Extremum-Seeking Active Object Recognition in Clutter Using Topological Descriptors

Vamsi K. Jonnalagadda[1], Chaitanya K. Mullapudi[2], Saniya Patwardhan[3], Ekta U. Samani[4], and Ashis G. Banerjee[5]

***Abstract*— Object recognition in unseen and cluttered indoor scenes is a challenging problem for semantic-level mapping and manipulation tasks involving low-cost mobile robots. In this paper, we propose a novel framework to address this problem through active robot navigation. Using this framework, the robot performs instance segmentation and identifies the objects using a 3D point cloud slicing-based topological descriptor. It also optimizes its pose autonomously via an extremum seeking controller to improve the identification confidence scores. Results show that our framework always improves the recognition success rate for any given scene as the robot moves to better pose(s), regardless of the number of objects in the scene, degree of clutter, distance to the objects, and lighting condition.**

I. Introduction

Object recognition is crucial for the robotic manipulation of objects. In environments such as warehouses and fulfillment centers, objects are often densely packed in storage, leading to strong visual occlusion of objects. Manipulation in such scenarios requires an object recognition system that is robust to occlusion. However, recognizing occluded objects is a challenging problem, especially when considering *one image frame at a time* [1], [2]. However, recognition systems mounted on a mobile platform, such as a mobile manipulator, can achieve better recognition by moving (the camera) to a better vantage point. This approach to recognition is often known as *active object recognition* [3], [4].

Several active recognition approaches have been proposed that perform multi-view recognition [5], end-to-end policy learning [2] and view-planning [6]. Alongside these works, we present an autonomous framework to actively recognize known objects in an unseen cluttered scene while being robust to variations in lighting conditions. Our approach is novel in two key ways. First, we use a 3D shape-based topological descriptor for recognition [1]. This descriptor enables training a recognition module exclusively using synthetically generated data that can be directly used in an unseen environment. Such a 3D shape-based descriptor also provides substantial robustness to variations in lighting conditions. Second, we use a model-free approach for viewpoint optimization, which does not require an explicit objective function or task model [7]. We describe our approach in Section II, report the experimental findings in Section III, and summarize the conclusions in Section IV.

*This work was supported UW + Amazon Science Hub, a collaboration between the University of Washington, Seattle and Amazon, Seattle

[1]V. K. Jonnalagadda is with the Department of Mechanical Engineering, University of Washington, Seattle WA 98195, USA jnvk@uw.edu

[2]C. K. Mullapudi is with the Department of Electrical & Computer Engineering, University of Washington, Seattle WA 98195, USA ckm26@uw.edu

[3]S. Patwardhan is with the Department of Mechanical Engineering, Indian Institute of Technology Gandhinagar, Palaj, Gujarat 382355, India patwardhan.saniya@iitgn.ac.in

[4]E. U. Samani was with the Department of Mechanical Engineering, University of Washington, Seattle WA 98195, USA eusamani@gmail.com

[5]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

II. Method: NanoSAM + TOPS + NN-ESC

Given an unseen cluttered scene, we aim to recognize all the objects in the scene using a mobile robot with a fixed camera. First, we capture an RGB-D image of the scene and use NanoSAM [8], a variant of Segment Anything Model (SAM) [9], to generate an instance segmentation map of the scene from the RGB image. Subsequently, we use the instance segmentation map and the corresponding depth image to generate point clouds for every object. We then compute 3D shape descriptors for these point clouds and use a multilayer perceptron (MLP)-based classifier to perform recognition. We use the recognition predictions and the corresponding confidence scores (probabilities) to compute an objective value. We then use a Neural Network-Extremum Seeking Controller (NN-ESC) [7] to optimize this objective value and navigate the robot to a different pose for improved recognition. This entire process constitutes one iteration of the active object recognition pipeline. The pipeline runs several iterations until the robot reaches a pose where the confidence scores of all the objects in the scene are above a certain threshold C. Fig. 1 illustrates the robot and the modules in our pipeline. The individual modules are described in the following subsections.

A. Segmentation Mask Generation using NanoSAM

NanoSAM requires spatial prompts or bounding boxes around the objects in a scene to generate segmentation masks. Generating good spatial prompts for SAM (or its lighter versions, such as MobileSAM [10] and NanoSAM) is a research problem in itself [11]. In this work, we compute spatial prompts as follows. First, we capture a reference RGB image of the environment without any objects. Whenever a robot looks at a cluttered scene, first, we perform background subtraction using this reference to obtain a foreground image. We then compute AKAZE local features [12] from the foreground image to detect keypoints. These keypoints are

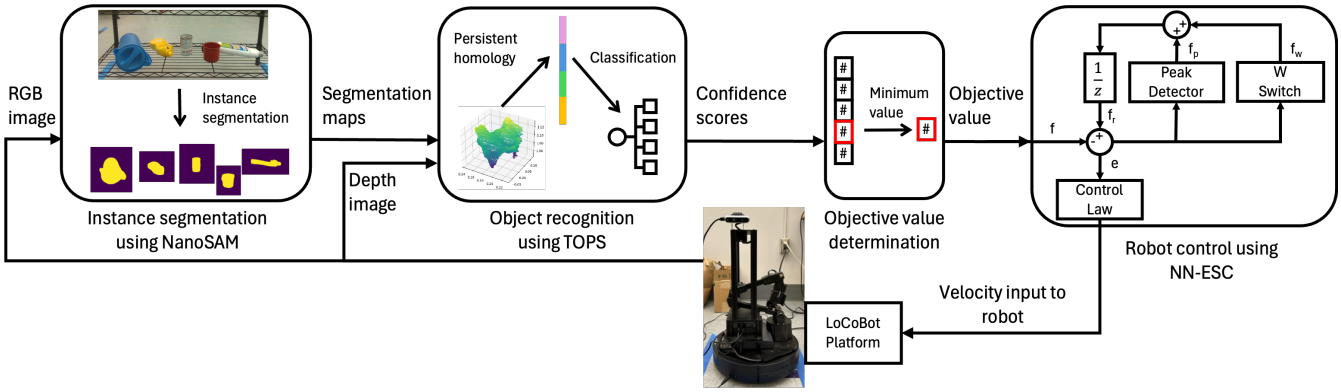


Fig. 1: Proposed framework for active object recognition using NanoSAM, point cloud slicing-based topological descriptors, and NN-ESC implemented on a LoCoBot.

then used as initial spatial prompts for NanoSAM. Note that this procedure is performed only once during the first iteration. To obtain the prompts for subsequent iterations, we use the CSRT object tracker [13]. It tracks the location of the objects in the scene as the robot moves and updates the relevant key points to avoid repeated keypoint computation in subsequent iterations.

B. Object Recognition via TOPS

We use the instance segmentation map obtained from NanoSAM along with depth image to obtain 3D point clouds of all the objects in the scene. We then compute 3D shape descriptors for every object. In particular, using persistent homology, we compute the previously presented topological descriptor known as TOPS [1]. TOPS, which stands for Topological features Of Point cloud Slices, is designed for recognizing partially occluded 3D point clouds. As in [1], we train an MLP classifier using training data that comprises synthetic depth images corresponding to all the possible views of all the objects. We use this model to obtain recognition predictions and corresponding prediction probabilities. These probability values are referred to as *confidence scores*, which we use to obtain the objective value to be optimized.

C. Objective Value Determination

Given a threshold C, the confidence scores output by the model for all the objects in a scene may not always be greater than that value in the first iteration. This is attributed to factors including, but not limited to, object poses and occlusion, lighting conditions, and distance to the scene. These conditions can be improved by moving the robot to a different pose, which can result in better confidence scores. However, from a control perspective, it is extremely difficult to model an objective function for confidence scores that can be optimized. Therefore, we implement a heuristic approach by choosing the least confidence scores as the objective value in every iteration and optimizing until it crosses the threshold C. Since the objective function for optimizing the confidence scores is unknown, NN-ESC is a good candidate for this optimization problem.

D. Optimization using NN-ESC

We implement an NN-ESC similar to [7], but in a discrete-time viewpoint optimization loop. To perform active object recognition effectively, we constrain the motion of the robot in a circle with a fixed radius r around the objects such that the camera points at the objects throughout the active recognition task. This requires the NN-ESC algorithm to provide a 1-D velocity vector as one of the inputs to the differential drive robot, which becomes its tangential velocity v_r. The other input, angular velocity ω_r, is automatically fed to the robot as:

$$\omega_r = \frac{v_r}{r}$$

We formulate the control law for the robot as follows: At time t_k, the controller uses the error $e[k]$ calculated as the difference between the reference value $f_r[k]$ and the objective value $f[k]$ to compute an appropriate control law for the system.

$$e[k] = f_r[k] - f[k], \quad k = 1, 2, 3, ... \tag{1}$$

The algorithm updates the reference value in every iteration with the help of W Switch and Peak Detector outputs f_w and f_p, respectively.

$$f_r[k] = \begin{cases} f[k], & k = 1 \\ f_r[k-1] + f_w[k-1] + f_p[k-1], & k = 2, 3, .. \end{cases} \tag{2}$$

In the first iteration, i.e., at $k = 1$, the W switch is OFF ($f_w[1] = 0$), which makes the reference value for our objective (confidence score) unknown. Therefore, we initialize it with the initial objective value ($f_r[1] = f[1]$), thereby making the initial error equal to zero ($e[1] = 0$). The peak detector updates the f_p value as follows:

$$f_p[k] = \begin{cases} M, & (e[k] < 0) \\ 0, & (e[k] \geq 0) \end{cases} \tag{3}$$

To ensure convergence to the maximum objective value, the error should converge to zero. This imposes a condition on the value of the positive constant M that it should

be greater than the absolute difference between any two consecutive objective values.

$$M > |f[k+1] - f[k]| \tag{4}$$

We initialize the system with the control law $u[1] = -U$, where U is a positive constant. For the subsequent iterations, we define the control law as:

$$u[k+1] = \begin{cases} -U, & e[k] < -\delta \\ U, & e[k] > \delta \\ u[k], & \text{otherwise} \end{cases} \tag{5}$$

The positive constant δ is used to define the hysteresis width $[-\delta, \delta]$. As long as the error stays within this interval, the previous state of the control law is retained so that the objective value to its optimum value. If the error is outside the bounds of the interval, the control law switches the direction of the robot. This switching mechanism can be interpreted as follows: initially, for $k = 1$, the sign of u is negative. If this control law drives the system to a state where the error exceeds the right boundary of the hysteresis width, the control law switches the sign of u, i.e., the robot moves in the opposite direction. Continuing in the same direction, if the error exceeds the left hysteresis boundary, the switching happens once again. This switching perpetuates as the f_r and f_p values keep updating in every iteration (as described in (2) and (3)) until and unless the error exceeds a threshold $\Delta(> \delta)$. Once the error crosses the threshold Δ, the controller activates the W switch to reset the f_r value (as described in (2)) using the following switching function:

$$f_w[k+1] = \begin{cases} -W, & (e[k+1] > \Delta) \\ 0, & (e[k+1] < -\Delta) \\ f_w[k], & \text{otherwise} \end{cases} \tag{6}$$

If the error is above the threshold Δ, the f_r value in (2) decreases because of the negative value $-W$. Consequently, from (1), the error eventually decreases and goes below $-\Delta$, thereby taking the control law to its initial state (as described in (5)).

III. Results and Discussion

We use a collection of 10 objects[1] from the YCB object set [14] to evaluate the performance of our pipeline with respect to the changes in degree of clutter, distance between the robot and the objects, and lighting conditions. We consider 40 sequences of scenes created using 5 randomly chosen objects from the collection. Similarly, we create another 48 sequences of 6-object scenes. For both the 5-object and 6-object scenes, we divide the corresponding sequences into eight groups each by varying the degree of clutter (less and moderate), distance of the robot from the objects (near and far), and background lighting (standard and dim). For every sequence, we record the number of iterations and the time it takes to recognize all the objects above the threshold C, segmentation success rate, and recognition success rates.

We implement the active recognition pipeline in a Python environment on a LoCoBot equipped with an Intel RealSense D435 camera and an NVIDIA Jetson AGX Orin processor. We use the ROS Kinetic distribution on the LocoBot platform to run NN-ESC. For our experiments, the NN-ESC parameters are $M = 0.7$, $W = 1$, $\delta = 0.1$, and $\Delta = 0.2$. U is chosen to be 0.2 and 0.35 for near and far instances of the robot, respectively, to maintain the same angular velocity. We set the threshold value to $C = 90\%$.

Table I summarizes the results for all the 16 groups. Based on the initial and final recognition success rates—percentage of confidently predicted, correctly recognized objects in a given scene (see columns 8 and 9 in Table I), we conclude that our pipeline actively improves object recognition performance in any scene as compared to (static) single-shot recognition. This is true regardless of the number of objects, clutter level, target distance, and variations in background illumination. As might be expected, the best recognition performance is seen in scenarios with fewer objects in the scene, less clutter, proximity of the robot to the objects, and well-lit environment.

Representative examples of our active recognition pipeline are shown in Fig. 2 and Fig. 3 for dim and standard lighting conditions, respectively. In Fig. 2, the robot recognizes all the five objects correctly within three iterations, whereas, it predicts five out of the six objects correctly in two iterations in Fig. 3. Notably, in both the cases, the recognition success rates increase over time as the robot actively moves to different viewpoints.

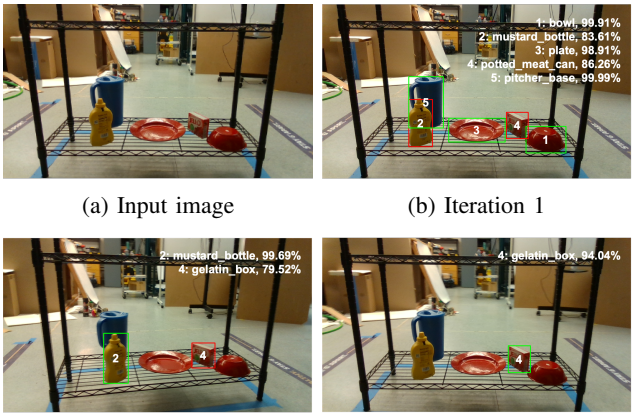
In general, we observe a drop in recognition performance in any scenario where NanoSAM outputs oversegmented (partial or incomplete) or undersegmented (compound masks for multiple objects) masks of the objects. Interestingly, the CSRT object tracker impacts the recognition in both beneficial and detrimental ways. For the objects that are initially oversegmented or undersegmented, the corresponding bounding boxes tend to grow or shrink in size in subsequent iterations, sometimes resulting in (more) complete masks that lead to correct recognition. On the other hand, when the robot is in motion, if the tracked object is occluded by a neighboring object, the bounding box may jump on to the latter object and cause inaccurate prediction using TOPS.

It is worth noting that the confidence scores output by the MLP-based classifier are not very reliable, as some objects in the scenes are wrongly predicted with high confidence. We intend to explore other classifiers, such as a Support Vector Machine (SVM), to mitigate this problem. Further, the object appearance information (color, texture, etc.) are currently not considered in the topological descriptor. We plan to explore this possibility in the future to offset the segmentation performance issues and enhance recognition accuracy.

[1]While there are many other objects in the YCB object set, we only choose pitcher base, plate, bowl, mustard bottle, bleach cleanser, mug, potted meat can, foam brick, gelatin box, and tomato soup can for the experiments as they consistently yield better segmentation masks than the others, which are necessary for accurate object recognition.

TABLE I: Summary of active object recognition results

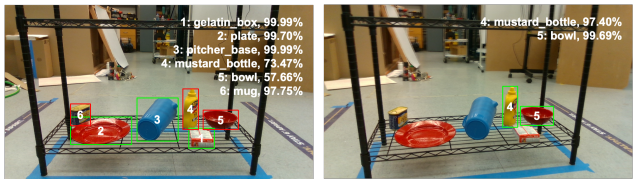
# of objects	Clutter level	Distance	Lighting	# of iterations	Recognition time (s)	Segmentation success rate (%)	Recognition success rate at first iteration with $> 90\%$ confidence (%)	Recognition success rate at the end with $> 90\%$ confidence (%)
5	Less	Near	Standard	3.60±1.34	44.92±1.08	92.00±10.95	56.00±8.94	76.00±8.94
			Dim	3.60±3.13	45.18±25.41	92.00±10.95	56.00±8.94	72.00±17.89
		Far	Standard	5.40±2.07	59.48±16.55	92.00±10.95	28.00±10.95	56.00±16.73
			Dim	5.00±3.32	56.62±27.05	92.00±10.95	32.00±26.83	60.00±20.00
	Moderate	Near	Standard	3.40±2.19	43.47±18.30	100.00±0.00	48.00±22.80	60.00±14.14
			Dim	4.40±2.70	52.09±22.26	88.00±17.89	40.00±14.14	48.00±22.80
		Far	Standard	5.40±3.13	59.13±25.63	68.00±10.95	24.00±8.94	40.00±0.00
			Dim	3.60±1.95	44.52±16.28	80.00±0.00	24.00±8.94	36.00±21.91
6	Less	Near	Standard	4.50±5.21	53.53±43.32	97.22±6.80	41.67±17.48	58.33±17.48
			Dim	3.83±3.13	47.05±26.80	91.67±9.13	47.22±24.53	58.33±22.97
		Far	Standard	4.50±2.81	51.87±23.44	88.89±13.61	41.67±13.94	52.78±16.39
			Dim	5.33±4.37	65.20±38.92	94.44±13.61	41.67±9.13	52.78±12.55
	Moderate	Near	Standard	4.17±1.17	50.08±10.50	86.11±19.48	44.44±20.18	55.56±13.61
			Dim	2.67±1.21	37.15±10.07	80.56±19.48	38.89±17.21	50.00±18.26
		Far	Standard	4.67±3.20	53.04±26.90	75.00±9.13	30.56±19.48	38.89±17.21
			Dim	4.83±2.14	60.50±16.92	77.78±13.61	27.78±17.21	38.89±13.61



(a) Input image (b) Iteration 1

(c) Iteration 2 (d) Iteration 3 (final)

Fig. 2: Results from a scenario with 5 objects in a dimly lit scene showing the predictions of objects and their confidence scores. The green bounding boxes represent correct and confident predictions, whereas, red boxes represent incorrect or non-confident predictions. The robot recognizes more objects correctly as it actively moves to better viewpoints.



(a) Iteration 1 (b) Iteration 2 (final)

Fig. 3: Results from a scenario with 6 objects in a well lit scene showing the predictions of objects and their confidence scores. The robot recognizes additional objects correctly as it actively moves to a better viewpoint.

IV. Conclusions

In this work, we present a topological descriptor-based framework to actively recognize objects in unseen cluttered scenes using a mobile robot equipped with an RGB-D camera. Given a scene with known objects, the robot automatically detects key points, generates segmentation maps using AKAZE features and NanoSAM, and performs on-the-fly object recognition based on TOPS and an MLP-based classifier. We employ NN-ESC to actively optimize the robot's pose such that the recognition confidence for all the objects in a scene is above a predefined threshold value. Implementation on a LoCoBot platform shows that the recognition success rate always increases as the robot moves to different viewpoint(s), regardless of the degree of clutter, distance to the scene, and variations in illumination. In the future, we plan to use the recognition labels as semantic information for mapping of cluttered scenes and mobile manipulation of the scene objects.

References

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

[2] D. Jayaraman and K. Grauman, "End-to-end policy learning for active visual categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1601–1614, 2018.

[3] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. J Comput. Vis.*, vol. 1, pp. 333–356, 1988.

[4] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *Comput. Vis. Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[5] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3813–3822.
[6] R. Zeng, Y. Wen, W. Zhao, and Y.-J. Liu, "View planning in robot active vision: A survey of systems, algorithms, and applications," *Comput. Vis. Media*, vol. 6, pp. 225–245, 2020.
[7] B. Calli, W. Caarls, M. Wisse, and P. P. Jonker, "Active vision via extremum seeking for robots in unstructured environments: Applications in object recognition and manipulation," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1810–1822, 2018.
[8] John, "Nvidia-ai-iot/nanosam: A distilled segment anything (sam) model capable of running real-time with nvidia tensorrt," 2023. [Online]. Available: https://github.com/NVIDIA-AI-IOT/nanosam.git
[9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 4015–4026.
[10] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster segment anything: Towards lightweight sam for mobile applications," *arXiv preprint arXiv:2306.14289*, 2023.
[11] J. Huang, K. Jiang, J. Zhang, H. Qiu, L. Lu, S. Lu, and E. Xing, "Learning to prompt segment anything models," *arXiv preprint arXiv:2401.04651*, 2024.
[12] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.
[13] A. Lukezic, T. Vojir, L. Cehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, July 2017.
[14] 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 Rob. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, 2015.