Building Long-term Spatial Temporal Semantic Map

[1]Ifrah Idrees, [1]Trevor Wiedmann, [1]Huda Abdulrasool, [1]George Konidaris, [2]Nakul Gopalan, [1]Stefanie Tellex
[1] Brown University, USA, [2] Arizona State University, USA

***Abstract*— Building dynamic 3D semantic maps that scale over days and weeks is central for household robots operating in unstructured real-world environments and interacting with humans over long periods. Such a long-term object map can assist human users by grounding their natural language queries and retrieving the object's spatial-temporal information [1]. To our knowledge, there does not exist an integrated approach for building a spatial-temporal map that handles days/weeks of diverse robotic sensor data in a partially observable environment, including dynamic objects. Our approach is agnostic to the object recognition algorithms used and the space of user queries in advance. We propose a representation for the long-term spatial-temporal semantic map that enables the robot to answer real-time queries about the unique object instances in an environment. We also present a Detection-based 3-level Hierarchical Association approach (D3A) that builds our long-term spatial and temporal map. Our representation stores a keyframe that best represents the unique objects and their corresponding spatial-temporal information organized in a key-value database. Our representation allows for open vocabulary queries and even handles queries without specific concepts, such as specific attributes or spatial-temporal relationships. We discuss the retrieval performance of our system with a parameterized synthetic embedding detector. When D3A is queried for 59 ground truth objects, the ground truth object instance is found on average in the 5th return frame, while for baseline, the ground truth object can be found in the 20th frame. We also present preliminary results for a self-collected robotics-lab environment dataset of 22 hours. We show that our queryable semantic scene representation occupies only 0.17% of the total sensory data.**

INTRODUCTION

Building 3D map representations for robotics with unsupervised learning enables enhanced reasoning capabilities for the robot without retraining every time. Home-service robots equipped with 3D semantic maps have great potential to assist human users by retrieving spatial-temporal information about objects from their long-term observations. For example, a person can ask a simple query such as *"What are my top favorite locations to place my keys and watch?"*. Service robots with such an ability will be well-suited to help the home, office users, and the elderly, especially those with dementia.

Many works have been introduced that build a semantic map by performing object detection and segmentation on visual sensor data acquired by a robot [11, 9, 14]. However, this line of research is limited by the closed set of concepts, a fixed set of labels available at training time. Further, these works do not focus on scaling over long periods and handling dynamic environments with partially observable moving objects. Recently, foundational models like CLIP [12], DINO [3] and their variants VLMaps [7], LM-Nav [13], CoWs [5], NLMap-Saycan [4], and ConceptFusion [8] have shown impressive performance on the open set scenarios, where the concepts of interest are supplied only at inference time. These works focus on building a semantic map of static scenes. One key challenge for building a long-term spatial-temporal map using 2D feature representation from pre-trained models such as CLIP or DINO is the multi-view association of partial views of unique objects over time and space and condensing them in a compact queryable scene representation for efficient retrieval during run-time. There are lines of work focusing on robotic object retrieval - handling partial views[1, 2] with point cloud matching, but these do not build a queryable semantic map of dynamic scenes. The above set of approaches when queried, leave the robot searching over countless detections in visual sensor data from many different time slices, which take up a lot of space and time. In addition, these detections contain overlapping partial views of the same object instances; not all of these objects will be relevant to the user query during run-time retrieval.

To our knowledge, there has not been an integrated solution that addresses all of the following challenges: 1)

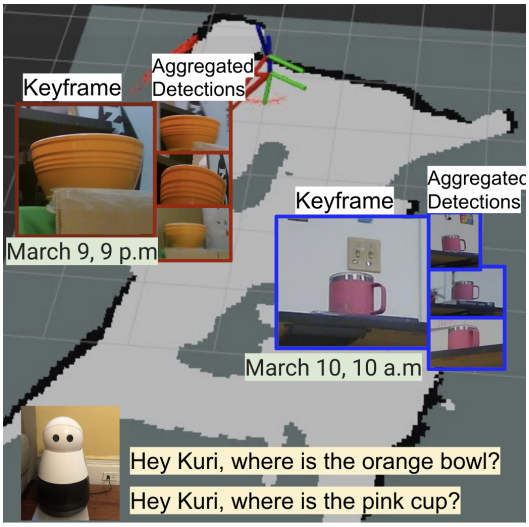

Fig. 1: Sample queries asked from our long-term semantic map, along with their retrieved keyframes, are shown. The retrieved frames and their associated temporal information are 2D projected on the map. Also shown are the partial view detections that are aggregated into the keyframe cluster.

[1]Spatial-temporal information of object referred to the whereabouts of the object such as the location(s) and the time(s) when the object was identified in the physical environment

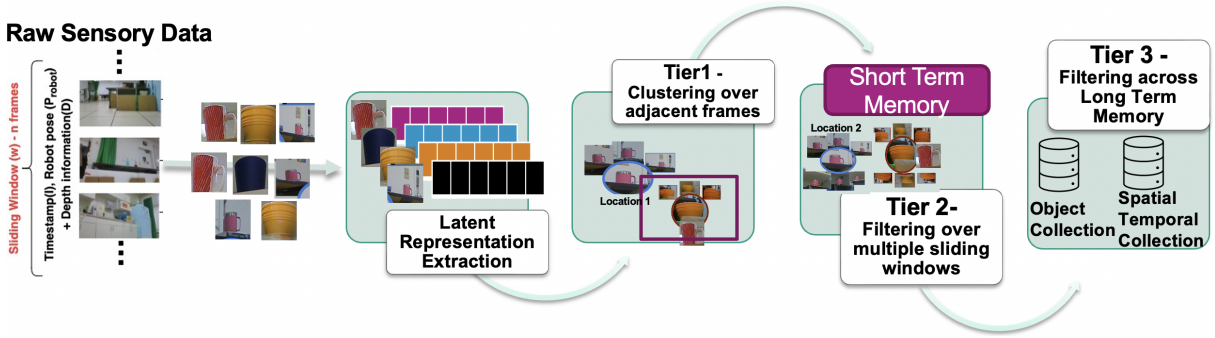

Fig. 2: Visualization of D3A Algorithm

not knowing the objects and the query type in advance, 2) aggregating detections of object instances from different views in a map over days of sensory information, and 3) handling the uncertainty of object poses and the objects going out of the view. To mitigate these challenges, we introduce a new algorithm that extends the concept of keyframe extraction that has been previously used for video frame retrieval to condense multiview detections of unique objects over both physical space and time for long observation periods into a compact and queryable unsupervised spatial, temporal semantic representation. This enables the robot to answer queries about unique object instances more efficiently regarding memory space and query retrieval time.

Our algorithm performs three-tier incremental clustering and filtering of visual observations into unique spatial-temporal instances. While performing association via clustering, our method keeps track of the keyframe that best represents each unique spatial-temporal location of the object. The information about keyframe-centroid clusters is then stored in a spatial-temporally in key-value database, leading to a compact and query-able representation for efficient object retrieval. Hence, making it storage efficient.

We test our generated semantic map's compactness and query retrieval performance on a self-collected environment dataset. Our dataset includes 22 hours of observations collected by the social robot Kuri Mayfield [10] over four days[2]. We show that our queryable semantic scene representation occupies only 0.17% of the total sensory data. We also discuss the retrieval performance of our system with a parameterized synthetic embedding detector. When D3A is queried for 59 ground truth objects, the ground truth object instance is found on average in the 5th return frame, while for baseline, the ground truth object can be found in the 20th frame.

Technical Approach

A robot monitoring the environment over long periods is given a query Q to retrieve spatial temporal information of target objects seen in the environment. The query is an open-vocabulary query asking about the whereabouts of a single object. The robot is initially equipped with the map of the environment, and at every given time step i gathers the following sensory information $s_i =< p_{robot,i}, d_i, f_i >$, where $p_{robot,i} \in P_{robot}$ is the associated robot's pose, and $d_i \in D$ is the corresponding depth information for image frame f_i. The robot collects large amounts of sensor data $S = \{s_1, s_2, ..., s_i\}$ However, the robot has partial observability of the environment, and the information required to answer the object retrieval query is going to be dispersed throughout S in the form of multiple partial view detections. To answer object-retrieval queries efficiently, the robot needs to condense the multi-view detections of the objects in S into a memory-efficient and query-able representation R. A diagram of our algorithm creating R is shown in Fig-2.

For compactness and speed efficiency, D3A reduces the dimensionality of the visual sensory data f_i in $s_i =< p_{robot,i}, d_i, f_i >$, as well as aggregate multiple detections of the same object over long periods. D3A extracts the relevant 2D pixel embedding l_{ij} for every detected object o_j in frame f_i using an visual-language pre-trained model. These embeddings are fused with the object's pose estimates $p_{obj,ij}$ that is calculated from $p_{robot,i}$ and d_i using geometric segmentation. Then, we apply a three-tier online clustering algorithm to aggregate l_{ij}'s. The fused representation for the jth detection in the ith frame l_{ij} is noisy because of the uncertainty in the object's pose estimate $p_{obj,i}$ and noise in the extracted embedding – emb_{ij}. To aggregate the different view detections of the same object and to identify the unique objects in the environment, we consider a sliding window w over sensory data $\{s_{i-w}, ..., s_i\}$. The fused representation l_{ij} of all detections in this sliding window's frames are then clustered to create a set of clusters G, each identifying a unique object instance. The motivation behind this is that the object instances with similar embeddings and positions in space will be associated with the same cluster hence pruning the noisy detections. We then perform an aggregation operation $\bigotimes$ on every cluster $g \in G$ to create a unified feature representation $l'_{ij}$of each of the clusters.

We also need to aggregate the detections of unique instances across the objects' multiple partial views in the past. This requires sharing information across multiple sliding windows via second-tier filtering. To facilitate this, we maintain a fixed-length short-term memory (STM) indexed over object instance ids and aggregate information in it for all instances in STM. G from the current sliding window

[2] A visualization of the initial state of the environment, along with its 2D map and sample of the collected dataset can found at this link: https://github.com/IfrahIdrees/D3A.git

Performance Metric	Naive	D3A
Database Size (MB)	3312	244
Total Processing Time	0m 8s	2m 26s

TABLE I: System Performance (On 3 Hours of Data)

either updates the existing cluster based on similarity with the associated unified representation l'_{ij} or else it will be entered as a new object instance. The keyframe of g is chosen to be the frame with the highest detection probability $prob_{ij}$ among the frames $f'_i s$ associated with the cluster in STM. If a new instance is to be added to the STM when it has reached its maximum capacity, our algorithm evicts the last recently viewed (least-recently viewed entry), lrv_entry, from the STM and moves it to the persistent storage key-value store R. This eviction strategy ensures that objects currently being viewed by the robot remain in the STM for further noise filtering and aggregation of fused representation from partial views.

The filtered entry lrv_entry from the STM could be directly added as the final aggregated cluster into our compact representation R. However, we want to aggregate detections of unique object instances over not just recent times but also long periods of time. To do so, every time a lrv_entry is evicted from STM, our third-tier filtering level looks up and updates the aggregated clusters stored in R.

We organize our aggregated spatial-temporal representation R in a key-value database over two collections. The **Object Identification Collection** (OIc) is updated using the object information of the evicted clusters represented by the lrv_entry, while the other store **Spatial-temporal Collection** (STc) stores the object entries indexed over time and position in physical space. Every object inserted in OIc is associated with a unique identifier $ObjectID$, which indexes STc to get the object detections over space and time. This allows for efficient retrieval during run-time.

Query Processing and Answering: The robot then uses this representation to return a small set of keyframes in response to a given query. This enables a person to find relevant information about the target object quickly. Our representation allows for open vocabulary queries and even handles queries without specific concepts, such as specific attributes or spatial-temporal relationships. We provide evaluation results for the queries with specific attributes provided for a single object of interest in each query. Text embedding for the given query is computed using the corresponding pre-trained CLIP text encoder. Given a query q, and a map with fused features L, we compute a per-cluster score $s \in [-1; 1]$ as the cosine similarity defined as $s = < l, q >$ and rank the clusters of object instances based on the score s and the probability associated with the key frame.

EVALUATION

The aim of our evaluation is to test the hypothesis that a database-backed system with our algorithm D3A improves both 1) the compactness of the spatial-temporal representation of objects in the environment and 2) the object retrieval performance by returning a small subset of keyframes as measured by the mean reciprocal rank, and miss rate as described in Experiment Design subsection. As a result, the user will have to search over just a few returned results to find the answer to their question. To test our hypothesis, we perform a real-world evaluation, using both a range of synthetic detectors built from ground truth data to control the noise level and real detectors deployed on a mobile robot.

Dataset Collection

The robotics lab environment in which our robot Kuri [10] patrolled for four days was uncontrolled and cluttered: people were allowed to use the space as is. The illumination was kept the same throughout the data collection. It has 10,132 image frames over 22 hours and contains static and dynamic objects, such as various cups and bottles. More details of dataset collection and robot localization are in the appendix section - Table II.

Dataset Annotation

For evaluation purposes, the collected dataset was manually annotated by the authors. Every unique instance in the dataset was assigned a unique id and a ground truth location that was used to perform object association over time and space. Due to time and manual labor constraints, we could only annotate 3 hours of data.

Parameter Selection

We use Detectron's object detector Girshick et al. [6] object detector to extract the latent representation. The sliding window length, the normalized threshold for feature matching (cosine similarity), and the size of short-term memory (STM) were set to 10, 0.4, and 400 (our RAM's maximum capacity), respectively. We performed a parameter sweep offline on the collected data and found this setting to be optimal.

Experiment Designs

We conduct two experiments with one baseline.

Baseline: Our baseline is a "Naive" baseline similar to Nlmap-Saycan's scene representation [4] that inserts every detection's spatial-temporal information directly into the database without aggregation. This baseline can still index detections on space and time but ablates the three-tier processing that is integral to our system for aggregating partial views of the objects.

Metric Definitions: We measure the quality of keyframes returned for a query with the MRR@50 evaluation metric, which is the multiplicative inverse of the rank of the correct keyframe in the top 50 returned frames ordered by their similarity score. We want the rank of the correct frame to be as close as the start of the list of returned frames; hence, higher MRR closer to 1 is better. We measure the miss rate (the number of objects missed by the algorithm during the clustering phase due to detection or classification error, hence cannot be found in the returned frames).

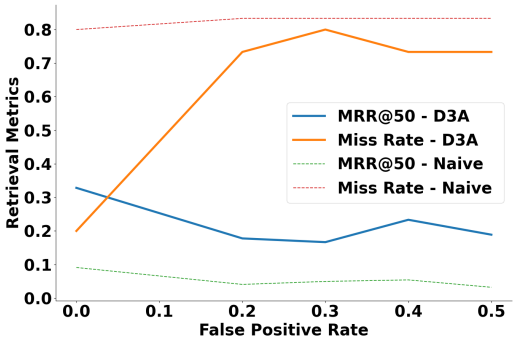

Fig. 3: Exp 1 - Retrieval performance with a synthetic embedding generator with increasing uncertainty

Exp 1 - Query Retrieval Performance with a Synthetic Embedding Generator with Increasing Uncertainty: The performance of D3A depends on the accuracy of the pre-trained visual-language model and object pose estimates. In this experiment, we design a synthetic embedding generator that replaces the object detector. Our synthetic embedding generator is parameterized with a false positive rate (fpr)[3] and false negative rate (fnr)[4]. We keep the fnr low since we assume that the object is not in the environment if the detector fails to detect the object. We annotate ground truth object embeddings as one-hot encodings ohe of the unique instances, where each index represents the unique id of a ground truth object. We then use the fpr and fnr to generate noise in the synthetic embedding by flipping the bits of the ohe. **Results**: We test our system on the query pattern with specific attributes provided for a single object of interest in each query - *Did you ever see an orange bowl?*, for all ground truth object instances. Results are shown in Fig-3. Our method's MRR@50 decreases as fpr increases. Despite this trend, D3A's MRR@50 remains considerably higher than Naive's MRR@50. This indicates that even when the object detector used in D3A is not good, D3A retreieves the ground truth queried object on average in the 5th return frame while for baseline, the ground truth object can be found in the 20th frame.

D3A's miss rate increases as the synthetic detector becomes noisier but is always lower than Naive's miss rate. The slight decreasing trend of D3A's miss rate for $fpr > 0.3$ is because, with a higher probability of flipping the bits of the ohe, the probability of different object instances being assigned a similar embedding also increases. At the time of retrieval for a given object, this causes more keyframes to be returned one of which includes the queried object.

Exp 2 - Compactness Comparison: We measure the cumulative number of cluster insertions in the database per hour over the complete 22 hours for both $D3A$ and the "Naive" baseline. As seen in Fig-4, the number of insertions per hour for $D3A$ is much less than that of the baseline and scales well as the amount of data increases. This demonstrates D3A's success in aggregating partial views. D3A takes two orders

[3] fpr denotes the probability of falsely labeling the detected instance
[4] fnr denotes the detector's probability of not detecting the object

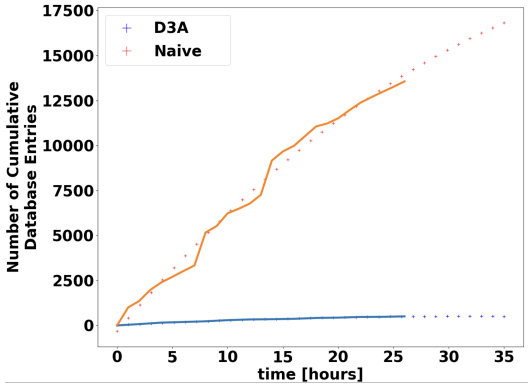

Fig. 4: Exp 2 - Cumulative number of insertions to the database per hour

of magnitude more time to process the raw data than the baseline but outputs a representation **14.7x** more compact and efficient in answering questions, as shown in Table I. This demonstrates that D3A allows for scalable spatial-temporal representation of objects.

Discussion & Future Work

There are certain limitations of our system that we plan to improve. The performance of our system depends on the parameters selected and the pre-trained model used. We notice that increasing the feature matching threshold increases the number of false positives during tier-1 processing. For short-term memory, the greater the buffer size, the more noise there will be in the object's pose and embedding to be filtered and processed. For our method, the more accurate the pre-trained model, the better the cluster aggregation. But even with the noisier model, our model performs better than the Naive baseline.

We plan to run more robust experiments for measuring query retrieval and plan to report the speed efficiency of our system by measuring the mean number of frames and object clusters returned for the queries, the total time taken by a query divided into retrieval time, and the evaluation time to find the correct frame by matching against the ground truth.

Conclusion

We present a novel algorithm for robots to efficiently answer spatial-temporal queries about objects in the environment over long periods. Our algorithm aggregates partial view detections of unique instances to create a compact and query-able representation of the objects. By explicitly performing detection-based three-level associations to identify the keyframes for unique object instances, our algorithm outperforms the baseline in answering queries regarding the mean reciprocal rank, missed rate, and processing time. A robot deployed with our algorithm was able to process 22 hours of sensor data and develop a compact representation of objects in its environment, which is an encouraging step towards enhancing the sensory capabilities of home-service robots that can help users find their lost forgotten objects.

References

[1] Rareş Ambruş, Nils Bore, John Folkesson, and Patric Jensfelt. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In *2014 IEEE/RSJ IROS*, pages 1854–1861. IEEE, 2014.

[2] Nils Bore, Patric Jensfelt, and John Folkesson. Retrieval of arbitrary 3d objects from robot observations. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–8. IEEE, 2015.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[4] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. *arXiv preprint arXiv:2209.09874*, 2022.

[5] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 2022.

[6] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[7] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022.

[8] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

[9] X Li and R Belaroussi. Semi-dense 3d semantic mapping from monocular slam. arxiv. *arXiv preprint arXiv:1611.04144*, 2016.

[10] Robotics Mayfield. Meet kuri! the adorable home robot, 2018. URL https://www.heykuri.com/.

[11] Xianyu Qi, Wei Wang, Mei Yuan, Yuliang Wang, Mingbo Li, Lin Xue, and Yingpin Sun. Building semantic grid maps for domestic robot navigation. *International Journal of Advanced Robotic Systems*, 17 (1):1729881419900066, 2020.

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[13] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning*, pages 492–504. PMLR, 2023.

[14] Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5079–5085. IEEE, 2017.

Duration	22hrs	3hrs
Total Number of Frames	$10,132$	991
Frame Rate (per minute)	7.67	7.67
Sensor Data Size (GB)	10.53	1.43
Total Number of Detections	$13,565$	$2,558$
Ground Truth Objects	N/A	59
		Static = 49, Dynamic = 10
Number of Times	N/A	Mean = 2.1 ± 1.3,
Dynamic Objects move		Max = 5.0

TABLE II: Dataset Details

APPENDIX

DATASET COLLECTION DETAILS

Details of our self-collected dataset can be found in Table-II. The robotics lab area used for data collection and experimentation included the kitchen and general areas with tables and chairs. Our dataset includes both static and dynamic objects. Our method will not be able to differentiate between two objects that exactly look the same and will consider them as one. For data collection, the robot collects observations by doing multiple scans over the scene at different times. During one scan, the robot collects multiple sliding windows and within one sliding window if the object moves, our method based on how good the object detector's visual features will either we will store both the positions or consider one the positions as noise.

ROBOT LOCALIZATION

We use the adaptive (or KLD-sampling) Monte Carlo localization approach (as described by Dieter Fox), which uses a particle filter to localize the robot's pose. Our method depends on the accuracy of the SLAM's output for estimating the robot's location and, inadvertently, object positions. For this work, we assume that we are limited by the SLAM algorithm,, and if SLAM fails, the object's location stored in the map will be noisy. Our proposed method aims to solve the object association and retrieval problem in a 3D world. It can be extended to Object-based SLAM Relocalization, but currently, our method does not support this.