2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

SOS-Match: Segmentation for Open-Set Robust Correspondence Search and Robot Localization in Unstructured Environments

Annika Thomas[1*], Jouko Kinnari[2*], Parker C. Lusk[1], Kota Kondo[1], and Jonathan P. How[1]

***Abstract*— We present SOS-Match, a novel framework for detecting and matching objects in unstructured environments. 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 and 2) a frame alignment pipeline that uses the geometric consistency of object relationships to efficiently localize across a variety of conditions. We evaluate SOS-Match on the Båtvik seasonal dataset which includes drone flights collected over a coastal plot of southern Finland during different seasons and lighting conditions. Results show that our approach is more robust to changes in lighting and appearance than classical image feature-based approaches or global descriptor methods, and it provides more viewpoint invariance than learning-based feature detection and description approaches. SOS-Match localizes within a reference map up to 46x faster than other feature-based approaches and has a map size less than 0.5% the size of the most compact other maps. SOS-Match is a promising new approach for landmark detection and correspondence search in unstructured environments that is robust to changes in lighting and appearance and is more computationally efficient than other approaches, suggesting that the geometric arrangement of segments is a valuable localization cue in unstructured environments. We release our datasets at `https://acl.mit.edu/SOS-Match/`.**

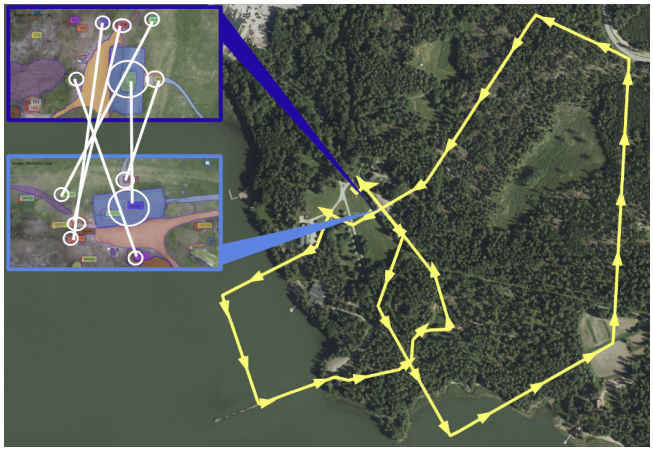


Fig. 1: SOS-Match tracks object masks produced with no pre-training or fine-tuning across sequential posed camera frames to build sparse object-based maps. It robustly associates object masks using their geometric relationship with each other, enabling correspondence detections between traverses over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

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). Map-based methods such as [2] allow localization but only in cases where a global map can be acquired of the environment prior to operation. SLAM-based approaches do not depend on the availability of localization infrastructure or a pre-acquired map of the operating environment, and are able to provide a notion of pose with respect to a robot's initial starting position and orientation [3]. In multi-agent SLAM cases such as [4], there is an additional need to find the alignment between the reference frames of different agents operating in the same environment.

A fundamental question to address in SLAM, as well as map-based localization approaches, is how to relate the current *environment measurements*, *i.e.*, sensor inputs in the vicinity of the robot, efficiently and accurately to a reference map. We propose four requirements of robust correspondence search in the localization problem in unstructured environments. To resolve the ambiguities in the correspondence search of current observations and past observations, or across observations by different robots, the description should (1) provide high precision and recall. The method of description should (2) enable operation in an environment which lacks prominent landmarks, operating in a zero-shot approach *i.e.*, without requiring significant engineering effort if the application domain changes. To allow operation over extended periods, it should (3) be robust to the variation in the appearance of the environment *e.g.*, due to appearance change over seasonal time in the year. The description of the environment should (4) be modest in terms of memory use, computation time and communication bandwidth requirement for multi-agent scenarios.

To meet these requirements, we present SOS-Match, an open-set mapping and correspondence search pipeline that makes no prior assumptions about the content of the environment to extract and map objects from visually ambiguous unstructured settings, and uses only the geometric structure

* Equal Contribution
[1]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.
[2]J. Kinnari is with Saab Finland Oy, Salomonkatu 17B, 00100 Helsinki, Finland jouko.kinnari@saabgroup.com

of the environment as cue for localization. Utilizing the Segment Anything Model (SAM) [5] for front-end segment detection, our pipeline tracks detected segments across frames and prunes spurious detections to construct a map of consistent object masks, without requiring additional training per usage environment. We use a robust graph-theoretic data association method [6] to associate object locations within object maps, leveraging the geometric arrangement of landmarks and their relative position as cues for localization. Since many localization 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. 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 over time.

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 0.5% 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 localizes up to 46x faster than feature-based data association approaches.
- SOS-Match achieves higher recall compared to classical and learned feature-based methods and a state-of-the-art visual place recognition approach evaluated on real-world flights across varied seasonal and illumination conditions, and provides increased robustness to viewpoint variation in comparison to learned feature-based methods.
- 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

Several approaches have been proposed for correspondence search in SLAM and map-based localization. We discuss these approaches by considering various environment representations, segmentation-based approaches, and review deep learning in visual navigation. Additionally, we consider challenges from operation in unstructured environments.

A. Environment Representation

Descriptive and efficient environment representation lays the framework for robust localization. In visual SLAM, common environment representations include feature-based or object-based approaches. In feature-based SLAM, the environment is described by consistently detectable features within a series of images like ORB [7], SIFT [8], SURF [9] or learned features [10], [11]. Several SLAM systems utilize these features for mapping, as shown in [12]. 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 [13] 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 [14].

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 [13] or semantic classifications with CLIP [15]. Segmentation can be used as a step in methods for path planning and object detection [16]. Segment Anything [5], uses a trained model to perform segmentation in any environment without assumptions or fine tuning and a modified version [17] has recently been used for coordinate frame alignment in multi-agent trajectory deconfliction [18].

C. Deep Learning in Visual Navigation

Some recent works in deep learning for visual navigation leverage foundational models [19], which are models trained in a self-supervised way that can accomplish many tasks without fine-tuning or additional training. Deep learning has been used to learn features in methods such as SuperPoint [11] and D2-Net [20]. In visual place recognition, learned global descriptors can facilitate robust scene recognition across viewpoints and differing illuminations. AnyLoc [21] is a technique for visual place recognition that uses DINOv2 [22], a self-supervised vision transformer model in combination with unsupervised feature aggregation using VLAD [23], which surpasses other Visual Place Recognition (VPR) approaches in open-set environments, but degrades in cases where contents are similar across frames.

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. 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 [24], integrating topological maps [25], and utilizing lidar point clouds [26]. 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.

E. Placement of This Work

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

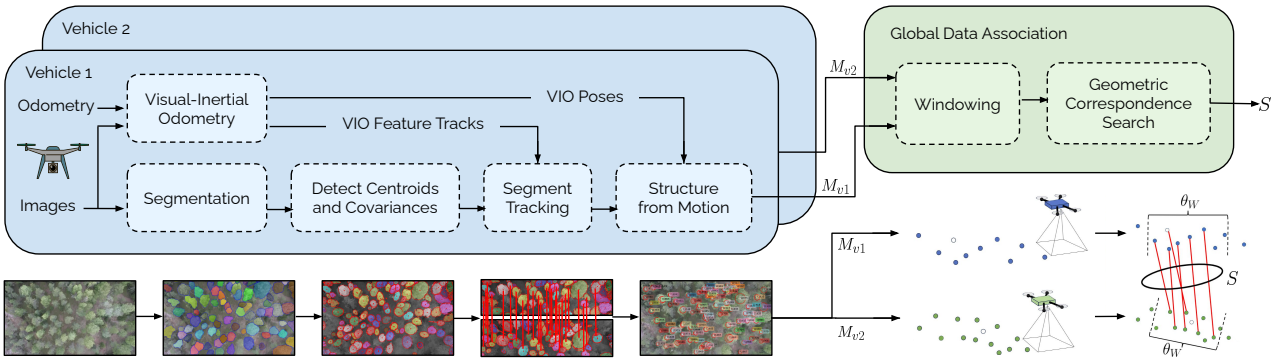


Fig. 2: SOS-Match incorporates two novel components. The front end mapping pipeline utilizes the vehicle odometry sensor along with camera images to perform SLAM and generate vehicle maps. The frame alignment pipeline offsets windows and uses our data association algorithm to filter the most likely correspondences.

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 pre-trained image segmentation model such as [5] or [17], 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).

We perform segment detection only after movement of θ_T meters after most recent keyframe, estimated with VIO. This enables the user of our algorithm the ability to adjust the performance of the system to match the computational resources available on the robot.

Given an image $\mathcal{I}(t)$, acquired at time t, we use the segmentation model 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 [27] from the region of the mask and store them as set $A_k(t)$. We emphasize that SIFT features are used solely for inter-frame tracking, and we 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 the 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 μ_p and standard deviation σ_p of the movement.

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

We associate the detections of segments across consecutive image frames using three techniques. First, we require that an epipolar constraint of the segment centroids is satisfied. Second, we require that the apparent shift in pixel coordinates of the segment centroid is in correspondence with the movement of features points tracked by a VIO algorithm. Third, we match segments that are similar in size and appearance.

For the epipolar constraint (see *e.g.*, [28]), we only allow associations with a margin of less than θ_a pixels. The comparison of the apparent shift to VIO points is based on only allowing movement less than a specified limit θ_v:

$$\frac{\big||m_i(t-1) - m_j(t)| - \mu_p\big|}{\sigma_p} < \theta_v. \tag{1}$$

After excluding infeasible matches based on the epipolar constraint and the requirement of similar movement as VIO detections, there may still exist more than one possible association between segments observed in latest keyframe to tracks. To this end, we compute the similarity of segment association hypotheses based on their appearance and size. For comparing appearance, 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 [27]. For comparing sizes, we use a scoring function $q_s(h_i, h_j)$ to weigh each putative association of areas with size descriptors h_i and h_j:

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

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

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

Finally, we compute a similarity score

$$q = \sqrt{q_s q_f} \tag{4}$$

as the geometric mean of the size scoring function q_s and the feature scoring function q_f. To provide an unambiguous mapping between latest object masks and history of masks, we use an implementation of the Hungarian algorithm [29] using weights from (4). We initialize new tracks for observations that cannot be matched to previous tracks.

Finally, for tracks with more than θ_n observations, we build a small SFM-style factor graph [30] for each track separately. We specify poses based on odometry and projection factors from the centroid pixel coordinates of segments, and use GTSAM [31] for finding a minimal cost solution to the factor graph. We record the mean positions of each segment, 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. Furthermore, to describe the size of the segment in a way that is invariant to the distance at which the segment is observed in each image frame, we compute a size descriptor $h_{S,n}$ scaled approximately to meters, based on the observed pixel size descriptors and distance from the camera to the estimated segment position.

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.

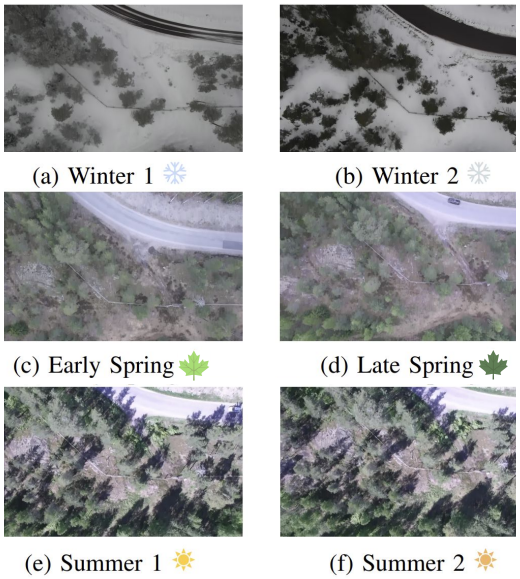
B. Finding correspondences between vehicle maps

With perfect knowledge of the correspondences between 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 θ_{WL} objects, and search for correspondences between the frames, moving forward the window by a stride length of θ_{SL} objects after each comparison.

We denote $\mathcal{M}_{v,i} = \{l_{i,n}\}$ for robot i, where $n \in \{0, 1, \ldots, N_i\}$. The windowed search thus attempts to find correspondences between subsets $\mathcal{M}_{v,i}[a_i \cdot \theta_{SL}, ..., a_i \cdot \theta_{SL} + \theta_{WL}]$ and $\mathcal{M}_{v,j}[a_j \cdot \theta_{SL}, ..., a_j \cdot \theta_{SL} + \theta_{WL}]$ 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.*, $r(h_{S,i}, h_{S,j}) > \theta_r$. We weigh putative associations using (2). We use a robust geometric data association framework [6] to approximate a set S of associations (object pairs). As a final step, we estimate with [32] 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 θ_α 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 θ_S for acceptance, we can balance precision and recall of our solution.



(a) Winter 1 (b) Winter 2 (c) Early Spring (d) Late Spring (e) Summer 1 (f) Summer 2

Fig. 3: Example images from Båtvik seasonal dataset, including variation in snow coverage, deciduous tree foliage and sharpness of shadows across different seasons.

IV. Experiments

We evaluate the performance of SOS-Match at varying levels of appearance change in the environment, comparing precision, recall, average F1 score, search time and map size with respect to five reference methods. In each experiment, we use a dataset collected for this task.

A. Dataset

We introduce the Båtvik seasonal dataset which includes six 3.5 km drone flights following the same trajectory plan as shown in Figure 1 at approximately 100 m above ground, over many seasonal conditions as illustrated in Figure 3 and outlined in Table I. 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 [33] drone flight controller. The flight takes place over an area that contains only a few buildings, and a large part

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

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.

B. Baseline approaches

Several visual SLAM approaches [12] use image features such as ORB or SIFT as front end, for detecting and describing feature points, and use random sample consensus (RANSAC) [34] 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, we use RANSAC to find correspondences that are consistent with respect to 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.

To compare against state-of-the-art learned detector and descriptor methods, we evaluate against LoFTR [35] with pretrained outdoor weights and the SuperPoint detector [11] using SuperGlue with pretrained outdoor weights from [10] for correspondence search. For each method, we retain only keypoint correspondences with confidence of at least 0.7, and add all keypoint match values as a metric for the overall match confidence of each image pair.

In addition to feature-based approaches, we compare our method against a modern VPR approach that uses global descriptors for images, AnyLoc [21]. AnyLoc outperforms universal place recognition pipelines NetVLAD [36], CosPlace [37] and MixVPR [38] in almost every evaluation, making it an appropriate benchmark that authors claim works across very different environmental and lighting conditions. 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 the 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 a 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 ground truth poses of flight controller EKF in SFM. We use 3.0 as pixel measurement noise standard deviation. By varying the runtime as function of window length θ_{WL} and stride θ_{SL}, we experimentally choose parameters $\theta_{WL} = 50$ and $\theta_{SL} = 10$. For other parameters, we choose $\theta_T = 2.0$ m, $\theta_v = 4.0$, $\theta_q = 0.2$, $\theta_h = 0.2$, $\theta_n = 5$, $\theta_\alpha = 22.5°$, $\theta_\varepsilon = 2.0$, 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, the image match confidence threshold for SuperPoint+SuperGlue and LoFTR, and the required level of cosine similarity for AnyLoc.

On an NVIDIA Quadro RTX 3000 with 6 GB VRAM, mean detection and description time is 5.78 s. A recent branch of research on the SAM problem suggests improvements to runtime of the SAM problem (*e.g.*, [17]). We forgo detailed discussion of minimizing the front end runtime for our method to focus on the correspondence search runtime characteristics. Our runtime evaluations in all experiments tabulated in Table II measure time consumed in correspondence search, reflecting time required for localization in real-time settings. The computational time evaluations in Table II are made with an 2x8 core Intel Xeon 6134 @ 3.2 GHz cluster computer from which we reserve 16 GB RAM. For SuperPoint+SuperGlue and LoFTR, which require a GPU for correspondence search, computational time evaluations are made on an NVIDIA RTX 3090.

D. Precision and Recall

First, we evaluate the performance of SOS-Match when an agent localizes within a previously collected map from another agent after time has passed. We include sections of the flight in Figure 1 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.

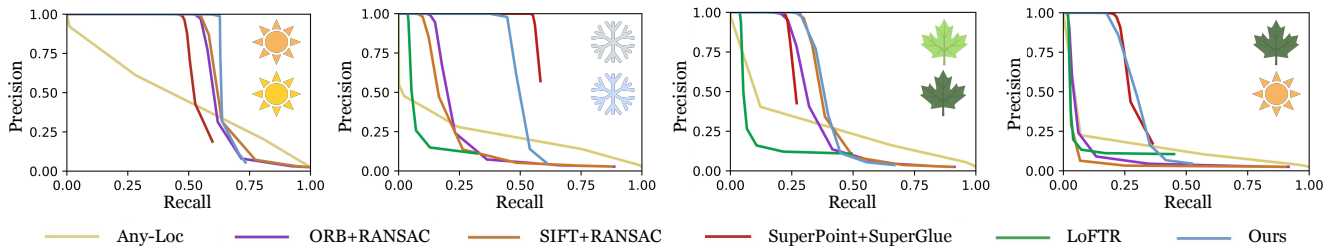


Fig. 4: Precision-recall curves with different approaches with increasing visual discrepancy between flights.

TABLE II: Mean search time and map size across flights with increasing visual discrepancy between flights. Best results are highlighted first and second, and worst is shown in red.

Case	Implementation	Mean search time [std] (s)	Map size (Mb)
Δ_T = 23 min	**Ours**	5.34 [0.13]	0.05
	ORB+RANSAC	49.11 [0.25]	25.43
	SIFT+RANSAC	76.82 [0.52]	406.95
	Any-Loc	0.11 [0.01]	310.05
	SuperPoint+SuperGlue	166.67 [1.17]	2593.72
	LoFTR	133.72 [0.23]	322.9
Δ_T = 1 day	**Ours**	4.35 [0.04]	0.07
	ORB+RANSAC	27.23 [0.48]	21.94
	SIFT+RANSAC	40.28 [0.67]	345.89
	Any-Loc	0.12 [0.01]	310.05
	SuperPoint+SuperGlue	201.17 [5.88]	2843.62
	LoFTR	129.66 [0.31]	454.9
Δ_T = 20 days	**Ours**	9.29 [0.12]	0.10
	ORB+RANSAC	38.35 [0.52]	22.07
	SIFT+RANSAC	49.75 [0.67]	337.47
	Any-Loc	0.12 [0.01]	310.05
	SuperPoint+SuperGlue	166.76 [1.77]	2843.62
	LoFTR	132.47 [0.19]	534.0
Δ_T = 15 days	**Ours**	8.06 [0.14]	0.05
	ORB+RANSAC	40.38 [0.44]	25.43
	SIFT+RANSAC	64.75 [0.81]	407.20
	Any-Loc	0.12 [0.01]	310.05
	SuperPoint+SuperGlue	220.87 [4.52]	2907.95
	LoFTR	154.05 [0.53]	179.6

In Figure 4, we show precision and recall of each comparison method localization case after time has passed, increasing visual discrepancy between flights from left to right cases.

Our method provides better localization performance than the reference methods in the Summer 1 vs. Summer 2, Early Spring vs. Late Spring, and Late Spring vs. Summer 2 cases. Based on Figure 4, in the Winter 1 vs. Winter 2 cases, SuperPoint+SuperGlue appears to outperform our method. While there is a performance benefit in this case for SuperPoint+SuperGlue, in all cases, our method outperforms all reference methods with a significant margin in terms map size, and it outperforms all reference methods aside from AnyLoc in terms of search time, as seen in Table II. A low search time is a particularly critical characteristic for the description of a localization approach where search time is critical for the suitability for online implementation. Furthermore, a small map size is a key enabler of collaborative localization, where robots must share their local maps with limited network bandwidth.

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 that benefit from trading off 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

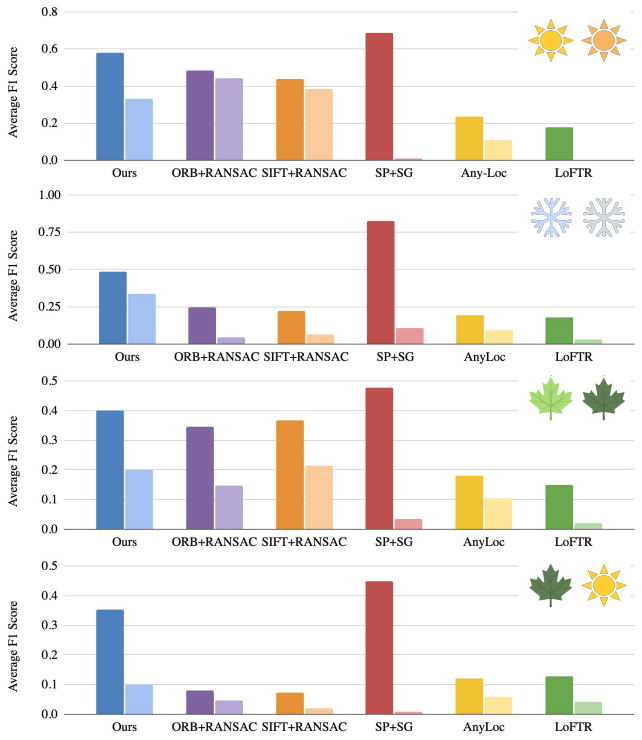


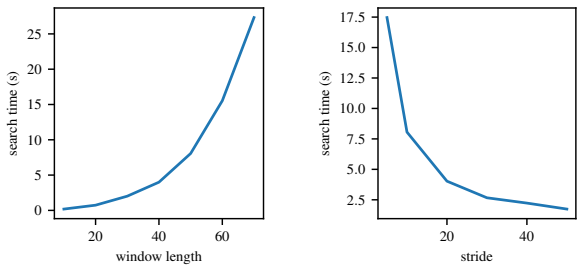
Fig. 5: Average F1 scores of different cross-season cases. Bars indicate the performance from the same viewpoint (left) and from different viewpoints (right).

viewpoint setting. We note that our method does not perform as well as ORB- and SIFT-based methods in the Summer 1 vs. Summer 2 case from multiple viewpoints due to our map representation being sparse and thus limited by field of view, but it maintains performance while others degrade when there are more visual variations between the flights.

In the same viewpoint case, Superpoint+Superglue surpasses our method in average F1 score. However, in the different viewpoint case, the Superpoint+Superglue fares unfavorably against our method and most comparison methods, suggesting that the approach is very sensitive to variation in viewpoint.

F. Ablation study

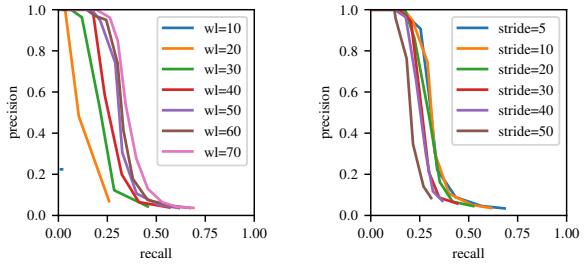
Due to the computational complexity of the correspondence search problem, we include tunable parameters for the windowed search approach that trade off between performance and runtime. Window length θ_{WL} is defined as the number of objects we consider at a time by their ID as assigned during map construction. Stride length θ_{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.



(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 (wl). Each graph is generated by varying minimum count of matches in window.

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-Match 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 trade-off between performance and runtime that can be tailored to applications.

V. Discussion

SOS-Match demonstrates the value of incorporating foundation models into front-end object detection and map construction in unstructured environments. Using segmentation in open-set unstructured settings such as dense forested regions provides sufficient geometric cues that are highly suitable for localization and loop closure detection. Our method also offers a significant speed improvement in search time and size reduction in map size in comparison to reference methods. We consider these major improvements towards satisfying the the requirements of a robust description of environment measurements for use in the localization problem, whose requirements we briefly listed in Sec. I.

We share the Båtvik seasonal dataset, which represents a challenging real-world scenario for visual navigation in unstructured environments, with significant ambiguities in the appearance of the environment. The data includes typical quality issues that occur in drones with hardware constraints such as image compression artifacts, which are useful for real-world evaluation. Our work reveals that most 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 enables evaluation of robustness to changing seasons and visual conditions.

Our method does not fully account for uncertainty, and we plan to address cases with less favorable (non-bird's eye view) triangulation geometry that may impact depth accuracy, as well as scenarios with significant odometry drift in future work.

The comparison of the localization performance against SuperPoint+SuperGlue suggests that it may be possible to find a balance between map size and localization capability by combining the information about the environment's structure with a learning-based approach to description. We thereby plan to incorporate additional information about the objects, including semantic information in environments containing variable discernable objects, incorporating anisotropic object location covariance, and developing robust descriptors for geometry to enable faster correspondence search over large hypothesis spaces. Our proposed method uses size descriptors as a means for excluding putative matches where the size difference of objects is significant, and further search time reduction may be achievable if richer descriptors can be extracted for the segments.

VI. Conclusion

We present SOS-Match, a framework for compact mapping and fast localization that is able to operate in open-set unstructured environments containing segmentable objects. The maps' compactness supports a multi-agent scenario, facilitating efficient communication streams between agents. Experiments with the Båtvik seasonal dataset demonstrate the pipeline's ability to align frames in a challenging unstructured environment with robustness to temporal and viewpoint variations. SOS-Match shows that segmentation provides geometric cues suitable for localization and robust correspondence search in unstructured environments.

Acknowledgement

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

References

[1] R. Morales-Ferre, P. Richter, E. Falletti, A. de la Fuente, and E. S. Lohan, "A survey on coping with intentional interference in satellite navigation for manned and unmanned aircraft," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 249–291, 2020.

[2] J. Kinnari, R. Renzulli, F. Verdoja, and V. Kyrki, "Lsvl: Large-scale season-invariant visual localization for uavs," *Robotics and Autonomous Systems*, vol. 168, p. 104497, 2023.

[3] 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.

[4] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, 2022.

[5] 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.

[6] 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.

[7] 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.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[9] H. Bay, T. Tuytelaars, 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.

[10] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[11] 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.

[12] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual SLAM," *Expert Systems with Applications*, vol. 205, p. 117734, 2022.

[13] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[14] 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.

[15] 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.

[16] 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.

[17] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[18] K. Kondo, C. T. Tewari, M. B. Peterson, A. Thomas, J. Kinnari, 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.

[19] 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.

[20] 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.

[21] 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.

[22] M. Oquab, T. Darcet, 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.

[23] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.

[24] 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.

[25] 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.

[26] 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.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[29] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[30] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, pp. 1–139, 2017.

[31] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, vol. 2, p. 4, 2012.

[32] 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.

[33] ArduPilot Community. (2022) Ardupilot - open source autopilot. Accessed: Dec 21, 2023. [Online]. Available: https://ardupilot.org

[34] 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.

[35] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," *CVPR*, 2021.

[36] 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.

[37] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4878–4888.

[38] 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.