Integrating Topological Object Recognition into Semantic SLAM for Unseen Cluttered Environments

Avania Bhattacharya[1], Ekta U. Samani[2], and Ashis G. Banerjee[3]

***Abstract*— Accurate semantic mapping is challenging for mobile robots in previously unseen environments, as vision-based SLAM pipelines are often sensitive to occlusion, viewpoint variation, and environmental clutter. We integrate topological object descriptors into Kimera Semantics for object-level semantic mapping and show that the topological recognition stage achieves higher recognition accuracy than YOLOv8 and DINOv2 RGB/RGB-D baselines across clutter levels and trajectories.**

I. INTRODUCTION

Accurate semantic mapping is essential for mobile robots operating in previously unseen environments. Semantic SLAM addresses this problem by jointly estimating the robot pose and building scene representations with semantic annotations, enabling tasks such as object manipulation and autonomous operation. Many existing semantic SLAM pipelines rely on vision-based deep learning models for object detection, segmentation, and recognition [1]–[3]. However, these methods are often sensitive to clutter, occlusion, and environmental variation, which degrade recognition performance and the resulting semantic map. Topological descriptors capture domain-invariant 3D shape structure and have shown robustness to occlusion and viewpoint variation [4], [5]. In this work, we integrate topological object descriptors into a semantic SLAM pipeline, namely Kimera Semantics, for object-level labeling and reconstruction in previously unseen cluttered environments.

II. METHODOLOGY

Fig. 1 provides an overview of the full semantic SLAM pipeline. Kimera Semantics provides visual inertial odometry (VIO)-based state estimation and 3D semantic mesh reconstruction [6]. In the standard pipeline, semantic labels are generated by a 2D semantic segmentation network [7], [8] and later fused with depth information. In our pipeline, object masks are first extracted from each RGB image using a NanoSAM-based instance segmentation network [9], with spatial prompts from a YOLOv8 object detector [10]. These masks are combined with the RGB-D input to obtain the corresponding object point clouds. Since the descriptor targets an object-scale structure [4], we filter out point clouds corresponding to large environmental regions and background clutter using heuristics based on distance, size, and point cloud density. We also maintain an exponential moving average objectness score across frames to reduce false positives [11]. At selected checkpoints along the trajectory, we compute topological descriptors, TOPS and TOPS2, for each object point cloud and perform recognition using THOR2 [5]. The resulting object labels are then propagated along the scene reconstruction meshes. Each labeled point

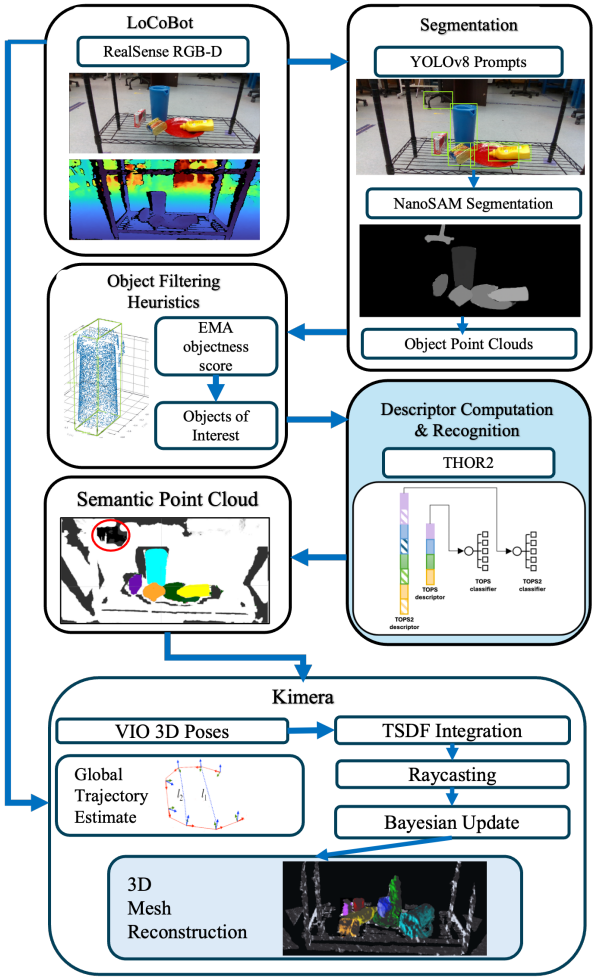

Fig. 1. Overview of the proposed pipeline. RGB-D input is processed using YOLOv8 and NanoSAM to generate segmentation maps and object-level point clouds. 3D heuristics remove background and large structures (red). Topological descriptors (TOPS/TOPS2) are computed at checkpoints and classified with THOR2. The labeled point clouds are transformed to the world frame using VIO and integrated into a TSDF, where labels are propagated via ray casting and fused through Bayesian updates to produce the final metric-semantic mesh.

[1]A. Bhattacharya is with Redmond High School, Redmond WA 98052, USA avaniabhattacharya@gmail.com

[2]E. U. Samani is with the Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213, USA esamani@andrew.cmu.edu

[3]A. G. Banerjee is with the Department of Industrial & Systems Engineering and the Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA ashisb@uw.edu

cloud is transformed to the world frame using the VIO estimates and integrated into a truncated signed distance field (TSDF) representation. We transfer labels to nearby surface voxels using bundled ray casting, and update voxel class probabilities through Bayesian fusion at each checkpoint. The final output is a global mesh whose vertices are assigned the most probable semantic label. Fig. 2 shows the results at different stages of the pipeline and the final reconstruction.

III. Experimental Results

We evaluated object recognition within the integrated semantic mapping pipeline on nine YCB object classes [12]. THOR2 was compared with three baselines: YOLOv8, DINOv2+CNN [13], and DINOv2 Depth [14]. In the latter, RGB images and depth maps, encoded from one to three channels, are processed using a frozen DINOv2 backbone, and the resulting RGB and depth features are fused using an MLP. All baselines were fine-tuned on the UW-IS-Occluded dataset [4] using only non-occluded scenes. The experiments were conducted in low (3-4 objects), medium (5-7 objects), and high clutter (6-8 objects) scenes with increasing occlusion. For each clutter level, we evaluated 18 scenes, each traversed using two trajectories: a standard trajectory with front and back views, and a varied trajectory with a 45°viewpoint offset. Recognition was performed at two and three checkpoints for the standard and varied trajectory, respectively.

At each checkpoint, accuracy was computed as the fraction of correctly recognized objects in the scene. The checkpoint accuracies were averaged to obtain a scene-wise accuracy. Table I reports the mean recognition accuracies and 95% confidence intervals across all scenes with the same clutter level and robot trajectory, and Table II presents the corresponding mean Intersection over Union (mIoU) results. mIoU scores are computed by comparing the predicted and ground-truth semantic labels at the voxel level for each reconstruction. The final mIoU is obtained by averaging across all classes, excluding unlabeled voxels.

Table I shows that THOR2 achieves the highest recognition accuracy across all clutter levels for both trajectories. The advantage is more pronounced in medium and high clutter scenes, where occlusion is more severe. DINOv2-based baselines remain competitive in low clutter scenes, but degrade more noticeably as clutter increases, while YOLOv8 performs worst overall despite a modest improvement on the varied trajectory. The mIoU results in Table II show similar trends across clutter levels and trajectories. THOR2 consistently achieves the highest mIoU, indicating an improved overall quality of the semantic reconstruction. The complete semantic mapping pipeline operates at 1.2 Hz on a LoCoBot platform, equipped with an Intel RealSense D435 camera using an on-board NVIDIA Jetson AGX Xavier.

IV. Discussion and Conclusions

In our experiments, most methods showed lower accuracy under the varied trajectory, consistent with the increased viewpoint change and occlusion in those trials. YOLOv8 was an exception: although its performance remained below that of THOR2 and DINOv2, its accuracy improved under the varied trajectory. We attribute this somewhat unexpected trend to lighting conditions. The standard trajectory was more directly aligned with the primary light source, increasing glare, whereas the angled trajectory reduced specular reflections. As an appearance-based detector, YOLOv8 appears to have benefited from these conditions. DINOv2 was less sensitive to reflections but struggled more with objects of similar color or partial shadowing, while THOR2 remained more robust across these cases.

The RGB baselines underperformed the methods that incorporate depth, but THOR2 achieved higher accuracy even among depth-based methods. Notably, this advantage was obtained without any real-world training data: THOR2's topological and color descriptor are computed from point clouds and trained exclusively on synthetic data, alleviating the need for the large-scale labeled datasets that deep learning baselines are constrained to. These results highlight that topology-based geometric approaches remain valuable, even as learning-based methods become increasingly prevalent.

However, THOR2 showed limitations when the center of an object was heavily occluded, when the objects leaned against each other, or when the rack partially occluded the object. More broadly, all the methods were sensitive to under- and over-segmentation. Improving segmentation accuracy, particularly during robot motion, remains an area of future work. The current pipeline also depends on an object detector to provide spatial prompts for segmentation. Replacing this stage with geometry or motion-based alternatives [15]–[17] is a promising direction for future work. Overall, these results show that topological descriptors can be integrated into Kimera Semantics to provide a more robust object-level labeling pipeline in unseen cluttered environments, which can support mobile manipulation tasks in the future.

References

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *IEEE Conf. Comp. Vis. Pattern Recognit.*, 2017, pp. 6243–6252.

[3] A. M. Webb, G. Brown, and M. Luján, "ORB-SLAM-CNN: Lessons in adding semantic map construction to feature-based slam," in *Annu. Conf. Towards Auton. Robotic Syst.* Springer, 2019, pp. 221–235.

[4] E. U. Samani and A. G. Banerjee, "Persistent homology meets object unity: Object recognition in clutter," *IEEE Trans. Robot.*, vol. 40, pp. 886–902, 2024.

[5] ——, "THOR2: Topological analysis for 3D shape and color-based human-inspired object recognition in unseen environments," *Adv. Intell. Syst.*, vol. 7, no. 4, p. 2400539, 2025.

[6] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1689–1696.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE Int. Conf. Comp. Vis.*, 2017, pp. 2961–2969.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[9] NVIDIA AI IOT, "NanoSAM: A distilled Segment Anything Model for real-time inference with TensorRT," 2024. [Online]. Available: https://github.com/NVIDIA-AI-IOT/nanosam

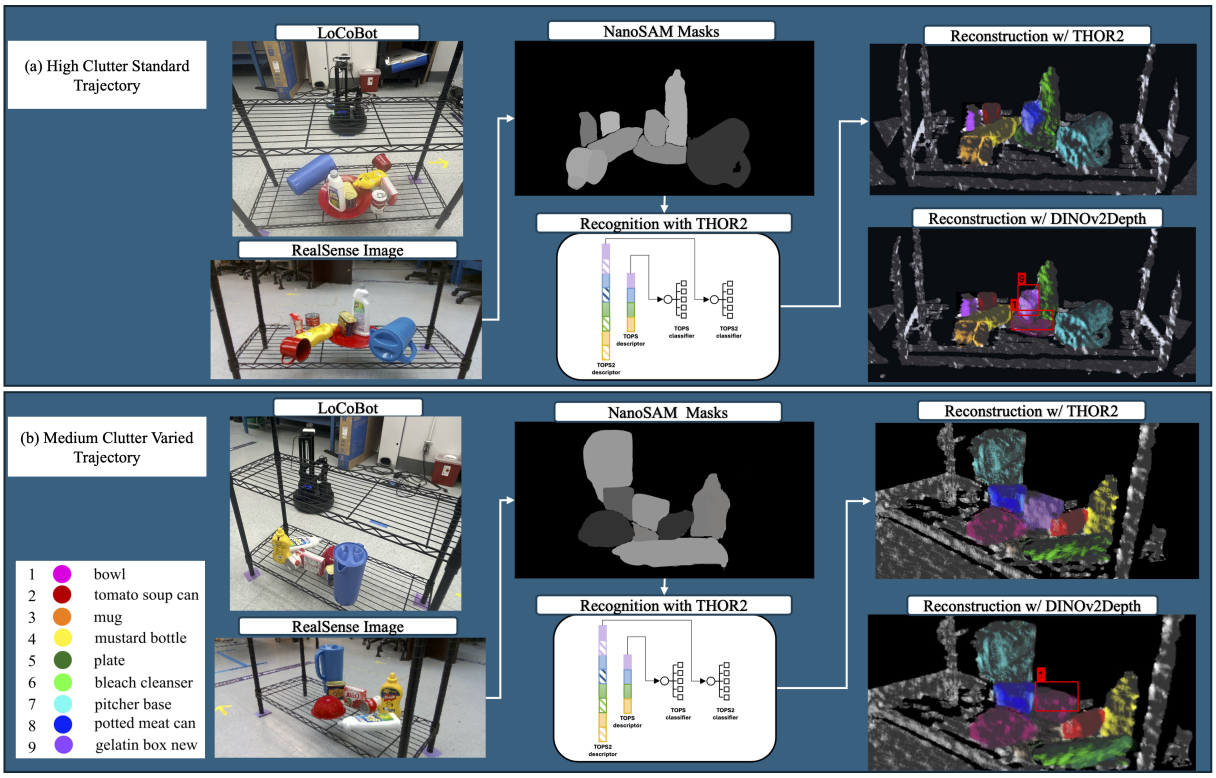

Fig. 2. Sample results from two experiments. YOLOv8 generates bounding boxes (i.e., spatial prompts) for NanoSAM instance segmentation, and THOR2 performs object recognition on the resulting object point clouds. Reconstructions using THOR2 labels are compared with those from the closest baseline, DINOv2 Depth. Red boxes on the reconstruction indicate incorrectly labeled objects.

TABLE I

MEAN RECOGNITION ACCURACY (%) WITH 95% CONFIDENCE INTERVALS ACROSS CLUTTER LEVELS AND TRAJECTORIES.

Trajectory	Clutter	YOLOv8	Mask R-CNN	DINOv2+CNN	DINOv2 Depth	THOR2
Standard Trajectory	Low	82.92 [74.44, 91.40]	85.50 [77.50, 93.50]	93.20 [87.37, 99.03]	94.30 [89.26, 99.34]	96.30 [93.18, 99.32]
	Medium	67.25 [57.67, 76.83]	70.50 [61.50, 79.50]	86.00 [78.40, 94.60]	85.25 [77.96, 92.54]	90.50 [84.08, 99.92]
	High	65.63 [60.37, 70.89]	68.50 [62.50, 74.50]	82.95 [77.10, 88.79]	83.79 [76.94, 90.63]	89.36 [83.49, 95.24]
Varied Trajectory	Low	79.17 [63.37, 94.97]	81.50 [66.00, 97.00]	92.71 [85.71, 99.71]	89.58 [80.35, 99.81]	91.67 [84.97, 98.37]
	Medium	75.45 [64.35, 86.54]	71.10 [65.50, 76.70]	87.67 [80.47, 94.87]	89.20 [82.20, 96.20]	91.22 [82.92, 99.52]
	High	72.17 [60.18, 84.16]	69.00 [57.00, 81.00]	71.82 [59.52, 84.11]	80.21 [70.28, 90.15]	89.35 [81.40, 97.32]

[10] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[11] R. Mendel, T. Rueckert, D. Wilhelm, D. Rueckert, and C. Palm, "Motion-corrected moving average: Including post-hoc temporal information for improved video segmentation," *arXiv preprint arXiv:2403.03120*, 2024.

[12] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, 2015.

TABLE II

MEAN INTERSECTION OVER UNION (MIOU) ACROSS CLUTTER LEVELS AND TRAJECTORIES.

Trajectory	Clutter	YOLOv8	Mask R-CNN	DINOv2+CNN	DINOv2 Depth	THOR2
Standard	Low	0.78	0.80	0.82	0.86	0.88
	Medium	0.70	0.72	0.76	0.83	0.84
	High	0.68	0.70	0.74	0.80	0.82
Varied	Low	0.75	0.77	0.80	0.85	0.86
	Medium	0.73	0.75	0.78	0.82	0.84
	High	0.60	0.63	0.70	0.74	0.80

[13] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res.*, 2024.

[14] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2015, pp. 681–687.

[15] L. Shao, P. Shah, V. Dwaracherla, and J. Bohg, "Motion-based object segmentation based on dense RGB-D scene flow," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3797–3804, 2018.

[16] J. Chen, Z. Kira, and Y. K. Cho, "LRGNet: Learnable region growing for class-agnostic point cloud segmentation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2799–2806, 2021.

[17] Y. Cao, Y. Wang, Y. Xue, H. Zhang, and Y. Lao, "FEC: Fast Euclidean clustering for point cloud segmentation," *Drones*, vol. 6, no. 11, p. 325, 2022.