Integrating Topological Object Recognition into Semantic SLAM for Unseen Cluttered Environments

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

***Abstract*— Reliable semantic mapping is challenging for mobile robots in previously unseen cluttered environments, as vision-based SLAM pipelines are often sensitive to occlusion, viewpoint variation, and environmental clutter. We integrate THOR2, a topological object recognition framework, into Kimera Semantics for object-level mapping and show that our approach achieves higher recognition accuracy than Mask R-CNN, YOLOv8, and DINOv2 RGB/RGB-D baselines across clutter levels and robot trajectories. Unlike conventional deep learning approaches, our method leverages domain-invariant shape-based features and improves robustness to partial observations. Integrating topological object recognition into semantic mapping is, therefore, a meaningful step toward scene understanding that generalizes to previously unseen cluttered environments and better supports downstream robot autonomy.**

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 collision-free navigation and object manipulation. 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. BACKGROUND: TOPOLOGY-BASED OBJECT RECOGNITION

In this work, we use TOPS and TOPS2, two descriptors designed for object recognition from RGB-D observations in cluttered scenes. TOPS, short for Topological features Of Point cloud Slices, is computed by slicing an object point cloud and applying persistent homology to each slice to capture the birth and death of connected components across a slicing-based filtration [4]. The resulting persistence

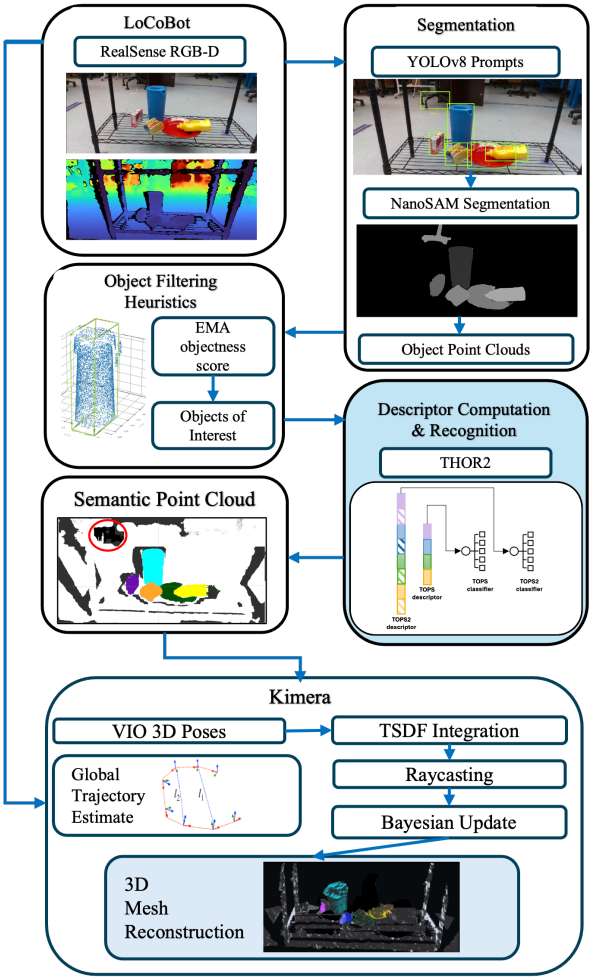

Fig. 1. Overview of the proposed pipeline. RGB-D input is processed using YOLOv8 and NanoSAM to generate segmentation masks and object-level point clouds. A set of 3D heuristics removes background and large structures (red). Topological descriptors (TOPS/TOPS2) of point clouds are computed at checkpoints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

images are vectorized and stacked to form a descriptor that summarizes the 3D shape of the object. TOPS embodies object unity, a human cognition mechanism: the slicing-based design preserves similarities between the descriptors of occluded objects and their corresponding unoccluded objects.

TOPS2 extends TOPS by incorporating color in addition to 3D shape. Specifically, TOPS2 retains the topological representation of 3D shape from TOPS, while capturing object color information through slicing-based color embed-

[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

dings using a network of coarse color regions [5]. These color regions are obtained using the Mapper algorithm, a topological soft-clustering technique. TOPS2 is intended to provide additional discriminative information for objects with similar geometry but different appearance.

THOR2 is the recognition framework that uses these topological descriptors for object recognition in unseen cluttered environments. Because THOR2 operates on point-cloud-based topological shape and color descriptors, it can be trained entirely on synthetic object data and then applied to real-world RGB-D images [5].

III. Methodology

The proposed pipeline takes an RGB-D sequence as input and produces a metric-semantic mesh in which reconstructed surfaces are assigned object-level labels. 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 Kimera pipeline, semantic labels are produced by a 2D segmentation model [7], [8] and fused with depth information. In our pipeline, object masks are extracted from each RGB frame using NanoSAM-based instance segmentation model [9], prompted by YOLOv8 bounding box detections [10]. These masks are combined with the depth map to yield per-object point clouds. Since the topological descriptor targets object-scale structure [4], we filter out point clouds corresponding to large environmental regions and background clutter using heuristics based on depth, bounding box area, and point cloud density. A per-instance validity score $s \in [0, 1]$ is maintained across frames using an exponential moving average for each tracked object. The score rises as the instance consistently passes the depth and bounding box heuristics, and decays when it does not, providing a heuristic measure of stability over time. Instances whose score falls below 0.5 are excluded from the topological descriptor, suppressing false detections arising from transient segmentation errors [11].

At selected checkpoints along the trajectory, we compute the TOPS shape descriptor and the TOPS2 shape-color descriptor for each retained object point cloud, and then pass these descriptors to THOR2 for recognition [5]. Each labeled point cloud is then transformed into the world frame using the VIO pose estimate and integrated into a truncated signed distance field (TSDF) map. Labels are propagated to nearby surface voxels via bundled ray casting, and each voxel maintains a probability distribution over the object classes that are updated at every checkpoint through a Bayesian fusion step combining the new observation with the accumulated prior. The final mesh assigns the maximum a posteriori (MAP) label to each vertex from its corresponding voxel distribution. Beyond the MAP label used for mesh reconstruction, the retained voxel-wise distributions give per-voxel uncertainty: observations consistent across multiple checkpoints converge to peaked distributions, while sparsely or ambiguously observed voxels have higher-entropy distributions. Fig. 2 illustrates representative pipeline outputs and the resulting reconstructions.

IV. Experimental Results

We evaluated object recognition within the integrated semantic mapping pipeline on nine YCB object classes [12]. THOR2 was compared to four baselines: Mask R-CNN, YOLOv8, DINOv2+CNN [13], and DINOv2 Depth [14]. For DINOv2 Depth, RGB images and depth maps encoded into three channels are passed through a frozen DINOv2 backbone, and the resulting features are fused via a multi-layer perceptron (MLP). Mask R-CNN [7] is the semantic segmentation model used in Kimera's original real-world experiments [6] and therefore represents the native Kimera pipeline. Unlike all the other baselines, evaluating Mask R-CNN does not invoke the NanoSAM-based instance segmentation pipeline introduced in this work. All the baselines were fine-tuned on the UW-IS-Occluded dataset [4] using only non-occluded training scenes.

Experiments were conducted in low (3–4 objects), medium (5–7 objects), and high (6–8 objects) clutter scenes with increasing levels of inter-object occlusion. For each clutter level, 18 scenes were evaluated, each traversed using two trajectories: a standard trajectory capturing front and back views, and a varied trajectory with a 45°viewpoint offset. Recognition was performed at two checkpoints per scene for the standard trajectory and three for the varied trajectory.

At each checkpoint, accuracy was computed as the fraction of correctly recognized objects. The checkpoint accuracies were averaged to yield a scene-level score. Table I reports mean recognition accuracies and 95% confidence intervals across all scenes that share the same clutter level and trajectory. Table II presents the corresponding mean Intersection over Union (mIoU) scores. mIoU is computed at the voxel level by comparing predicted and ground-truth semantic labels in the reconstructed mesh. For each class, IoU is the ratio of the overlap to the union of predicted and ground-truth voxels. The final mIoU is obtained by averaging per-class IoU scores, excluding unlabeled voxels.

V. Discussion and Conclusions

Table I shows that THOR2 achieves the highest recognition accuracy across all clutter levels and 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 Mask R-CNN, representing the native Kimera pipeline, outperforms YOLOv8 in most conditions, but falls short of both DINOv2-based methods and THOR2. YOLOv8 performs worst overall despite a modest improvement in the varied trajectory. The mIoU results in Table II reflect the same trends: THOR2 consistently achieves the highest mIoU, indicating superior overall quality in the reconstructed semantic mesh. The complete semantic mapping pipeline operates at 1.2 Hz on a LoCoBot equipped with an Intel RealSense D435 camera and an on-board NVIDIA Jetson AGX Xavier.

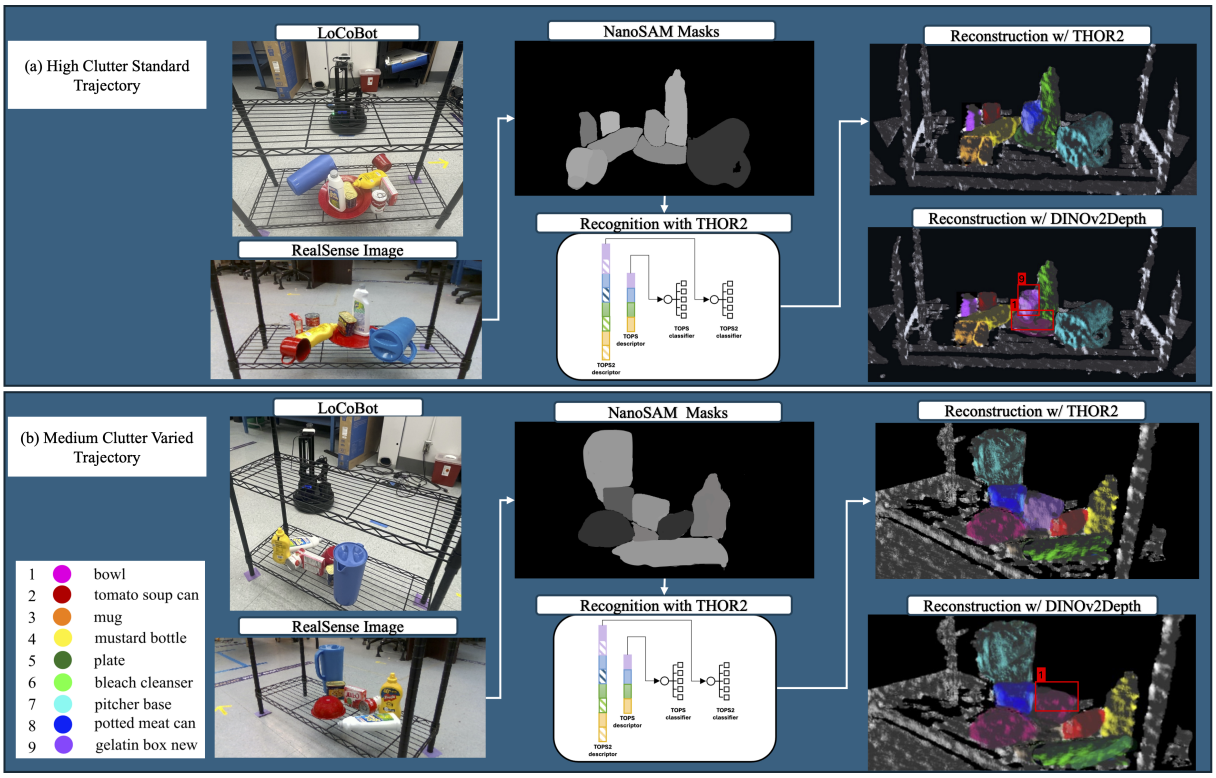

Fig. 2. Sample results from two experiments. YOLOv8 bounding boxes serve as spatial prompts for NanoSAM instance segmentation, and THOR2 performs object recognition on the resulting point clouds. Reconstructions produced using THOR2 labels are compared to those from the nearest baseline, DINOv2 Depth. Red boxes 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]

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

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 believe that this somewhat unexpected trend is due to lighting: 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 with objects of similar color or partial shadowing, while THOR2, relying on geometric rather than appearance-based features, remained more robust across these cases. Mask R-CNN performance fell below THOR2 across all conditions, highlighting the benefit of our pipeline modifications.

Although the RGB-only baselines underperformed depth-incorporating methods, THOR2 achieved the highest accuracy even among depth-based methods. Crucially, THOR2's topological and color descriptors are computed directly from point clouds and trained exclusively on synthetic data, with no access to real sensor captures during training [4], [5]. Our results suggest that for geometry-based representations, domain-invariant shape features can bridge a significant portion of this gap, potentially reducing the cost and effort

associated with curating large labeled real-world datasets. This is in contrast to the deep learning baselines, which required fine-tuning on real-world images to achieve competitive performance.

More broadly, all the methods were sensitive to under and over-segmentation. Segmentation maps can be incomplete or degraded due to YOLO prompt failures or NanoSAM errors. Objects absent from the segmentation map or where the majority of the object is not visible are excluded from evaluation; however, objects with some degree of under- or oversegmentation, where the masks keep substantial portions of the target objects, are retained. 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. It would reduce dependence on appearance-based models and improve generalization to truly novel environments.

Our results highlight that reliable semantic understanding cannot be assumed from appearance-based recognition alone, particularly as scene complexity increases: accuracy degrades across all baselines under higher clutter, while THOR2 remains robust to occlusion and viewpoint variation. The Bayesian fusion additionally yields per-voxel uncertainty estimates that could be utilized by uncertainty-aware planners in future work. Integrating a topological recognition stage into Kimera Semantics is therefore a meaningful step toward semantic SLAM pipelines that generalize to previously unseen environments and better support downstream robot autonomy.

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

[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.

[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.