiple viewpoints due to the sparse map representation being limited by field of view, but it maintains performance while others degrade when there are more visual variations between the flights.



(a) Search time as function of window length.  
(b) Search time as function of stride.

Fig. 6: Search time evaluations varying window length and stride.



(a) Varying window length, stride = 10.  
(b) Varying stride, window length = 50.

Fig. 7: Precision and recall at different values of stride and window length. Each graph is generated by varying minimum count of matches in window.

#### F. Ablation study

Due to the computational complexity of the correspondence search problem, we include tunable parameters for the windowed search approach that trade off between performance and runtime. The window length  $\theta_{WL}$  is defined as the number of objects we consider at a time by their ID as assigned during map construction. The stride length  $\theta_{SL}$  is the number of objects we skip as we slide the window along the entire traverse. In Figure 6, the search time throughout the entire traverse is evaluated. We demonstrate that increasing the window length increases runtime and increasing the stride length, and increasing the stride length decreases runtime.

To evaluate the performance of the system as the window length and stride length are tuned, we look at the precision and recall curves in Figure 7. Increasing the window length increases the performance of SOS-SLAM and decreasing the stride increases the performance of the system.

We find that the performance increases up to a window length of 50 and a stride length of 10, reflecting the parameters used in our experiments. These parameters enable tuning as a tradeoff between performance and runtime that can be tailored to applications.

#### V. DISCUSSION

This work demonstrates the value of incorporating foundation models into front-end object detection and map construction in unstructured environments. This formulation provides sufficient geometric cues that are highly suitable for localization and loop closure detection. The ability for us to extract geometric information from new environments across a variety of visual conditions enables robust and accurate map formulation for SLAM.

An important aspect of SOS-SLAM is the significant speed improvement the correspondence search problem offers compared to existing approaches with similar performance. The speed advantage opens the door to real-time applications and makes our pipeline suitable for deployment in resource-constrained robotic systems. This approach can also be extended to various other use cases beyond drones as demonstrated, including rovers. It may also be adaptable to different sensor modalities such as Long-Wavelength Infrared (LWIR).

With our work, we share the Båtvik seasonal dataset which represents a challenging real-world scenario for visual navigation in unstructured environments. The dataset contains very few trackable objects and includes artifacts that occur in drones with hardware constraints such as image compression in a video encoder. Our evaluation reveals that baseline methods are affected by even short time gaps between traverses, highlighting the need for robust visual approaches in these environments. The release of this dataset introduces a new baseline in SLAM, enabling evaluation of robustness to changing seasons and visual conditions.

Our evaluation does not fully account for uncertainty, and we plan to address cases with less favorable triangulation geometry that may impact depth accuracy and scenarios with significant drift in future work. Properly modeling and utilizing uncertainty will help provide a more formal approach to accurate loop closure detection. We also plan to incorporate additional information about the objects, including semantic information in environments containing variable discernible objects, incorporating anisotropic object covariance, and developing robust descriptors for geometry to enable faster correspondence search over large hypothesis spaces.

#### VI. CONCLUSION

We present SOS-SLAM, a framework for compact mapping and fast localization that is able to operate in any open-set environment containing segmentable objects. The compact nature of the maps lends the pipeline well to a multi-agent scenario, allowing for efficient communication streams between agents. Experiments with the Båtvik outdoor dataset demonstrate the pipeline's ability to align frames in a challenging unstructured environment across different seasonal conditions and to detect loop closures within agents' own traverses. In both cases, the size of the reference maps generated by each agent is efficient and concise enough to be communicated between agents with limited communication bandwidth.

## ACKNOWLEDGEMENT

This work was supported by Saab Finland Oy, Boeing Research & Technology, and the National Science Foundation Graduate Research Fellowship under Grant No. 2141064. The dataset was collected as part of Business Finland project Multico (6575/31/2019). We acknowledge the computational resources provided by the Aalto Science-IT project.

## REFERENCES

- [1] F. Dovis, *GNSS Interference, Threats, and Countermeasures*. Artech, 2015.
- [2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [3] J. Ankenbauer, P. C. Lusk, and J. P. How, “Global localization in unstructured environments using semantic object maps built from various viewpoints,” *arXiv preprint arXiv:2303.04658*, 2023.
- [4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” *arXiv preprint arXiv:2304.02643*, 2023.
- [5] P. C. Lusk, K. Fathian, and J. P. How, “CLIPPER: A graph-theoretic framework for robust data association,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 828–13 834.
- [6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [8] H. Bay, T.uytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 404–417.
- [9] I. Abaspuor Kazerouni, L. Fitzgerald, G. Dooley, and D. Toal, “A survey of state-of-the-art on visual SLAM,” *Expert Systems with Applications*, vol. 205, p. 117734, 2022.
- [10] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
- [11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [12] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, “On the segmentation of 3D LIDAR point clouds,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2798–2805.
- [13] K. Kondo, C. T. Tewari, M. B. Peterson, A. Thomas, J. Kin-nari, A. Tagliabue, and J. P. How, “PUMA: Fully decentralized uncertainty-aware multiagent trajectory planner with real-time image segmentation-based frame alignment,” *arXiv preprint arXiv:2311.03655*, 2023.
- [14] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2021.
- [15] D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperPoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.
- [16] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-net: A trainable CNN for joint description and detection of local features,” in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.
- [17] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, “AnyLoc: Towards universal visual place recognition,” *arXiv preprint arXiv:2308.00688*, 2023.
- [18] M. Oquab, T. Darcret, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, “DINOv2: Learning robust visual features without supervision,” *arXiv preprint arXiv:2304.07193*, 2023.
- [19] R. Arandjelovic and A. Zisserman, “All about VLAD,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.
- [20] M. Grimes and Y. LeCun, “Efficient off-road localization using visually corrected odometry,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2649–2654.
- [21] T. Ort, L. Paull, and D. Rus, “Autonomous vehicle navigation in rural environments without detailed prior maps,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2040–2047.
- [22] R. Ren, H. Fu, H. Xue, X. Li, X. Hu, and M. Wu, “Lidar-based robust localization for field autonomous vehicles in off-road environments,” *Journal of Field Robotics*, vol. 38, no. 8, pp. 1059–1077, 2021.
- [23] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [24] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [25] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [26] F. Dellaert, M. Kaess *et al.*, “Factor graphs for robot perception,” *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.
- [27] F. Dellaert, “Factor graphs and GTSAM: A hands-on introduction,” *Georgia Institute of Technology, Tech. Rep*, vol. 2, p. 4, 2012.
- [28] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-D point sets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.
- [29] ArduPilot Community. (2022) Ardupilot - open source autopilot. Accessed: Dec 21, 2023. [Online]. Available: <https://ardupilot.org>
- [30] 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.
- [31] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.
- [32] G. Berton, C. Masone, and B. Caputo, “Rethinking visual geolocation for large-scale applications,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4878–4888.
- [33] A. Ali-Bey, B. Chaib-Draa, and P. Giguere, “MixVPR: Feature mixing for visual place recognition,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 2998–3007.