Continuous Dynamic Semantic Mapping With Zero-Shot Vision Encoders

Carson Stark, Surya Sunkari, Tarun Mohan, and Luis Pabon

***Abstract*— This work presents an approach for flexible semantic mapping in dynamic environments which leverages recent advancements in foundation models trained on extensive text and image data. These models enable zero-shot classification, facilitating the registration of objects even if they are not present in narrow training data-sets or predefined dictionaries. A three-step process is introduced, combining the Segment Anything Model (SAM) for image segmentation, OpenAI's Contrastive Language-Image Pre-Training model (CLIP) for filtering background elements, and Salesforce's Bootstrapping Language-Image Pre-training model (BLIP) for labeling unknown objects. The positions of identified objects are stored within a semantically labeled OctoMap, allowing the accumulation of diverse object angles and dynamic map updates to accommodate changing environmental conditions. The system is evaluated in its current state based on the percentage of objects properly segmented and the percentage of segments properly labeled.**

I. Introduction

This semantic mapping system aims to achieve accurate, comprehensive perception of the environment and maximum flexibility by utilizing zero-shot vision encoders such as SAM, CLIP, and BLIP. The computational cost of these models limits their application for real-time semantic mapping purposes on board a robot platform; however, our design makes use of a sparse set of quality snapshots instead of frame by frame processing. The robot autonomously navigates using predefined waypoints, captures image snapshots, and employs the Segment Anything Model to generate object segmentations. CLIP is used to find relevant object segments, while BLIP captions unidentified objects, thereby expanding the system's label dictionary. The object poses are stored in an OctoMap, enabling the aggregation of multiple perspectives and the ability to dynamically update the map when objects change position.

The ability to classify a wide variety of contextually relevant objects is especially important to keep pace with rapid advancements in flexible robot decision making brought about by AI agents driven by Large Language Models [9]. Such agents are capable of intelligently adapting their behaviour to the state of their environment, but they require semantically rich information about their surroundings for this capability to be useful. Our system, when fully developed, is designed to integrate with flexible service robots, enabling them to continuously monitor and update the state of their environment for the purpose of making more informed decisions in the future.

II. Related Works

A method has been devised which seamlessly integrates the Segment Anything Model with Structure from Motion (SfM) techniques [1]. This approach involves segmenting objects within video frames using SAM and projecting these segments onto a 3D video representation created by SfM. By leveraging this integration, the system effectively propagates object segments to future frames. Label propagation and SfM can be used to construct a highly accurate semantic map of an environment, however the computation time for label prorogation prohibits the use of this method for real time-applications. In contrast, we aim to design a lightweight semantic mapping solution that works on board a robot in near real time. Additionally, our system should integrate easily with common robot software such as ROS localization and OctoMap packages, maximizing ease of use.

While object segmentation is critical for this system, it would be useless without a way to semantically differentiate objects. Building on the foundations of autonomous semantic exploration, a cutting-edge strategy has been formulated [6]. This approach optimizes Next Best View algorithms through a frontier-based exploration strategy. The system autonomously explores the environment, generating precise labels for objects of interest. A key advancement lies in the semantic segmentation of objects, essential for accurate grasp pose detection. This strategic fusion of exploration techniques and semantic understanding enhances the system's ability to navigate and autonomously label objects with high accuracy. This gives the system the ability to be optimized for Next Best View algorithms in the future through its robust algorithm for semantic segmentation.

To deal with mapping the environment without a strict Next Best View algorithm, this system draws insights from the Next Best View Planner for 3D Exploration and Inspection [7]. This approach optimizes the OctoMap integration for robotic manipulators, enabling efficient exploration of unknown areas and generating labeled OctoMaps. The integration of a robust OctoMap generator based on movement by the robot, whether automatic or through predefined way points, ensures the systematic exploration of uncharted territories, providing comprehensive insights into the environment.

Additionally, the system benefits significantly from the Semantic SLAM framework, enabling real-time 3D semantic map construction using handheld RGB-D cameras [10]. This open-source project was adapted to achieve the creation of a detailed, voxel-based semantic map. A pivotal capability en-

abled by this integration is the labeling of an OctoMap from a labeled point cloud, allowing for accurate identification of object locations within the mapped environment. Semantic SLAM makes use of a standard object detection model that requires a dataset of known classes. We improve upon the flexibility of this system by employing zero-shot foundation models for object detection, at the cost of a much slower update time.

III. Semantic Mapping Procedure

Due to the computationally expensive nature of the proposed zero-shot object segmentation method, this method is centered around the idea of taking several sparse snapshots at quality poses in the environment instead of performing object detection on each frame of video footage. For each snapshot, a corresponding RGB image, depth image, and camera pose is required. From the RGB image, a labeled image mask is extracted, which is used in conjunction with the depth map to generate a labeled point cloud. This PointCloud is inserted into the OctoMap to register the semantics of the environment in world space.

A. Zero-shot Object Segmentation

CLIP harnesses natural language supervision to learn visual concepts, making it highly adaptable for recognizing a wide array of visual categories [8]. Given the joint embedding space established by CLIP, the similarity between any given image and textual phrase can be estimated. While this is a powerful tool for object classification, it is incapable of object localization and segmentation. Additionally, classification requires a pre-defined dictionary of possible labels to iterate over. Another multi-modal encoder, BLIP, presents a solution to this problem by generating image captions itself [5]; however, it still lacks segmentation abilities. Therefore, this procedure makes use of the Segment Anything Model (SAM) [4]. SAM first extracts all image segments, which are then individually cropped and classified using the capabilities of CLIP and BLIP.

The generation of image segmentations involves the use of the AutomaticMaskGenerator() method from the Python library Mobile SAM [11]. The top half of the image is cropped off, as there are not likely to be relevant objects located above the camera. Segments are filtered by size, IOU, and stability score. Given a collection of image segments, most will contain background elements. CLIP is used to filter these segments by comparing the cropped segment with a generic label (”household item”, ”graspable object”, etc) and a collection of junk labels (”wall”, ”floor”, ”ceiling”). When CLIP is queried, the full prompt is ”a photo of a [label]”. A softmax operation is applied to the scores output by CLIP to normalize the probabilities for each label (1). As a result, the score for the generic item label will be decreased if the junk labels also receive high scores. After the softmax operation, any segment with a high score for the generic label is likely to contain an object of interest.

Fig. 1: Example of parsing a label name from BLIP’s output text. The red text is rejected and the blue text becomes the label name.

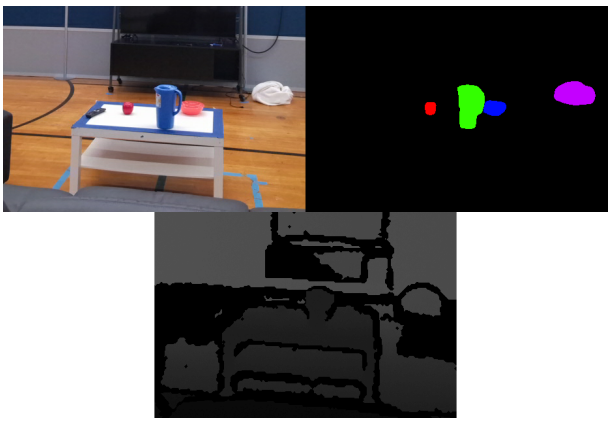

Fig. 2: RGB image and depth map pair with a generated label mask based on object segments.

$$\sigma(y_i) = \left(\frac{e^{y_i}}{\sum_j e^{y_j}} \right) j = 1, ..., n \tag{1}$$

For each object segment, the item may either be known or unknown. CLIP compares the segment to each label in the dictionary of known objects. If the maximum score is below a certain defined threshold, then the object is considered unknown, and BLIP is used to generate a label. BLIP is prompted with the text ”The object in this image is a: ”. To parse a label name, the output is cut off at the words ”on”, ”with”, or ”and”, as illustrated in Fig. 1. This new label is then added to the dictionary. After all the segments have been labeled, a single channel label mask is created by storing the label index of each pixel into the mask. Unlabeled pixels store a value of 0. Fig. 2 shows an example of a label mask and its corresponding depth-color pair. Algorithm 1 outlines the whole process for assigning a label.

B. Semantic OctoMap Generation

An OctoMap is used to create an efficient probabilistic representation of a 3D environment. Each voxel in the map is stored in an Octree data structure, a tree-based spatial partitioning method [2]. This allows for memory-efficient storage and fast search operations within three-dimensional space, as illustrated in Fig. 3. Both free and occupied voxels are stored within an OctoMap. When a new PointCloud is added to the OctoMap, voxels which contain the added points are set to occupied, and ray tracing is used to clear voxels that are no longer occupied. We utilize an OctoMap in our system

Algorithm 1 Zero-shot Object Classification Algorithm

```
for each segment in segments do
    crop ← crop_image(image, segment)
    if is_object(crop) then
        store label and confidence using CLIP
        if confidence is low or label not found then
            get the label using BLIP
            add segment to labeled segments
            if label not in dictionary then
                add label to dictionary
            add label to labels
        else
            add segment to labeled segments
            add label to labels
```

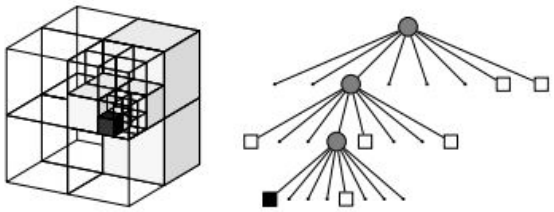

Fig. 3: An octree is a tree data structure that represents three-dimensional space by recursively subdividing it into smaller octants, allowing for spatial partitioning and efficient spatial queries.

because it allows for the map to be dynamically updated. Labeled voxels are expected to be freed if an object changes position.

Once a labeled object mask is provided, the depth map can be reconstructed into a semantically rich PointCloud using the pinhole camera model (2), which maps each pixel in the label mask to a point in 3D space. Each point in the point cloud stores an integer corresponding to a label in the dictionary of objects as well as an xyz coordinate.

$$\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y} \tag{2}$$

Before the PointCloud is inserted into the OctoMap, it needs to be transformed from camera space to world space. This transformation is provided by the ROS tf module. Additionally, the PointCloud is downsampled and points near ground level or further than max distance are filtered out to increase registration accuracy. To correct for errors in the robot's estimated pose, the input PointCloud is fitted to the previous three PointClouds using an iterative closest point algorithm (3). Transformations are constrained to 3 degrees of freedom (rotation about the vertical axis and planar translation). If registration does not converge, the PointCloud is considered to represent an entirely novel view and is added without additional transformation.

$$E(R,t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\mathbf{x}_i - R\mathbf{p}_i - t\|^2 \tag{3}$$

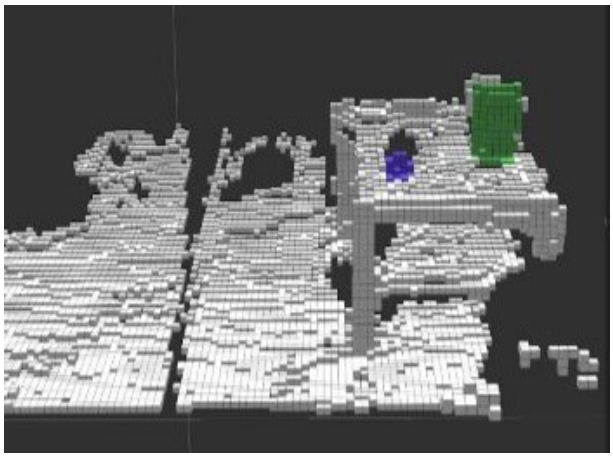

Fig. 4: A semantically labeled octomap generated from a single PointCloud input. The colored voxels represent an apple and pitcher sitting on a table.

When updating the OctoMap with a PointCloud, the label of an unlabeled voxel is set to the most frequent label of the points within it. The proposed object segmentation method frequently misses objects, which would result in labeled voxels immediately losing their semantics in subsequent iterations. To account for this, each voxel also stores a confidence value which is adjusted based on Algorithm 2. This ensures that a voxel must receive conflicting labels multiple times in a row before it gives up its semantics. Fig 4 provides an example of an OctoMap after this algorithm is applied.

Algorithm 2 Voxel Confidence and Label Update

```
s.label ← −1
s.confidence ← 0
max_label ← most frequent label within voxel
if s.label = −1 or s.label = 0 then
    // unlabeled or unknown voxel
    s.label ← max_label
    s.confidence ← 2
else if s.label = max_label then
    // same label
    s.confidence ← min(2, s.confidence + 1)
    s.label ← max_label
else
    // different label
    s.confidence ← s.confidence − 1
    if s.confidence = 0 then
        s.label ← max_label
        s.confidence ← 1
```

After a new point cloud is inserted, adjacent voxels with equivalent labels can be saved as a single object. The centroid and bounding box of each object cluster can be computed using a flood fill algorithm, where neighboring voxels are discovered via a recursive depth-first search.

C. Next Best View

Although our system currently depends on predetermined way-points that the robot can travel to in order to obtain high quality snapshots of the environment, our approach is intended to work in tandem with a next-best-view algorithm, which would allow the robot to autonomously explore and update the semantics of the environment over time. Such an algorithm can improve the effectiveness of our current design with built in features like a preference to capture views of detected objects, unseen angles, and rich components of the environment such as tables, shelves, and counters. Additionally, a timestamp could be stored with each voxel to prioritize updating stale information over a continuous time period.

IV. Evaluation

We evaluated our system in its current state by measuring the zero-shot object segmentation and classification accuracy over a series of snapshots with a variety of different objects. More work must be done before we can meaningfully test the responsiveness of the OctoMap to these detections. Namely, improvements to registration and object clustering are necessary. To set up the experiment, we defined a set of eight snapshot points around the lab space, all oriented toward a table in the center of room. We ran our program on a BWI bot, which uses LIDAR-based SLAM to localize itself and navigate given a previously generated map of the environment [3]. The BWI bot travels to each way-point, taking a snapshot of the environment when it arrives and continuing the process in a loop. The objects detected in each image were recorded and the segmented image was saved to a file for later analysis. For each snapshot, the number of objects missed, number of objects detected, and number of segments correctly labeled was manually recorded.

V. Results

Over three trials in a static room containing 24 everyday objects, the semantic mapping system demonstrated its capabilities to explore space and identify items at increasing levels of difficulty.

A. Object Discovery and Labeling

B. Spatial Mapping Precision

In each of the trials, every object that was identified to be one of the 24 objects that we placed throughout the room was put into the OctoMap in the correct place. However, due to the robots manual localization, there were cases in which certain objects were duplicated in relatively the same position. This resulted in there being more objects found by the system than there physically were in the environment. This issue could be mitigated by scanning for duplicate objects

VI. Conclusion

This work focuses on improving flexibility in semantic mapping for service robots by integrating zero-shot foundation models. It enables the identification of predetermined and unknown objects, incorporating object segmentation to create a detailed semantically labeled OctoMap. Currently, the robot explores predefined waypoints, discovering new objects and updating the environment map. Future work will focus on implementing Next Best View algorithms to optimize the robot's exploration. Additionally, the plan is to synergize this mapping capability with an LLM-based planner to enhance environmental reflection and object recall and to facilitate effective searches for unforeseen elements. Being able to map the robots surroundings, as well as properly identify objects to be manipulated based on the user's request and generate accurate environment states, is critical for a highly adaptable robot platform that can achieve tasks set forth by the user.

References

[1] David Balaban et al. *Propagating Semantic Labels in Video Data*. 2023. arXiv: 2310.00783 [cs.CV].

[2] Armin Hornung et al. "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees". In: *Autonomous Robots* (2013). Software available at https://octomap.github.io. DOI: 10.1007/s10514-012-9321-0. URL: https://octomap.github.io.

[3] Piyush Khandelwal et al. "BWIBots: A platform for bridging the gap between AI and human–robot interaction research". In: *The International Journal of Robotics Research* (2017). URL: http://www.cs.utexas.edu/users/ai-lab?khandelwal:ijrr17.

[4] Alexander Kirillov et al. *Segment Anything*. 2023. arXiv: 2304.02643 [cs.CV].

[5] Junnan Li et al. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. arXiv: 2201.12086 [cs.CV].

[6] Ana Milas, Antun Ivanovic, and Tamara Petrovic. "ASEP: An Autonomous Semantic Exploration Planner With Object Labeling". In: *IEEE Access* 11 (2023), pp. 107169–107183. DOI: 10.1109/ACCESS.2023.3320645.

[7] Menaka Naazare, Francisco Garcia Rosas, and Dirk Schulz. "Online Next-Best-View Planner for 3D-Exploration and Inspection With a Mobile Manipulator Robot". In: *IEEE Robotics and Automation Letters* 7.2 (Apr. 2022), pp. 3779–3786. DOI: 10.1109/lra.2022.3146558. URL: https://doi.org/10.1109%2Flra.2022.3146558.

[8] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[9] Carson Stark et al. *Dobby: A Conversational Service Robot Driven by GPT-4*. 2023. arXiv: 2310.06303 [cs.RO].

[10] Zhang Xuan and Filliat David. *Real-time voxel based 3D semantic mapping with a hand held RGB-D camera*. https://github.com/floatlazer/semantic_slam. 2018.

[11] Chaoning Zhang et al. "Faster Segment Anything: Towards Lightweight SAM for Mobile Applications". In: *arXiv preprint arXiv:2306.14289* (2023).