LOSS-SLAM: Lightweight Open-Set Semantic Simultaneous Localization and Mapping

Kurran Singh[1], Tim Magoun[1], and John Leonard[1]

***Abstract*— Enabling robots to understand the world in terms of objects is a critical building block towards higher level autonomy. The success of foundation models in vision has created the ability to segment and identify nearly all objects in the world. However, utilizing such objects to localize the robot and build an open-set semantic map of the world remains an open research question. In this work, a system of identifying, localizing, and encoding objects is tightly coupled with probabilistic graphical models for performing *open-set* semantic simultaneous localization and mapping (SLAM). Results are presented demonstrating that the proposed lightweight object encoding can be used to perform more accurate object-based SLAM than existing open-set methods, closed-set methods, and geometric methods while incurring a lower computational overhead than existing open-set mapping methods.**

Supplementary Material

Code implementation and data is made available here: https://kurransingh.github.io/open-set-slam/

I. Introduction

Developing the ability for robots to understand the world around them in terms of semantically meaningful objects is necessary in order for them to perform higher-level autonomous behaviors ("place the mug in the sink"), interact and cooperate with humans, and have a compressed map representation for low-bandwidth communications [1] or for long-term navigation. As we seek to enable such behaviors in an increasing number of situations, it has become more and more critical for robots to be able to perform *open-set* mapping, where the system can detect and map objects even if they are out of distribution from the training dataset. Recent advances in the machine learning community have made it possible to detect such objects, but incorporating such objects into a simultaneous localization and mapping (SLAM) framework remains an open question. Furthermore, most existing open-set works focus on dense mapping, which are useful for tasks in small scale environments, but do not scale to larger environments or long term mapping as we demonstrate in Section IV-C. This work proposes a computationally efficient method for open-set semantic localization and mapping that utilizes self-supervised vision transformer features (DINO) [2] to augment geometric correspondence matching at the object level. Specifically, the contributions of this work are as follows:

1) A lightweight (sparse) open-set object representation
2) A tightly-coupled open-set semantic SLAM system that uses the proposed object representation along with geometric information to improve the vehicle's positioning accuracy and vice-versa
3) Experimental results on collected and public datasets demonstrating that the proposed method can be used for more accurate and efficient data association and localization compared to dense methods, geometric only methods, and closed-set methods, while also providing more complete maps than closed-set methods.

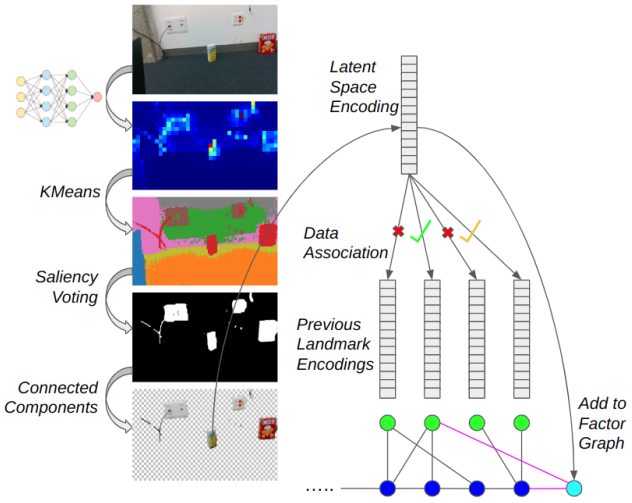


Fig. 1: An overview of the proposed open-set data association system coupled with a factor graph framework when a new image and odometry pair is received. The image is fed into the DINO network to get patch-level encodings, which are then clustered into objects. Those clusters are determined to be either foreground or background based on the attention heads. A connected component analysis yields instance level segmentations, from which for each object, a single encoding vector is used as the object representation. The encoding is compared against the existing landmarks' encodings to determine class matches. The pose of the object is also compared against the existing objects' poses as the final data association filter (not pictured). After building a factor with all matches (green check: best match; orange check: second best match) that pass the filter (depending on back-end method, either expectation-maximization, max-mixtures, or max-likelihood factor), we add a new pose (light blue) to the factor graph with a factor connecting it (light pink) to the previous landmark.

II. Related Work

A. Foundation Models

A promising avenue for self-supervised segmentation comes from attention-based networks, and especially transformers, which were first proposed by Vaswani et al. [3] for use with natural language processing. Caron et al. [2] showed that self-supervised training with self-attention mechanisms improves the performance of transformers to the level of the

[1]K. Singh, T. Magoun and J. Leonard are with the Computer Science and Artificial Intelligence Laboratory (CSAIL) at the Massachusetts Institute of Technology (MIT), 32 Vassar St, Cambridge, MA 02139, USA. Corresponding author: Kurran Singh (singhk@mit.edu)

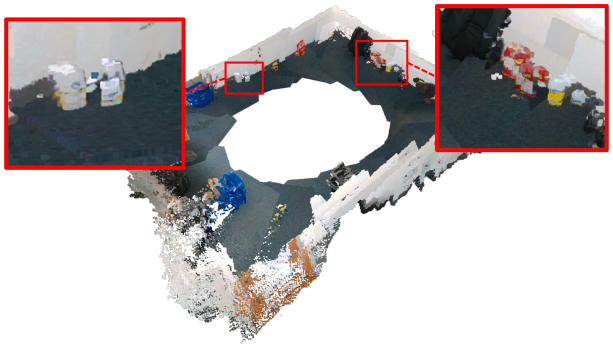


Fig. 2: The results of running an existing open-set mapping method [8] on our data. Artifacts such as the doubled or tripled versions of objects that are a result of not explicitly reasoning about the objects in the scene.

previous state of the art for visual feature extraction in a method that they refer to as DINO. Finally, Amir et al. [4] explore the use of DINO features for semantic segmentation and correspondence matching on common terrestrial objects. Singh et al. [5] extend the previous work by demonstrating the need for finetuning for underwater settings. However, neither of previous two works provides instance-level segmentations, and furthermore, neither explores the connection between the image segmentations, data association, and SLAM, unlike this work.

Radford et al. [6] proposed Contrastive Language-Image Pre-Training (CLIP) to tie image and text together, providing the possibility for open-set mapping with text labels. Our work focuses solely on lightweight open-set mapping, with the assumption that labels can be added with minimal human interaction, the use of CLIP, or something such as Grounding DINO [7] which connects DINO with text labels.

B. Semantic SLAM

Jatavallabhula et al. [8] propose a method seen in Figure 2 for open-set mapping using both visual-language and visual foundation models. Their method performs dense reconstruction using a ∇-SLAM [9] back-end, unlike our work which is more scalable due to a sparse object representation. Their work uses a K-Means clustering scheme in the feature space to extract semantic segmentations similarly to our approach, but due to their dense mapping approach, they do not need any further processing of the segmentation areas, whereas our method refines the segmentations with geometric criteria to improve our centroid-based object representation.

Mazur et al. [10] demonstrate a method of neural field feature fusion that performs open-set mapping in real time using an iMAP [11] back-end. Their method is again a dense method that also cannot be easily extended to include efficient multi-hypothesis data association methods, unlike our method. Grinvald et al. [12] is another dense semantic mapping method that allows for objects not observed in training to be included in the map. Similar to our method, they run both a semantic segmentation network and geometric segmentation method to refine the semantic methods. However, their geometric method assumes that objects tend to have convex surface geometries, unlike ours. They perform data association through the use of an overlapping area method. Furthermore, they assume localization is known, unlike our method which jointly optimizes over both map and sensor poses.

Wu et al. [13] describes a system that includes downstream tasks such as manipulation and active exploration in an object-based mapping framework. While they refer to a centroid in their object parameterization (which also includes a semantic label, scale, and position information), the centroid they use is the geometric centroid rather than the latent space centroid used in our work.

Fu et al. [14] propose a single latent vector embedding for objects that has the advantage of being SE(3)-equivariant, allowing for a object pose estimation and optimization directly in the latent space. However, their method requires a separate set of training data for each object class, making their embedding method infeasible to use for open-set mapping, unlike ours which can operate over any object class without requiring additional training. Doherty et al. [15] present a closed-set semantic SLAM system with a similar Mahalanobis distance-based filtering mechanism for generating hypothesis data associations to ours, albeit with the use of discrete semantic classes rather than a continuous vector space representation. Their work thus utilizes a discrete-continuous optimization framework [16] to optimize the discrete semantic class assignments as well as the continuous object and vehicle poses, whereas our work simply uses the continuous vector representation of the object as proxy for the object class that can be backed out in post-processing. Generally, our work differs from existing works for open-set SLAM in its lightweight, sparse map representation, and the use of that representation for data association that can aid the overall SLAM system.

III. Methods

The proposed method is to generate semantically meaningful embeddings for each patch of input images as in Section III-A, and then run a clustering and connected components procedure to obtain object-level segmentations and latent space encodings as in Algorithm 1. Observations of individual object encodings are associated with existing landmarks or added as new landmarks through a variety of different data association schemes as in Section III-D. A factor graph representation for the joint optimization of landmark positions and vehicle positions is estimated using iSAM2. A visualization of the entire approach can be seen in Figure 1, with detailed explanations of each component in the following sections.

A. Object Embedding

The image embedding network, DINO [2], follows a student-teacher structure in training, with two separate networks that have the same architecture but different parameters, which is used for self-supervised learning by generating a set $\mathcal{V}$ of global views x_1^g and x_2^g and multiple, smaller resolution local views. The student sees all views, whereas the teacher only sees the global views, which thus allows the student network to learn a local-to-global correspondence.

This network has been demonstrated in previous works [2] [4] to produce semantically meaningful features.

B. Clustering

We modify the implementation from [4], which uses the FAISS library [17] to perform an iterative process of K-Means clustering on the DINO features and obtain semantically segmented areas. The number of clusters is iteratively increased for each image until a sufficient amount of the data is explained (through a set elbow parameter) by the current number of clusters. The clustering process returns a centroid vector in the latent space (*not* a geometric centroid), which we utilize as our full object representation. Salient clusters are determined through a voting procedure based on the attention heads again following the method from [4], which determines which clusters are in the foreground and should be used for mapping. The salient areas are refined through erosion and GrabCut [18]. Our method to convert these semantic segmentations into instance level segmentations is through a connected components analysis on the salient clusters. We then calculate the geometric centroid of the salient clustered area and find the range to that point in order to obtain a point in space for that object. A complete description of the full object latent and geometric centroid extraction procedure is in Algorithm 1, with a sample result as compared to closed-set detectors seen in Figure 5. It is not obvious that using the centroid of an object in the latent space will be an effective and meaningful representation of the object that can be used for data association and mapping, since in fact e.g. the ear of a cat and the ear of a dog are more similar in the feature space than the ear of a dog and the tail of a dog [4]. Thus, one of the contributions of this work is to show experimentally that the latent space centroid can be used effectively as a lightweight representation for the entire object, and that the representation can be used consistently for data association across different viewing angles over the course of trajectory.

C. SLAM Framework

The problem of semantic SLAM is formulated as a *maximum a posteriori* (MAP) problem as follows:

$$X^*, L^* = \underset{X,L}{\operatorname{argmax}}\, p(X, L \mid Z). \tag{1}$$

where $X \triangleq \{x_i : x_i \in \mathrm{SE}(3), i = 1, \dots N\}$ are robot poses (provided in our method through the various sources of odometry described in Section IV-A), and $L \triangleq \{\ell_j \triangleq (\ell_j^\rho, \ell_j^c),\ j = 1, \dots, M\}$ are semantic landmarks, where each landmark ℓ_j is split into a continuous geometric component ℓ_j^ρ and a discrete semantic class component ℓ_j^c. Discrete class assignments are determined through equation 3 and are used only for admitting data association hypotheses. Thus we only need to consider the geometric component of the landmark measurements $\ell_j^\rho \in \mathbb{R}^3$. The associations between landmark observations and previously seen landmarks are considered unknown in our work, and thus these data associations must also be determined as explained in section III-D, thus making

Algorithm 1 Instance-Level Open-Set Object Detection and Localization

Given: DINO network Θ
Inputs: RGBD image I_{input}, patch size n, saliency threshold $thresh$, minimum object size min_size
Output: List of object centroid locations and encodings
for n x n patch P in I_{input} **do**
  DINO features $\leftarrow \Theta(P)$
Clusters $\leftarrow$ K-Means Clustering(DINO features)
for C in Clusters **do**
  Saliency voting using attention head weights
  if $C_{votes} > thresh$ **then**
    Salient Clusters $\leftarrow$ C
for Class in Salient Clusters Classes **do**
  Convert salient clusters of Class to binary image $I_{fg/bg}$
  $I_{fg/bg} \leftarrow Erode(I_{fg/bg})$
  $I_{fg/bg} \leftarrow GrabCut(I_{fg/bg})$
  $ConnectedComponents(I_{fg/bg})$
  for component in Connected Components **do**
    if $size(component) < min\_size$ **then**
      Remove from cluster list
    if $component \cap image\_border \neq \phi$ **then**
      Remove from cluster list
Retrieve geometric and feature space centroid for remaining clusters
Return: Latent space centroids, Geometric centroids

equation (1) with the inclusion of data association variable D

$$X^*, L^*, D^* = \underset{X,L,D}{\operatorname{argmax}}\, p(X, L, D \mid Z). \tag{2}$$

Through the conditional independence structure of the factor graph we can use iSAM2 [19] to obtain MAP estimates.

D. Data Association

We utilize a cosine similarity metric:

$$sim_{cos}(A, B) = \frac{A \cdot B}{\|A\|_2 \|B\|_2} \tag{3}$$

to determine whether objects are of the same class, where A and B are latent encodings from Algorithm 1. Given that two objects are of the same class as determined by a cosine similarity threshold α

$$sim_{cos}(A, B) > \alpha \tag{4}$$

we additionally check that the Mahalanobis distance between the observed object and existing object falls under a set threshold. Following the derivation from Kaess et al. [20], we can calculate the Mahalanobis distance between the landmark measurement $\tilde{\mathbf{z}}_k$ and landmark correspondence hypothesis $j_k = j$ given the state $\mathbf{x}$ and all previous measurements Z^- as

$$\begin{aligned} &P(\tilde{\mathbf{z}}_k, j_k = j \mid Z^-) \\ &\approx \frac{1}{\sqrt{|2\pi C_{i_k j}|}} e^{-\frac{1}{2}\left\| h_{i_k j}(\hat{\mathbf{x}}) - \tilde{\mathbf{z}}_k \right\|^2_{C_{i_k j}}} \end{aligned} \tag{5}$$

where $\|\mathbf{x}\|_{\Sigma}^2 := \mathbf{x}^T\Sigma^{-1}\mathbf{x}$, h is the measurement model, and the covariance C_{i_kj} is defined as

$$C_{i_kj} := \left.\frac{\partial h_{i_kj}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \Sigma \left.\frac{\partial h_{i_kj}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}}^T + \Gamma.$$

Thus our distance function is

$$G_{kj}^{2,\mathrm{ML}} := \|h_{i_kj}(\hat{\mathbf{x}}) - \tilde{\mathbf{z}}_k\|_{C_{i_kj}}^2 \tag{6}$$

where again we evaluate the hypothesis that a specific measurement $\tilde{\mathbf{z}}_k$ taken in image i_k was caused by the j^{th} landmark. Equation 6 follows a chi-squared distribution, and thus we can use a d-degree of freedom chi-square test:

$$G_{kj}^{2,\mathrm{ML}} < \chi_{d,\beta}^2 \tag{7}$$

If the observed object meets both the criteria (equation 7 and equation 4), we add the existing observation to the list of hypothesis data associations.

After checking against all existing objects within a given geometric distance, we rank the hypotheses based on the log marginal measurement likelihood. For max-likelihood, we simply select the most likely estimate given prior estimates as the associated landmark for the observation, and then use that fixed data association for computing landmark and pose estimates. For max-mixtures, we build a mixture factor as in [15], where we can marginalize out the data association factor given all measurements. For expectation maximization, we calculate the marginal MAP as in [21].

If there are no hypotheses that meet both of the criteria, the observation is determined to be of a new landmark, and a new landmark observation is added to the optimization.

IV. Experimental Results

A. Datasets

Experimental data was collected using an Intel RealSense D435i for RGBD data and an OptiTrack motion capture system for ground truth trajectories. A variety of objects were placed along the floor, and a Clearpath Jackal navigated through three loops around the room with the D435i mounted onboard.

We also evaluate our method on the publicly available TUM Pioneer Robot datasets [22] that include larger indoor scenes, for which we use the provided wheel odometry.

Finally, we evaluate our method on the much larger scale (2.2 kilometers) KITTI dataset [23], with visual-inertial odometry generated using RTABMAP [24]. The chosen datasets thus span a wide variety of scenes, object types, and lighting conditions, as well as a different odometry sources and sources of depth estimation, highlighting the generalizability of our method.

B. Alternate Methods Comparison

We compare our method with a popular existing open-set semantic mapping system, ConceptFusion [8]. ConceptFusion uses all available GPU memory quickly, and thus comparisons were made to a modified version of their implementation where the number of images used was downsampled to 1/70th of the total number of images processed by our system. We also provided their system with the same noisy odometry as our method since their built-in odometry system relies on frame-to-frame alignments that are not possible at subsampled framerates. Qualitative results for ConceptFusion are shown in Figure 2. As their method does not jointly optimize the trajectory, the trajectories are exactly equivalent to the provided odometry (i.e. Noisy Odometry in table I, Wheel Odometry in table II and Stereo Odometry in table III. However, note that for all datasets other than self-collected data, even with downsampling, the method exhausts all compute resources before completion.

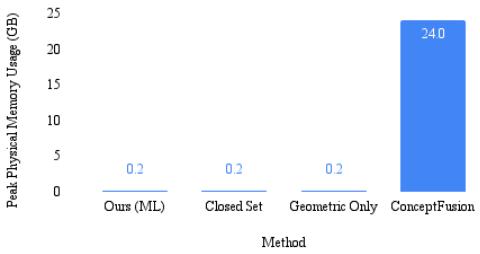


Fig. 3: Memory usage on sequence 1 of collected data. Sparse methods drastically reduce the memory consumption for open-set SLAM as compared to dense methods.

We also run a comparison against a modern closed-set detector (YOLOv8 [25]) whose class is then processed as a one-hot encoding which can be fed into the rest of our system. As an ablation, we run a method that chooses the most likely data association based only on Mahalanobis distance without using the landmark encoding cosine similarity metric which we refer to in results tables as Geometric Only. We provide a trajectory from ORBSLAM3 as a feature-based comparison. A systematic study of the effects of increasing the noise on the results of the system is presented in Table I. Results demonstrate that our method is able to successfully mitigate odometry-based drift through the use of open-set object detections, and furthermore, that the use of the object encodings aids the accuracy of the trajectory. Further discussion of the results is in Section V. Trajectories were evaluated against the ground truth using evo [26].

C. Compute Resource Usage

DINO feature extraction is completed as a preprocessing step as done in competing methods, at a rate 4.6 seconds per image on an NVIDIA GeForce RTX 3080 Ti Laptop GPU. Concurrent works such as [27] enable faster (50 Hz) open-set object detection that would remove our method's main bottleneck of DINO feature extraction. However, object encodings from their system would need to be further tested for suitability in our framework. The rest of our system runs in real time. Due to the sparse representation and the ability to close loops, the memory and storage usage of our system grows at a much slower rate than competing dense methods, while we simultaneously achieve accurate trajectories. A comparison of peak memory usage is presented in Figure 3.

TABLE I: Sequence 1, Average pose error in meters. ML: Max-likelihood, EM: Expectation-Maximization, MM: Max-Mixtures. Gaussian noise with base sigmas of $(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_r^2, \sigma_p^2, \sigma_y^2)$ = (0.001, 0.001, 0.001, 0.001, 0.001, 0.001) was added to the relative odometry measurements at each of the 1,838 keyframes, which occurred at 5 Hz. The indicated added noise multipliers were multiplied with the base sigmas. The entire trajectory was 21.7 meters. The ORBSLAM3 result is included without added noise as a baseline comparison. Only the best performing data association method result is included for closed-set and geometric only.

Added noise multiplier	Ours (ML)	Ours (EM)	Ours (MM)	Closed-Set (MM)	Geometric Only	Noisy Odometry
1	0.021	0.011	0.014	**0.007**	0.444	0.035
2	0.039	0.022	0.029	**0.019**	0.441	0.047
3	0.057	**0.024**	0.044	0.032	0.436	0.061
4	0.073	**0.031**	0.058	0.041	0.450	0.081
5	0.095	**0.038**	0.044	0.049	0.449	0.098
ORBSLAM3 (Baseline)	1.490					

TABLE II: TUM Pioneer Datasets results, Average Pose Error in meters. ML: Max-likelihood, EM: Expectation-Maximization, MM: Max-Mixtures. The pioneer SLAM dataset is 23.8 meters, while the Pioneer SLAM 2 dataset is 43.1 meters long. Only the best performing data association method result is included for closed-set and geometric only.

Dataset	Ours (ML)	Ours (EM)	Ours (MM)	Closed-Set (ML)	Geometric Only (ML)	Wheel odometry only
Pioneer SLAM	**0.139**	0.184	0.197	0.180	0.180	0.202
Pioneer SLAM 2	**0.231**	0.342	0.341	0.602	0.472	0.341

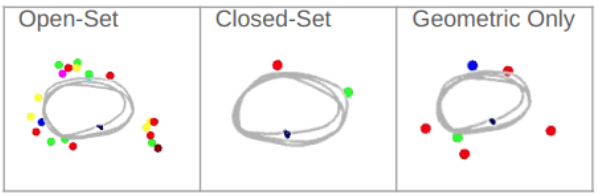
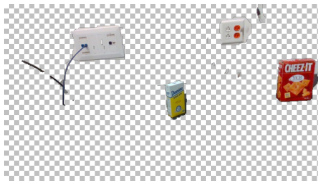


Fig. 4: The closed-set map identifies fewer objects in the scene. The geometric only data association method incorrectly associates objects in close proximity as being the same object as it does not have the object encoding to help differentiate those instances; the incorrect data associations result in a less accurate map and trajectory.
Object color to class mappings were identified by a human in post-processing as follows for open-set: Red - electric socket; Green - sugar box; Dark blue - CheezIt box; Yellow - spam can; Pink - skateboard; White - trash bag; Light blue - trash bin. For closed-set: Red - skateboard; Green - trash can. For geometric only (note that the classes were not used during mapping, and are identified for comparison purposes only): Red - electric socket; Green - sugar box; Dark blue - Skateboard.

(a) The detector incorrectly identifies the Cheez-It box as a book, and fails to identify the other three objects in the scene.



(b) An open-set detector is able to extract all objects in the scene and our method assigns each one an encoding vector.

Fig. 5: Images from the collected data were fed to YOLOv8 [25], a state-of-the-art and widely used object detector. The detector failed to identify many of the common everyday items in the scene, and even incorrectly labeled one item. Our open-set detector identifies each object and associates it with a latent vector encoding.

V. Discussion

It is interesting to note that by enabling the system to run open-set segmentation, several unanticipated objects were identified and used by the system. These include electric sockets, Ethernet sockets, and large scuffs on the wall.

While occlusions are not explicitly handled, sections of the trajectory do include partial and occluded views of objects. Occlusions remain a challenge even for existing methods since bounding box centroids and other common geometric representations encounter the same issue of inconsistently identified geometric centroids.

TABLE III: KITTI Sequence 05 results in meters. ML: Max-likelihood, EM: Expectation-Maximization, MM: Max-Mixtures. The trajectory is 2.21 kilometers long.

Method	Mean	Med.	Max.	RMSE	σ^2
Ours (ML)	4.460	3.269	12.184	5.454	3.139
Ours (EM)	**3.970**	3.161	**12.056**	**4.744**	2.600
Ours (MM)	4.037	**3.058**	13.238	4.805	2.606
Closed Set (ML)	12.427	12.514	15.270	12.511	1.450
Closed Set (EM)	5.037	3.172	17.063	6.236	3.675
Closed Set (MM)	4.220	3.171	15.091	5.263	3.144
Geo. Only (ML)	8.820	6.118	28.256	11.751	7.768
Geo. Only (EM)	9.656	7.289	27.929	12.398	7.777
Geo. Only (MM)	8.143	8.136	21.176	9.776	5.410
Stereo Odometry	4.948	3.894	16.191	5.886	3.188

The ORBSLAM3 baseline method for sequence 1 in Table I has high error due to the method losing tracking.

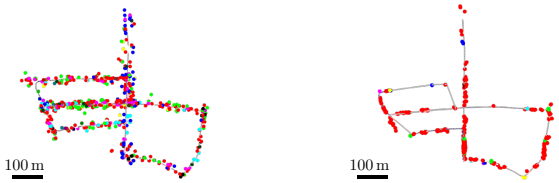
The max-mixture method for data association consistently performed worse than max-likelihood and expectation-maximization on the TUM dataset as seen in Table II, which is most likely due to the non-convexity of the cost landscape that is introduced by the method, which thus makes it difficult for the local optimization methods used in GTSAM to move to the global minimums. However, the ability to maintain and switch between multiple data association hypotheses allowed the max-mixture and expectation maximization methods to outperform max-likelihood on the other three datasets as seen in Table I and Table III.

Our method produces more accurate trajectories on the KITTI dataset thanks to the increased number of loop closure opportunities afforded by the open-set detector. The maximum error is relatively high across all methods, albeit lower in the open-set methods. This is due to the trajectory ending on a long straight road that moves away from previously seen areas (see the tops of Figure 6a and Figure 6b), meaning that there are no opportunities for loop closures regardless of the method used, and the trajectory accuracy is thus subject to odometric drift for that section. However, overall the lower maximum error seen for our methods is a result of the increased accuracy in all sections before the long straight road at the end.

Across all datasets, our method consistently outperformed the geometric only version of the system, thus highlighting the need for semantic descriptors. Our method also was able to improve upon the drift from pure odometry methods in all cases, with ConceptFusion trajectory results exactly equivalent to the pure odometry. Finally, independent of trajectory accuracy, our method produces more complete maps across all datasets as seen in Figure 4 and Figure 6.

VI. Conclusion

In conclusion, we present a novel system for tightly-coupled open-set semantic SLAM in sparse environments. We take an off-the-shelf image encoding network and run several post-processing steps to obtain instance-level object segmentations. We propose a lightweight single vector encoding for each object, and demonstrate that the object encoding is amenable to several data association methods in a factor graph-based SLAM framework. Our method is computationally more efficient than competing dense methods, and achieves high localization accuracy. By reasoning at the object level rather than at the pixel or dense feature level, the maps from our method are more semantically consistent than dense methods. Our method builds a more complete map and achieves higher localization accuracy than closed-set methods.



(a) The map and trajectory generated by the open-set method with max-mixtures data association. It identifies, maps, and localizes against 14 classes of objects.

(b) The map and trajectory generated by the closed-set method with max-mixtures data association. It identifies, maps, and localizes against 8 classes of objects.

Fig. 6: Trajectory and maps generated by open-set and closed-set methods for the KITTI dataset (sequence 05). The gray line represents the estimated trajectory, while different colored dots represent different object classes. The open-set method maps and localizes against more objects and object classes.

Acknowledgments

This work was supported by the MIT Lincoln Laboratory Autonomous Systems Line which is funded by the Under Secretary of Defense for Research and Engineering through Air Force Contract No. FA8702-15-D-0001, ONR grants N00014-18-1-2832, N00014-23-12164, and N00014-19-1-2571 (Neuroautonomy MURI), and the MIT Portugal Program.

References

[1] A. Y. Tolstonogov and A. D. Shiryaev, "The Image Semantic Compression Method for Underwater Robotic Applications," *Oceans Conference Record (IEEE)*, vol. 2021-September, 2021.

[2] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9630–9640, 2021.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 5999–6009, 2017.

[4] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep ViT Features as Dense Visual Descriptors," no. ii, 2021. [Online]. Available: http://arxiv.org/abs/2112.05814

[5] K. Singh, N. Rypkema, and J. Leonard, "Attention-based Self-Supervised Hierarchical Semantic Segmentation for Underwater Imagery," in *OCEANS 2023 - Limerick*, 2023, pp. 1–6.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," 2021.

[7] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection," 3 2023. [Online]. Available: http://arxiv.org/abs/2303.05499

[8] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. B. Tenenbaum, C. M. D. Melo, K. M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "ConceptFusion: Open-set Multimodal 3D Mapping Open-set Multimodal 3D Maps 3D Spatial Reasoning Open-set Queries for Autonomous Driving." [Online]. Available: https://concept-fusion.github.io/

[9] K. M. Jatavallabhula, S. Saryazdi, G. Iyer, and L. Paull, "gradSLAM: Automagically differentiable SLAM," 10 2019. [Online]. Available: http://arxiv.org/abs/1910.10672

[10] K. Mazur, E. Sucar, and A. J. Davison, "Feature-Realistic Neural Fusion for Real-Time, Open Set Scene Understanding," 10 2022. [Online]. Available: http://arxiv.org/abs/2210.03043

[11] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit Mapping and Positioning in Real-Time," 2021.

[12] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery," *IEEE Robotics and Automation Letters*, vol. 4, pp. 3037–3044, 7 2019.

[13] 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*, 2023.

[14] J. Fu, Y. Du, K. Singh, J. B. Tenenbaum, and J. J. Leonard, "NeuSE: Neural SE (3)-Equivariant Embedding for Consistent Spatial Understanding with Objects," *arXiv preprint arXiv:2303.07308*, 2023.

[15] K. Doherty, D. Baxter, E. Schneeweiss, and J. Leonard, "Probabilistic Data Association via Mixture Models for Robust Semantic SLAM," 2019. [Online]. Available: http://arxiv.org/abs/1909.11213

[16] K. J. Doherty, Z. Lu, K. Singh, and J. J. Leonard, "Discrete-Continuous Smoothing and Mapping," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 395–12 402, 2022.

[17] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search With GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[18] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut - Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.

[19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree," *The International Journal of Robotics Research*, vol. 31, pp. 217–236, Feb. 2012.

[20] M. Kaess and F. Dellaert, "Covariance Recovery from a Square Root Information Matrix for Data Association," *J. of Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1198–1210, Dec. 2009.

[21] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic Data Association for Semantic SLAM," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1722–1729.

[22] 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 International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[24] M. Labbé and F. Michaud, "RTABMAP Odometry," https://wiki.ros.org/rtabmap_odom, 2022.

[25] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[26] M. Grupp, "evo: Python Package for the Evaluation of Odometry and SLAM." https://github.com/MichaelGrupp/evo, 2017.

[27] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," 2024. [Online]. Available: https://arxiv.org/abs/2401.17270