Online Object-Level Semantic Mapping for Quadrupeds in Real-World Environments

Emad Razavi
Dynamic Legged Systems
Istituto Italiano di Tecnologia
DIBRIS, University of Genova
Genoa, Italy
5782734@studenti.unige.it

Angelo Bratta
Dynamic Legged Systems
Istituto Italiano di Tecnologia
Genoa, Italy
angelo.bratta@iit.it

João Carlos Virgolino Soares
Dynamic Legged Systems
Istituto Italiano di Tecnologia
Genoa, Italy
joao.virgolino@iit.it

Carmine Recchiuto
RICE lab
University of Genova
Genoa, Italy
carmine.recchiuto@unige.it

Claudio Semini
Dynamic Legged Systems
Istituto Italiano di Tecnologia
Genoa, Italy
claudio.semini@iit.it

Abstract—We present an online semantic object mapping system for a quadruped robot operating in real indoor environments, turning sensor detections into named objects in a global map. During a run, the mapper integrates range geometry with camera detections, merges co-located detections within a frame, and associates repeated detections into persistent object instances across frames. Objects remain in the map when they are out of view, and repeated sightings update the same instance rather than creating duplicates. The output is a compact object layer that can be queried (class, pose, and confidence), is integrated with the occupancy map and readable by a planner. In on-robot tests, the layer remained stable across viewpoint changes.

Index Terms—mapping, quadruped robots, semantic mapping, object detection

I. Introduction

Accurate mapping is fundamental for autonomous navigation. In complex environments featuring uneven terrain, low friction, gaps or stairs, wheeled robots struggle, while bipedal robots are usually unstable and slow. Quadrupeds, however, offer advantages in mobility and stability in such scenarios [1].

In these settings, navigation goals are often tied to objects rather than geometric locations, e.g., "find the door", "go to the toolbox", or "stop by the charger." A purely geometric map could be insufficient in such scenarios. Object-level semantics can help a planner decide where to go [4].

Earlier work showed that maps can combine geometry with object and location labels, giving planners a representation they can query directly for tasks like "go to the kitchen" [2], [3]. Learning-based methods train a policy that maintains a category-level semantic map from RGB–D (via both segmentation and detection) and conditioned on the goal class, scores frontiers or regions to select the next waypoint [4]. Multi-modal mapping fuses RGB-D images with LiDAR data to reduce depth measurement errors and can improve 3D placement. [5]. On quadrupeds used in search-and-rescue scenarios, semantic layers focus on people and structural cues
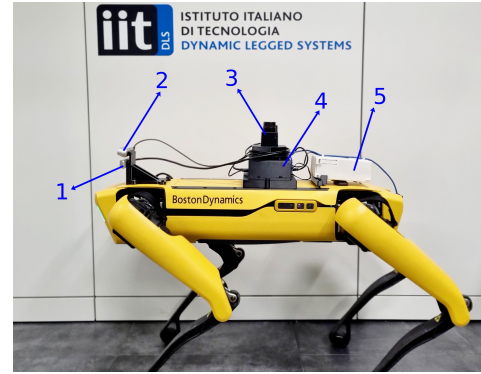

Fig. 1: Spot by Boston Dynamics with onboard payload: 1. Intel RealSense T265 Camera, 2. Intel RealSense D435 Camera, 3. 2D LiDAR, 4. Intel NUC 11 i7-1165G7, and 5. A lithium-ion battery to power the NUC.

such as cracks, stairs, and doors, confirming the value of task-specific object information [6]. Recent open-vocabulary and vision–language methods add flexible labels and frontier scoring, but they often require comparatively heavy computation and dense 3D updates [7]–[9]. In parallel, many practical Spot deployments such as industrial sites, tunnels, and labs, still rely on prior maps for stable localization in confined or hazardous spaces, which underlines the need for reliable, lightweight mapping substrates on real robots [1], [10].

In this work, we present a lightweight semantic mapping method for a quadruped robot. The system fuses RGB-D camera object detections with LiDAR scans and T265 odometry to project the objects into a global map, merging nearby duplicates per frame, and associating repeated observations over time. Fig. 2 summarizes the methodology. We avoid heavy 3D fusion and large vision–language scoring loops that raise latency and power draw [7]–[9], being consistent with onboard computational constraints [1], [10]. The platform is
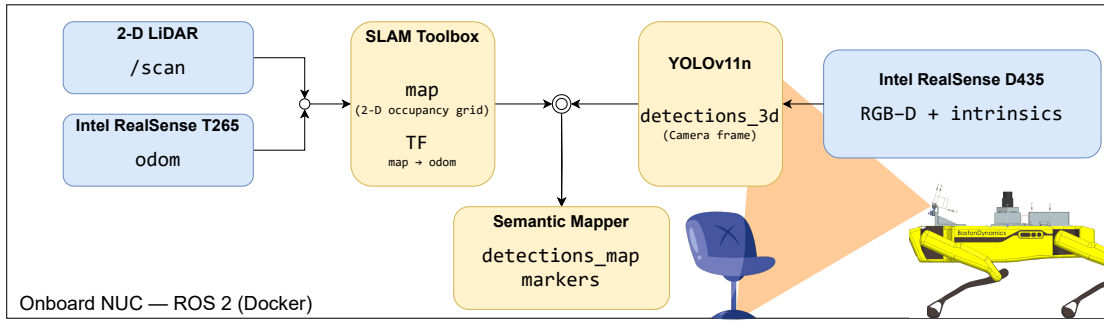
Fig. 2: Overview of the proposed methodology. A 2D occupancy grid map is built using 2D LiDAR scans and odometry obtained from the T265 camera. Object detections are obtained from RGB images, and depth is used to estimate 3D positions. The semantic layer places detected objects on the map, merges nearby ones, and keeps them over time.

a Boston Dynamics Spot equipped with a sensor payload and an onboard NUC (Fig. 1).

II. METHODS

A. Mapping (SLAM Toolbox with T265 odom and 2D LiDAR)

We build a 2D occupancy map online with the *SLAM Toolbox* [11]. A 2D LiDAR provides range geometry, while the Intel RealSense T265 provides visual–inertial odometry. The static transformations between the robot body frame, the LiDAR, and the T265 are fixed (calibrated once). During operation, the SLAM Toolbox uses visual odometry computed by the T265 tracking camera as a motion prior and refines it with scan matching and pose-graph loop closure, limiting drift and maintaining a globally consistent estimate. This occupancy map provides the geometric base for the semantic layer described next.

B. Visualization and Detection (D435 + YOLOv11)

The Intel RealSense D435 provides synchronized RGB–D input for detection and projection: color images, depth aligned to the color stream, and the associated camera intrinsics [12]. We also use a static transform between the robot body frame and the camera frame.

For object detection, we use YOLOv11 [13]. Each color frame is preprocessed to the model input; outputs are class labels, scores, and 2D bounding boxes in the camera frame after non-maximum suppression. We publish a lightweight detection topic and an RViz overlay. Depth fusion and map anchoring are handled by the semantic layer in the next subsection.

C. Semantic object layer: association and memory

We transform each 3D detection into the map. We set the height (z-axis) to 0 and keep only the yaw rotation (roll and pitch are set to zero) to have a 2D map. We drop low-score detections by using per-class thresholds. Using Euclidean distance on the map, within the same frame, we remove near-duplicates by keeping only the highest-score detection for a class when two appear very close on the map. The key parameters used in our runs are summarized in Table I.

TABLE I: Core settings for association and memory.

Per-class confidence cutoff	0.65
Same-frame merge radius [m]	0.20
Reuse radius in map [m]	0.80
Promotion gate (hits / s / mean)	10 / 2.0 / 0.50

We have two memories: a short-term buffer of recent observations and a long-term list of confirmed objects. For each detection, we first check the long-term list with a nearest-neighbor search (NNS) in the 2D map. If a same-class object is nearby, we recognize it as seen again, increase its detection count by one, and leave its stored pose unchanged. Otherwise, we compare it with the short-term buffer: if no same class is within the radius, we start a new short-term record; if one is close, we update it with the new measurement. A new object is added to the long-term list only after the same class is re-observed several times within a short interval at approximately the same position and with sufficient confidence. Finally, we publish the current set of confirmed objects in the map frame. Fig. 3 summarizes the association and memory logic.
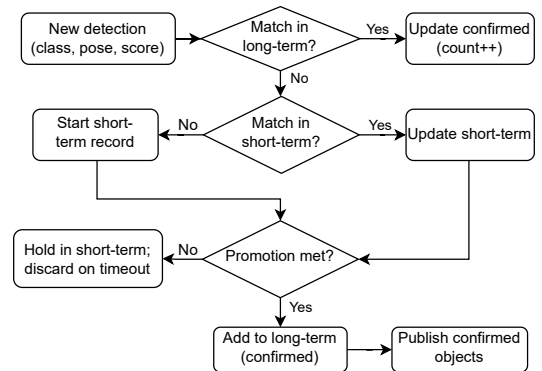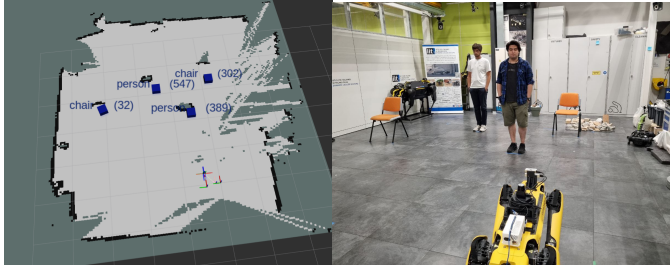

Fig. 3: Association pipeline for confirmed objects.

(a) RViz—map with confirmed objects (b) Lab scene (people, chairs)

Fig. 4: Setup and output. The semantic layer tracked only `person` and `chair`.
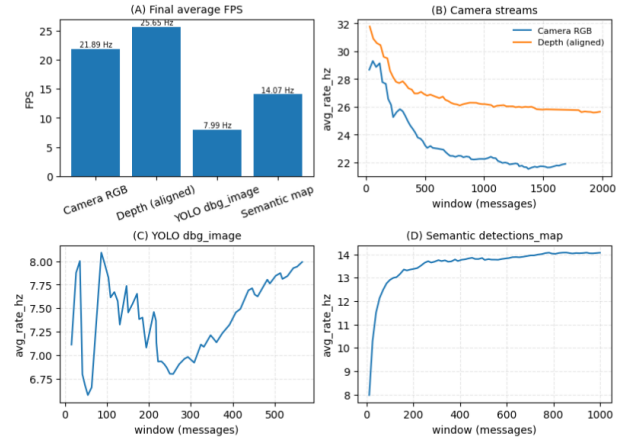

Fig. 5: (A) Final FPS. (B) Camera RGB and aligned depth, average rate vs. cumulative messages (window). (C) YOLO debug-image rate. (D) Semantic object-map rate. Note: the YOLO stream publishes only when detections exist.

III. RESULTS

We mounted a D435 RGB–D, a T265, and a 2D LiDAR on the quadruped and ran the SLAM Toolbox with our semantic mapper inside a ROS 2 container. An operator drove the robot with a joystick through a lab with two people and two chairs, as shown in Fig. 4(b). As the robot moved, the SLAM system built a 2D occupancy map, while the semantic mapper projected detections into the map in real time[1]. Using the depth images, we computed a 3D position for each detection and transformed it into the global map via the calibrated camera–to–body transform and the SLAM-estimated robot pose. In RViz, blue cubes, shown in Fig. 4(a), mark confirmed objects. Each one is labeled with its class and hit count. Repeated sightings update the same instance, and per-frame near-duplicate boxes are suppressed to the highest-score to keep the map uncluttered. Fig. 5 shows the asynchronous frame rates of the visualization topics. The semantic layer maintains stable instances despite these mismatched stream rates.

IV. CONCLUSION

We built an online, object-level semantic map for a quadruped. Geometry comes from the SLAM Toolbox (2D occupancy with T265 and a 2D LiDAR) and detections from a D435 with YOLO are anchored in the map frame as persistent instances. Simple map-frame association plus short- and long-term memory keeps objects stable through viewpoint changes while avoiding per-frame duplicates. An autonomous navigation system that uses this layer is left for future work.

The limitation we observed is depth misalignment: D435 depth is not consistent with the LiDAR scan plane, which introduces range/pose bias when placing objects in the map. Other limitations are reliance on detector centers, purely geometric association (no appearance cues), and evaluation limited to indoor runs.

V. ACKNOWLEDGMENT

[1]Demo video: online.

REFERENCES

[1] A. Agha, K. Otsu, N. Abate *et al.*, "NeBula: TEAM CoSTAR's Robotic Autonomy Solution that Won Phase II of the DARPA Subterranean Challenge," *Field Robotics*, vol. 2, pp. 1435–1506, 2022. doi:10.55417/fr.2022047.

[2] I. Kostavelis, K. Charalampous, A. Gasteratos, and J. K. Tsotsos, "Robot navigation via spatial and temporal coherent semantic maps," *Engineering Applications of Artificial Intelligence*, vol. 48, pp. 173–187, 2016. doi:10.1016/j.engappai.2015.11.004.

[3] J. Crespo, *et al.*, "Relational Model for Robotic Semantic Navigation in Indoor Environments," *Journal of Intelligent & Robotic Systems*, vol. 86, pp. 617–639, 2017. doi:10.1007/s10846-017-0469-x.

[4] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object Goal Navigation using Goal-Oriented Semantic Exploration," in *NeurIPS*, 2020, pp. 12318–12329.

[5] F. Rollo, G. Raiola, A. Zunino, N. Tsagarakis, and A. Ajoudani, "Artifacts Mapping: Multi-Modal Semantic Mapping for Object Detection and 3D Localization," arXiv:2307.01121, 2023.

[6] Z. Betta, C. T. Recchiuto, and A. Sgorbissa, "People, cracks, stairs, and doors: vision-based semantic mapping with a quadruped robot supporting first responders in Search & Rescue," in *IEEE RO-MAN*, 2024. doi:10.1109/RO-MAN60168.2024.10731303.

[7] N. Yokoyama, *et al.*, "VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation," arXiv:2312.03275, 2023.

[8] F. L. Busch, *et al.*, "One Map to Find Them All: Real-time Open-Vocabulary Mapping for Zero-shot Multi-Object Navigation," arXiv:2409.11764, 2024.

[9] K. Qu, J. Tan, T. Zhang, F. Xia, C. Cadena, and M. Hutter, "IPPON: Common Sense Guided Informative Path Planning for Object Goal Navigation," arXiv:2410.19697, 2024.

[10] A. Koval, J. Kot, P. Saska, and V. Kurfess, "Experimental evaluation of autonomous map-based Spot navigation in confined environments," *Biomimetic Intelligence and Robotics*, vol. 2, 2022, Art. 100035. doi:10.1016/j.birob.2022.100035.

[11] S. Macenski and I. Jambrecic, "SLAM Toolbox: SLAM for the dynamic world," *Journal of Open Source Software*, vol. 6, no. 61, p. 2783, 2021. doi:10.21105/joss.02783.

[12] "Intel® RealSense™ D400 Series Product Datasheet," July 2023. Available: https://www.intelrealsense.com/wp-content/uploads/2023/07/Intel-RealSense-D400-Series-Datasheet-July-2023.pdf

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv:1506.02640, 2016. doi:10.48550/arXiv.1506.02640.