IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 July 2024, accepted 7 August 2024, date of publication 15 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444478

RESEARCH ARTICLE

Semantic-Based Multi-Object Search Optimization in Service Robots Using Probabilistic and Contextual Priors

AKASH CHIKHALIKAR, (Student Member, IEEE), ANKIT A. RAVANKAR, (Member, IEEE), JOSE VICTORIO SALAZAR LUCES, (Member, IEEE), AND YASUHISA HIRATA, (Member, IEEE)
Department of Robotics, Tohoku University, Sendai 980-8577, Japan

Corresponding author: Akash Chikhalikar (a.k.chikhalikar@srd.mech.tohoku.ac.jp)

This work was supported in part by the JST Moonshot Research and Development Program under Grant JPMJMS2034, in part by JSPS KAKENHI under Grant JP21K14115, and in part by JST SPRING under Grant JPMJSP2114.

ABSTRACT In recent years, the demand for service robots capable of executing high-level tasks has grown. In the future, service robots will be expected to perform complex tasks like 'Set table for dinner'. Such high-level tasks require that the robot possess the ability to retrieve multiple objects from the environment. Thus this paper investigates the challenge of locating multiple objects in an environment, termed 'Find my Objects'. In our approach, we present a novel model for extraction of 'Environment-specific' priors from generalized data available in public datasets. We present a novel heuristic specifically designed to optimize Multi-Object search in indoor spaces while considering User Preferences. We also propose a novel Post-task Position Optimization (PTPO) strategy for improved performance in successive tasks. PTPO enables the robot to leverage information gained during a task to improve its inferencing for the next task. Our approach is built on a Semantic Mapping framework that combines semantic object recognition with geometric data to generate a multi-layered map. We fuse the Semantic Map with environment-specific priors in our inferencing strategy. Importantly, our method is agnostic to object detectors, Visual SLAM techniques, and local navigation planners. We demonstrate the 'Find my Objects' task in real-world indoor environments, yielding quantitative results that attest to the effectiveness of our methodology. This strategy can be applied in scenarios where service robots need to locate, grasp, and transport objects, taking into account user preferences.

INDEX TERMS Service robotics, object search, probabilistic inference, semantic map.

I. INTRODUCTION

Service robots for domestic environments are in high demand. They are being deployed for a diverse array of applications including object delivery, patrolling, and cleaning. Additionally, advancements in SLAM (Simultaneous Localization and Mapping), deep learning, and object manipulation have accelerated the development [1] of service robots. Their integration spans a spectrum of use-cases, from automating household chores to specialized long-term care for the elderly [2], [3]. A fundamental skill that all service robots require is the ability to retrieve items effectively. The ability to locate and pick objects is crucial for downstream tasks including but not limited to manipulating items, assisting individuals with disabilities, and environmental maintenance.

Relying solely on exhaustive exploration or random navigation can be energy-intensive and may not meet time requirements. Therefore, it is essential to harness information from diverse sources to refine robotic object search. Semantic SLAM can provide significant support in this regard. Semantic SLAM involves the extraction and integration of semantic understanding with geometric data to produce detailed, knowledge-based maps. These maps not only identify landmarks like furniture and appliances typical in a house setting but also classify areas such as living rooms [4], [5]. Incorporating semantic information can also assist in problems related to long-term localization [6]. In addition, state-of-the-art approaches incorporate beliefs

The associate editor coordinating the review of this manuscript and approving it for publication was Byoung Wook Choi.

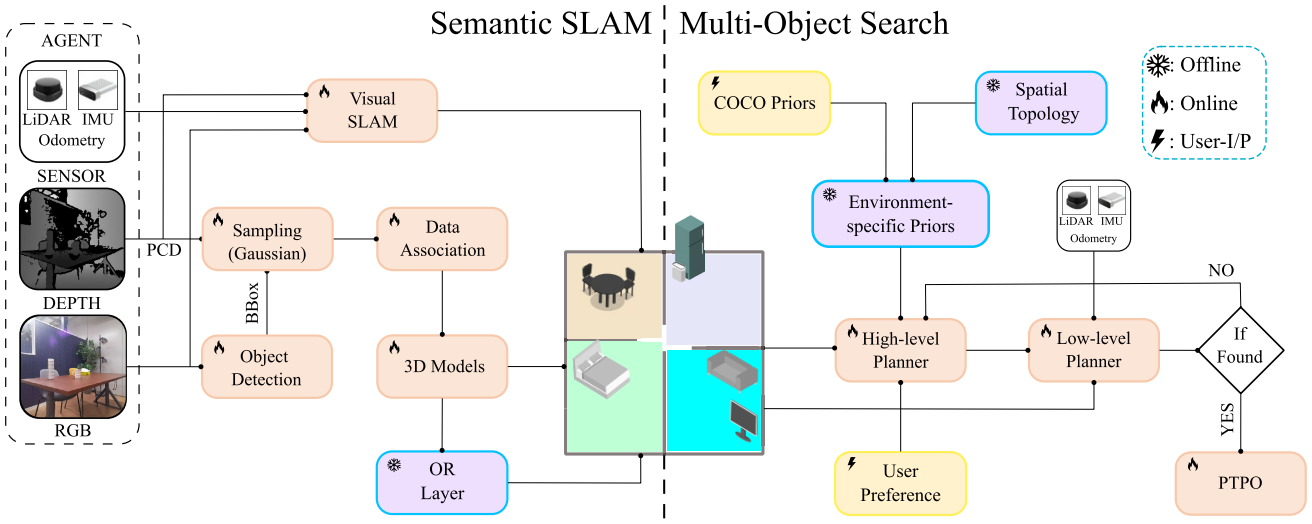


FIGURE 1. Overview of our framework. We implement multi-object search on top of a semantic map. The implementation is split into various modules; some of which are online, some are offline, and some are user-inputs. For a detailed explanation of each module, the interested reader can go through Sec. III-A (Visual SLAM), Sec.III-B (Data Association), Sec. III-C (OR Layer), Sec. IV-A (Environment-specific priors) and Sec. IV-C (PTPO).

about dynamic entities such as other agents or humans [7]. Robots can build informative and contextually rich maps with the help of semantic SLAM techniques, facilitating faster object search.

With the evolution of deep neural networks, extracting semantic information has become more streamlined. Convolutional Neural Networks (CNN) like Yolov7 [8] facilitate real-time object detection, while models like MaskRCNN [9] enable detailed instance segmentation. Service robots can benefit from the combined prowess of semantic SLAM and deep learning to enhance object search capabilities.

Despite significant advancements in robotics, robots today still lack the intuitive *scene-awareness* that humans naturally possess. For instance, while a robot can identify an area as a kitchen and an object as a cup, it should possess the common-sense that 'cups are typically found in the kitchen'. Secondly, it needs to understand negative correlations to identify objects that are 'out of place'. A holistic awareness of what is 'in place' and 'out of place' is necessary for execution of complex tasks like *Prepare bag for work* or *Tidy up my house* [10]. Moreover, user preferences significantly influence what can be deemed as 'in-place' or 'out of place' in the environment. For e.g., a user working from home may prefer their laptop on the couch while someone else may arrange their work setup differently (e.g., in the bedroom). Due to the preferences of an individual, indoor spaces exhibit high variability in layout and clutter. This poses formidable challenges for robot navigation and decision systems. Robots will need to obtain personalized knowledge of the environments they are deployed in to overcome this challenge. Embodied agents with the capability to prepare and leverage knowledge-based maps of indoor environments have a lot of potential for semantic *scene-awareness*. They can use Ontologies to capture common-sense knowledge and spatial associations between objects and places [11], [12]. Blending information from common-sense priors with semantic maps is crucial for an efficient object search.

Object search is necessary for different assistive tasks such as 'Bring cup at the kitchen'. A robot equipped with the ability to search for objects, pick them up and then navigate with semantic awareness will be able to achieve this. Object search can also assist individuals with Dementia since they tend to forget the locations of common items. A robot that can search for objects on demand would considerably improve their life. In future, robots will have to deftly plan and execute the retrieval of multiple items in the environment to accomplish higher-level tasks. Given these considerations, our research aims to address the multi-object search task in indoor settings or the '*Find my Objects*' task. We have designed a framework that can effectively explore semantic maps to look for items commonly found in daily life.

A. CONTRIBUTIONS

The main contributions of this paper include:

- A framework for building Semantic Maps that encodes information about different landmarks.
- A novel strategy to blend environment specific information with generalized knowledge to obtain 'Environment-specific' priors.
- Defining a multi-object search task in indoor settings with region-to-region navigation. We propose a novel inferencing strategy with Post-task Position Optimization for improved performance in multi-object search.
- Quantitative results obtained in real environments and comparisons of our method with baselines and State-of-the-Art for object search.

B. ORGANIZATION

The remainder of the paper is as follows. Section II reviews related work and highlights our contributions.

Section III details our methodology to create a Semantic Map. Section IV discusses the extraction of environment-specific priors, multi-object search algorithm, and post-task position optimization scenario (PTPO) and is the crux of this submission. Section V describes our experiments, results, and analysis of the observed data. Finally, Section VI concludes the paper with a discussion.

II. RELATED WORK

Early attempts at solving the task of Object Search focused on mitigating sensor errors and uncertainties in object detection. A boom in those sectors has led researchers to focus more on the three core elements of object search: common-sense priors, mapping, and inferencing mechanisms. We review related works in these areas which helped formulate our idea.

A. COMMON-SENSE PRIORS

Common-sense priors can be defined as probabilistic relationships between objects that reflect spatial relationships and functional associations in an environment. Priors are important since they leverage information from public datasets which would otherwise require collecting large amounts of data and computation to obtain. Extraction of common-sense priors is a challenging task. Various multi-modal sources can be used for determining the priors. One of the earliest works mined a database from the Flickr website to count co-occurrences of objects and landmarks in the tags of each image [13]. Mining priors with textual data can be done using the Skip-Gram model [14]. In [15], the authors synthesized textual sequences by ranking position relationships in ascending order and used the Skip-Gram model for extracting the common-sense prior. A similar approach involved segregating textual data on the basis of nouns (e.g., beds, tables) and prepositions (e.g., in, on) for creating a hierarchical knowledge base [16], [17]. Besides generalized sources, the environment in which the task will be carried out can be used for extraction of priors as well. The authors in [18] collected temporal data as the robot navigated through the environment to build a prior belief based on time stamps of observations. Additional sources such as human instructions or large language models (LLMs) can also provide common-sense priors. Unlike these methods, we present a novel framework to leverage environment-specific information for extraction of priors from public datasets.

B. EXPLORATION-BASED SEARCH

Frontiers, regions on the boundary between open and unexplored spaces, were used as the basis of exploration in this seminal paper [19]. Frontier-based exploration ensures coverage of the entire space and is also highly scalable in terms of area to explore. Since then, exploration has developed tremendous interest as an academic challenge as well as end-application utility [20]. This has resulted in Embodied Agents being challenged with navigating toward objects in an unseen environment (Object Navigation) while simultaneously exploring it [21]. A photo-realistic simulator, called Habitat, was designed for this challenge [22]. A state-of-the-art solution to the Habitat challenge considered distance prediction using semantically related objects as cues [23]. Using semantic concepts for exploration is presented in [24]. Exploration is also conducted using drones by evaluating the semantic and geometric gain for all frontiers [25].

C. SEMANTIC SLAM

Semantic SLAM is an output of detection, tracking, and data association models combined with conventional SLAM techniques. The interested reader can refer to different data association models as a primer for Semantic SLAM frameworks [26], [27], [28]. Early approaches handled localization of semantic and geometric data independently and then incrementally built the semantic map [29]. A more recent method [30] used semantic data as landmarks to improve the efficiency of conventional SLAM techniques. Extra layers related to place categorization can be added using empirical data [31] to prepare multi-layered maps.

D. TASK-ORIENTED INFERENCE

Previously, researchers have tried to solve the object search/retrieval problem in three ways. One approach is to formulate object search as a Partially Observable Markov Decision Process (POMDP) and optimize the semantic gain from sensor observations at every time step. The Informative Path Planning (IPP) approach aims to develop an informative costmap corresponding to the metric map to supplement object search. Lastly, the Next Best View (NBV) methods consider object search to be a discrete problem and use different strategies to determine the best locations for search.

POMDPs are a generalization of Markov Decision Processes by including uncertainties in the observation along with state transition uncertainty. These have been popular among researchers trying to push the state-of-the-art in this domain. However, POMDPs suffer from intractability when solving for large domains [32]. To tackle the computational burden, researchers have introduced hierarchies in spatial scales or planning [33], [34], [35].

In the IPP method, path planning is driven by a joint cost function consisting of information gained and distance travelled along the path. The work in [36] first uses Gaussian Mixture Models and Bayesian relationships to prepare an information map. A sampling-based informative path plan is generated using this map for object search. IPP can also benefit from Reinforcement Learning, as explored in [37].

NBV methods, as described earlier, depend on determining the best locations for object search. In [38], the authors evaluate all routes to find the object and then store information about different objects seen during navigation for quicker retrieval in the next task. The authors in [39] determine a set

of candidate viewpoints and evaluate the viewpoints based on probabilistic belief around the viewpoint.

Our approach can be broadly categorized as an NBV method. In our approach we present a novel framework for extraction of environment-specific priors. Unlike other Object Search approaches, we consider multiple objects while planning and optimize our trajectory accordingly. This provides better results as compared to splitting the multi-object search into sequential single-object searches. We also introduce a novel PTPO framework to improve performance in successive task execution. Furthermore, we provide the robot the capability to prioritize objects based on user preferences.

III. SEMANTIC MAPPING FRAMEWORK

This section presents our semantic mapping framework, which builds upon our previously introduced framework [40] by adding additional layers to the semantic map and increasing the number of object classes. We begin with a brief description of the pre-processing steps followed by a detailed description of the second layer of our semantic map. This layer involves the integration of conventional SLAM processes with a filtering and tracking module, enabling the overlay of the grid map with object information. Finally, we compute an 'observation layer' based on the obtained object information. Similarly, several such layers can be added with new information in our framework (e.g., map topology or region segmentation) [41].

Our framework is built upon the open-source middleware Robot Operating System (ROS) [42] and its associated libraries. We use the 'Grid Map Library' [43] to build, visualize, and maintain our triple-layered map in conjunction with ROS. Additionally, we utilized the Azure Kinect DK sensor SDK [44] for synchronization and rectification of RGB and Depth images.

A. VISUAL SLAM

The RGB and depth images from the Azure Kinect sensor are registered to prepare the semantic map. The depth image is registered in the RGB frame since the RGB image has a lower FoV (Field of View) and subsequently, the compressed depth and RGB images are transported from the onboard computing system to the server computer. These images are rectified using intrinsic camera calibration parameters provided by the SDK, and finally, the point cloud data is generated. The resulting point cloud data is used to create a 3D grid map (octomap) of the environment. While any kind of 3D map representation like RGBDSLAMv2 [45] can be used for navigation, we choose RTAB-Map [46] as the preferred method because of its appearance-based loop closure in real-time.

B. DATA ASSOCIATION

Five of the most ubiquitous object classes in indoor spaces, namely the *chair, bed, table, TV*, and *sofa*, are taken into consideration for generating the semantic map. During the mapping process, YOLOv7 [8] is employed to detect objects in frames. The YOLOv7 network generates a 5-D output for each detected object, including the object class and four bounding box parameters (center coordinates: C_x and C_y, breadth: B, and length: L). The bounding boxes are randomly sampled, and the mean X and Y coordinates are computed from the corresponding point cloud data. Multiple measurements over time are fused using Kalman filtering before placing 3D models on the map frame.

After an object is placed on the map, it needs to be tracked to determine if it is a previously detected instance of that object or not. For tracking the objects in these scenarios, we maintain the record of $'K'$ previously seen objects of a class and current observations in the latest frame ($'L'$ objects of the same class) as follows:

$$\mathbf{P} = \{P_0, P_1, \cdots, P_K\} \tag{1}$$

$$\mathbf{C} = \{C_0, C_1, \cdots, C_L\} \tag{2}$$

We then calculate a cost-association matrix $\mathbf{D}_{k,l}$ between both the sets using Euclidean distance as shown in equation 3 below.

$$\forall (k, l) \in (K, L) : \ \mathbf{D}_{k,l} = \sqrt{(\mathbf{P}_k - \mathbf{C}_l)^T (\mathbf{P}_k - \mathbf{C}_l)} \tag{3}$$

After the cost-association matrix is computed, the association between new observations (i.e., set $\mathbf{C}$) and previous observations (i.e., set $\mathbf{P}$) is determined using the Hungarian algorithm [47] as prescribed in [4].

Association $P \leftrightarrow C$ such that:

$$\min \sum_k \sum_l \mathbf{D}_{kl} \mathbf{X}_{kl} \tag{4}$$

where, $\mathbf{X}_{kl} = \begin{cases} 1, & \text{if 'k' is assigned to 'l'}, \\ 0, & \text{otherwise} \end{cases}$

The determined association ($P_k \leftrightarrow C_l$) is also distance dependent; meaning that if the association distance is less than the threshold value α ($\mathbf{D}_{k,l} < \alpha$), the association is considered to be valid else the observation is appended to set $\mathbf{P}$ as a new instance.

If the object is determined as previously seen, a Kalman filter is used to combine current and prior observations of the same instance over time. For a prior state of the instance *'k'* given by $\mathbf{P}_k^{t-1}$ and an associated new observation $\mathbf{C}_l^t$, determined via the Hungarian algorithm, the following computations are carried out:

$$\text{Prediction Step} : \mathbf{P}_k'^t = \mathbf{S}\mathbf{P}_k^{t-1}, \tag{5}$$

$$\text{Correction Step} : \mathbf{P}_k^t = \mathbf{P}_k'^t + \mathbf{K}(t)(\mathbf{C}_l^t - \mathbf{Z}\mathbf{P}_k'^t.) \tag{6}$$

where, $\mathbf{S}$ is the 2×2 state transition matrix (set to identity), $\mathbf{Z}$ is the measurement matrix (set to identity), and $\mathbf{K}(t)$ is the Kalman gain. The optimal Kalman gain, $\mathbf{K}(t)$, is computed with a diagonal process noise covariance matrix of $0.3\mathbf{I}$ and a diagonal measurement noise covariance matrix of $0.5\mathbf{I}$. We summarize this semantic mapping framework in the Algorithm 1 below.

FIGURE 2. Robot navigation in the semantic map. First Layer: metric grid map with obstacle information as occupied cells. Second layer: landmarks shown by CAD models. Third Layer: observable regions shown by red grids.

Algorithm 1 Algorithm for Semantic Mapping

Require: $Odom$, PCD, $Scan$, $Prev(P)$

$Input: Curr(C), Transforms_{Cam \rightarrow Map}$
while $Not\_Shutdown$ **do**
  $Map \leftarrow Update\ (Map_{t-1}, Odom, Scan)$
  if $CurrObs$
    $C' \leftarrow Sampling(PCD, BBox)$
    $C \leftarrow Transforms_{Cam \rightarrow Map}[C']$
  endif
  $D_{k,l} \leftarrow Dist(P, C)$ ▷ $\forall (k, l) \in (Prev, Curr)$
  $[P \leftrightarrow C] \leftarrow Hungarian\ (D_{k,l}, X_{k,l})$
  if $[P \leftrightarrow C]$ *is valid*
    $P'_{t+1,k} \leftarrow Prediction(S, P_{t,k})$
    $P_{t+1,k} \leftarrow Correction(P'_{t+1,k}, Z, C_{t+1,l}, P_{t,k})$
  endif
end while

C. OBSERVABLE REGION LAYER

After completion of the mapping phase, we determine the observation regions. An observable region in our context is defined as an area around the landmark such as a chair, from where it is sufficiently visible. We define a rectangular region based on the location of the landmark.

The Observable Region layer (refer Fig. 2 below) enables the robot to perform region-to-region navigation. When the robot explores the chair space to search for an object, it navigates to the associated region. Once the robot is within the region, it reorients itself to search for objects on the chair.

The purpose for defining *observable regions* is manifold. First, it reduces the time required and the cumulative distance travelled by the robot to find the object. Secondly, it provides robustness to occlusions of the navigation goal. This is achieved by adding a 'Recovery' behavior to the navigation stack. In the event of an occlusion, the local planner fails which triggers this behavior. Then the closest accessible point within the observable region is sent as the new goal to the low-level controller. Since the robot's final pose is facing the desired landmark, it can still search for the object without reaching the exact navigation goal. If the entire region is occluded, the recovery behavior is retriggered, and the robot will move to the next best location. Lastly, it eliminates the need to undertake oscillatory movements or recovery manoeuvres due to erroneous planning or overshoots during execution. The results necessary to quantify the impact of observable regions are shown in Table 1 (Sec. V-C) below. We use this Semantic Map in our multi-object search strategy, which is described in the following section.

IV. MULTI-OBJECT SEARCH

We focus on the multi-object search strategy in this section. This strategy is implemented on top of the Semantic Map. In our approach, we first extract a quantitative relationship between the objects and the landmark locations as environment-specific priors. We then propose a novel heuristic to incorporate user preferences in the search task. Finally, we explain our PTPO scenario with results.

A. ENVIRONMENT-SPECIFIC PRIORS

Priors help determine the probability of finding an object near a known associated landmark. We use Ontologies to obtain the prior relationship between different objects and

landmarks. Ontologies are a formal representation of concepts within a domain along with the relationships between them. Considering landmarks and objects as *concepts* within the domain (indoor environment), we consider three types of relationships between the *concepts* that the Ontology may describe:

- isOn Ontology
- isNear Ontology
- isIn Ontology

isOn ONTOLOGY

Ontology of such type is *object T isOn landmark L* (e.g., cup is on the table). We interpret the ontology *isOn (object-T, landmark-L)* as the probability of finding object T on landmark L (i.e., $P_{On}(T|L)$). We calculate this probability from the well-known and publicly available dataset 'Common Objects in Context (COCO)' [48]. We first filtered the dataset to exclude images of outdoor areas/spaces due to our focus on indoor environments. Next, we use the instance-level masks provided by the dataset. If the intersection of the object mask and landmark mask is greater than a threshold value (θ_m) and the pixel distance between their centroid is less than a threshold value (θ_d), we consider the object is on the landmark.

isNear ONTOLOGY

Ontology of such type is *object T isNear landmark L* (e.g., Remote is near TV). In our scenario, we interpret the ontology *isNear (object-T, landmark-L)* as the probability of finding object T given landmark L (i.e., $P_{near}(T|L)$), in the same image frame. We use additive smoothing to calculate the associated probabilistic priors using the COCO dataset as given below:

$$P_{Near}(T|L) = \frac{N(T \cap L) + \alpha}{N(T) + \alpha d} \tag{7}$$

where, $N(.)$ is the count of observations in the COCO dataset, d is the number of classes in the dataset and α is a smoothing parameter to account for unobserved object-landmark pairs. We set $\alpha = 0.5$ according to Lidstone's law. The priors obtained using the isNear Ontology are shown in Fig. 3.

isIN ONTOLOGY

Ontology of such type includes, for e.g., milk is in the fridge. We consider that to be out of scope for our study since our robot does not possess the capability to open fridges: a problem still considered non-trivial by the community.

However, planning based on COCO priors as it is would not be accurate. The COCO dataset can provide a generalized relationship between objects and landmarks but fine-tuning is necessary to adjust the priors according to the specific environment in which the robot is operating. This is because indoor environments tend to be unorganized and distinct from each other to some extent and the generalized priors fail to capture that knowledge. For example, if the robot is present in an environment where there is a table close to the

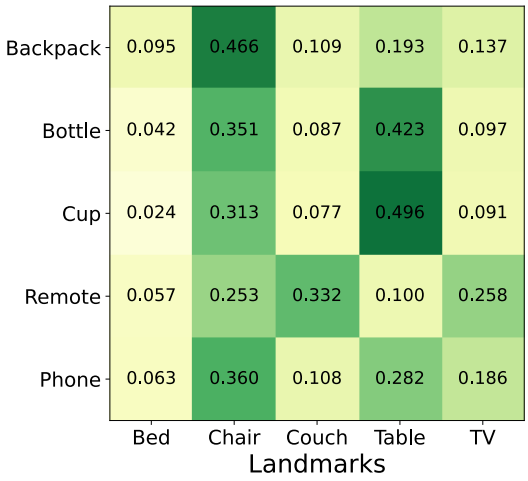


FIGURE 3. Priors: *Near (object-T, landmark-L)*.

TV, the priors for finding a remote at the table or the TV should scale appropriately. Essentially, the spatial proximity between different landmarks should influence the priors for those landmarks. It is also possible that multiple instances of the same landmark class may be present in the environment. In such cases, the priors are scaled according to the instance count. We calculate the Euclidean Distance between different landmarks $d(L, L')$, and propose a Spatial Decay Function (SDF) for scaling the priors.

$$d(L, L') = \sqrt{(L_x - L'_x)^2 + (L_y - L'_y)^2} \tag{8}$$

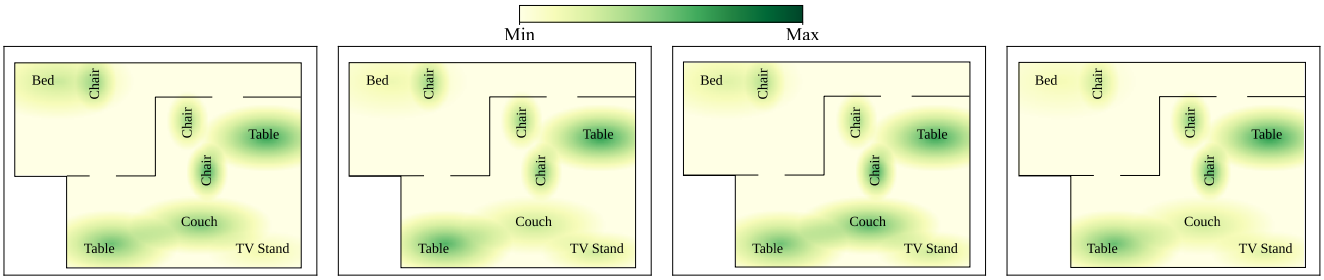
$$SDF(L, L') = \begin{cases} 1 & \text{if } 0 \le d \le d_{min} \\ 1 - \left(\dfrac{d - d_{min}}{d_{max} - d_{min}}\right)^2 & \text{if } d_{min} < d \le d_{max} \\ 0 & \text{if } d_{max} < d \end{cases} \tag{9}$$

where, $d_{min} = 1$m and $d_{max} = 3$m considering our experimental space. We use the SDF to fuse the probabilistic priors obtained from the 'isOn' and 'isNear' Ontologies to obtain the environment-specific priors. For an object T conditioned on landmark L is:

$$P_{env}(T|L) = \eta \left(P_{On}(T|L) + \lambda \sum_{\forall L' \neq L} \Psi(L, L') \right) \tag{10}$$

$$\Psi(L, L') = P_{Near}(T|L') \cdot SDF(L, L') \tag{11}$$

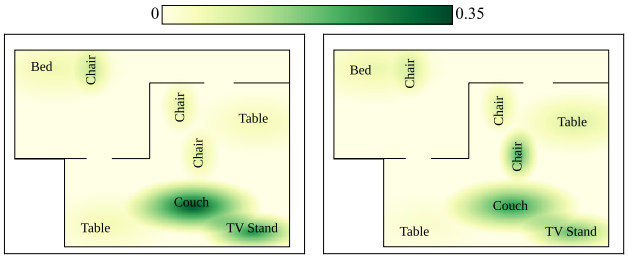
where, η is a normalization constant and $\lambda(= 0.5)$ is the hyperparameter for recombination. The value of λ must be less than 1 since the 'isOn' prior is more relevant for Object Search as compared to 'isNear' prior. Further, an exact value of λ=0.5 was selected to optimize the influence of the 'isNear' ontology on the environment-specific priors. Since the location of landmark 'L' is known, we use the environment-specific priors to prune the search space for the object.



(a) Prior: Backpack (b) Prior: Bottle (c) Prior: Phone (d) Prior: Cup

FIGURE 4. Heatmaps for different objects specific to the environment. A fusion of empirical data from the COCO dataset with environment-specific information encoded by the SDF (eq. 10) is used to obtain heatmaps for all objects.

Fig. 5 below shows the heatmaps for object 'Remote' before and after the fine-tuning of the generalized priors. We see in Fig. 5b that the 'heat value' of the chairs and table has increased due to their proximity to the TV stand. The heatmaps were populated based on probabilistic relationships with an elliptical decay from the center of landmarks. The probability of an object on a grid not populated by a landmark instance (i.e., on plain ground) is negligible but never zero. Heatmaps for other objects are shown in Fig. 4.



(a) Remote (Generalized) (b) Remote (Fine-tuned)

FIGURE 5. Fine-tuning of generalized priors obtained from the COCO dataset.

B. HEURISTIC SEARCH

Heuristics are essential for evaluating the cost of visiting a landmark to search for objects. They should minimize the energy spent in searching while maximizing the chances of finding the object. Furthermore, in our case, the robot needs to plan for multiple objects while incorporating user preferences, in cluttered indoor environments. The robot needs to consider these adversarial conditions while planning. Towards that end, this paper proposes a novel heuristic for a multi-object search that includes user preference:

$$\mathbf{H} = \frac{Dist(A, L)}{\alpha P_{env}(T_1|L) + \beta P_{env}(T_2|L)} \tag{12}$$

where, **H**: Cost of visiting the landmark
$Dist(A, L)$: Dist. to landmark L from the robot (A)
$P_{env}(T_i|L)$: Prior for object T_i at landmark L

The hyperparameters α and β are used to incorporate user preferences. If the user prioritizes object 1, then $\alpha > \beta$, and vice versa. The distance to visit the location L is obtained from the A* global planner. A greedy search is initiated according to this heuristic. When one of the objects is found, the heuristic drops down to its single-object equivalent. The search ends when both objects are found.

Since the robot will visit high likelihood landmarks first, the influence of the priors (P_{env}) is noteworthy. A generalized prior may be erroneous in some environments, due to which the performance will deteriorate. However, our prior (i.e., P_{env}) adapts to the environment topology. Scenarios where the prior is erroneous despite considering the environment topology, large scale data, and user preferences are thus, highly unlikely.

The user preferences also change the trajectory taken by the robot while searching for the objects. We show this qualitatively in Fig. 6. The robot is tasked with searching for a Cup and Remote in all three scenarios (Fig. 6a-6c). The robot follows the sequence, Table → Sofa → TV when searching for Cup is the priority (Fig. 6a). For an equal priority search, the robot follows Sofa → Table → TV sequence (Fig. 6b). If the user prioritizes Remote, then the sequence of locations visited is Sofa → TV → Table (Fig. 6c). An overview of multi-object search strategy can be found in Algorithm 2 below.

Algorithm 2 Algorithm for Multi-Object Search

Require: *Planner*, *landmarks* (L) ▷ Sofa, Table, etc.
Input : *Priors* (P_{env}), *Objects* (T), *Pref*
ObservationSpace
$(\alpha, \beta) \leftarrow f(Objects, Pref)$
while *Not_Found* **do**
  For $L \in \mathbf{L}$
    $D \leftarrow Planner(A, L)$
    $H_{list} \leftarrow \frac{D}{g(\alpha, \beta, Objects)}$ ▷ Ref. Eq. 12
  end For
  $NavGoal \leftarrow Max(H_{list})$ ▷ $NavGoal \in L$
  if *Robot* in *OSpace* **then**
    $Robot \leftarrow Reorient(Robot, NavGoal)$
  end if
  $Not\_Found \leftarrow Update(NavGoal, Detections)$
end while

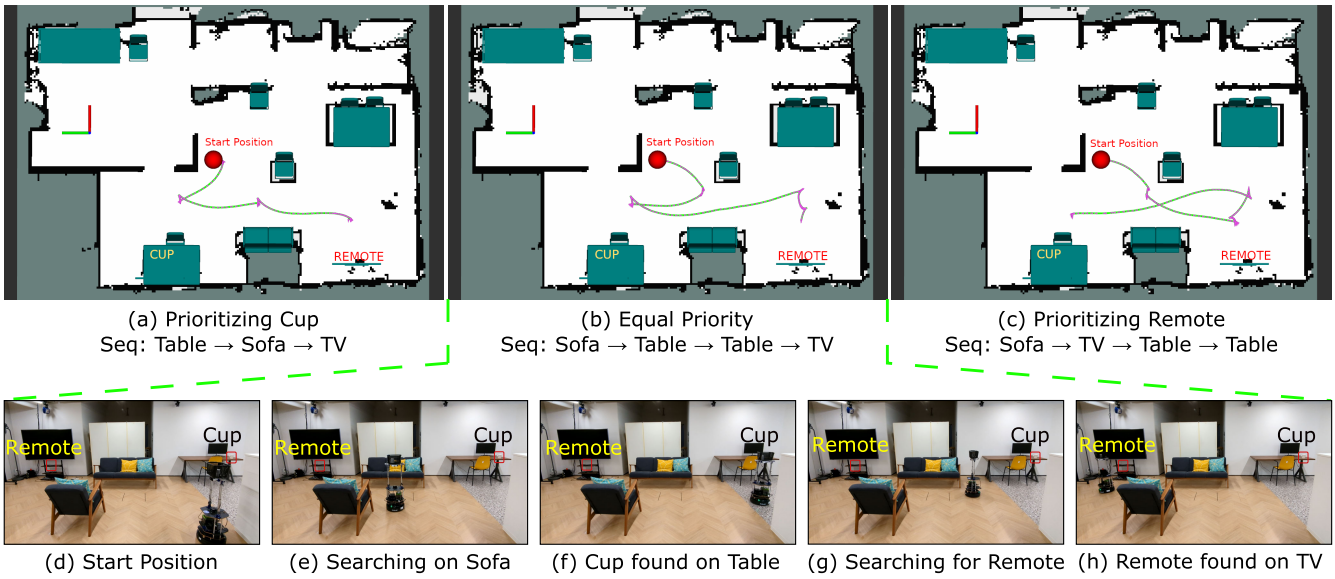


FIGURE 6. Trajectory followed with respect to the user priority. The next goal position is determined based on the object prioritized and the proximity of landmarks to the robot. Different trajectories {(a)-(c)} are the outcome of different priorities set by the user. Figures {(d)-(h)} show snaps of the trajectory followed when the robot (Turtlebot in picture) searches for Cup and Remote with an equal priority for both. The robot finds the Cup on the table and the Remote on the TV.

C. POST-TASK POSITION OPTIMIZATION

For previous works, the only task completion criterion is that of successfully finding the object. However, to develop long-term capabilities, the robot should execute consecutive search tasks with improved efficiency. Thus, we dive one stage further and propose a novel Post-task Position Optimization. Post-task Position Optimization leverages the updated understanding obtained from the current task, to autonomously navigate to optimal positions instead of default locations (for e.g., docking station). For the sake of understanding, consider a scenario where the robot is tasked with searching for a Phone and Bottle. During the search, the robot may encounter other objects (e.g., Laptop) in the environment. After completion of the task, the robot may choose to ignore the object detected, go back to its docking station, and wait for the next request. The docking station could be far away from its current location. Instead of returning, the robot can utilize the information gained during the task as well as conserve its battery by navigating to any of the pre-determined positions. The robot confirms its battery level before navigation to avoid running out of battery. The positions are determined so as to not hinder the movement of other robots or humans in the environment.

For this research, the rooms are considered as known however one can use Voronoi structures for performing geometric segmentation of the map [49] to obtain the room-related information. The robot considers the following criteria for choosing optimal post-task positions.

- Proximity to landmarks with a higher prior likelihood of finding objects to improve performance in subsequent tasks.
- Proximity to itself for minimizing energy and time spent in navigating to the post-task position.
- Minimizing room transitions while navigating to the position to not interfere with other robots or humans in the environment.

This optimization drastically improves the search times of the robot when it is repeatedly tasked with searching for objects. If the robot is present in room R, the robot evaluates each '*Position*: Q' present in rooms R' of the environment as follows:

$$\arg\max_{Q} U(Q) = \sum_{\forall R'} \sum_{\forall L \in R'} \frac{\sum_{\forall T} P_{env}(T|L)}{Dist(A, Q)} 2^{-\delta(R,R')} \quad (13)$$

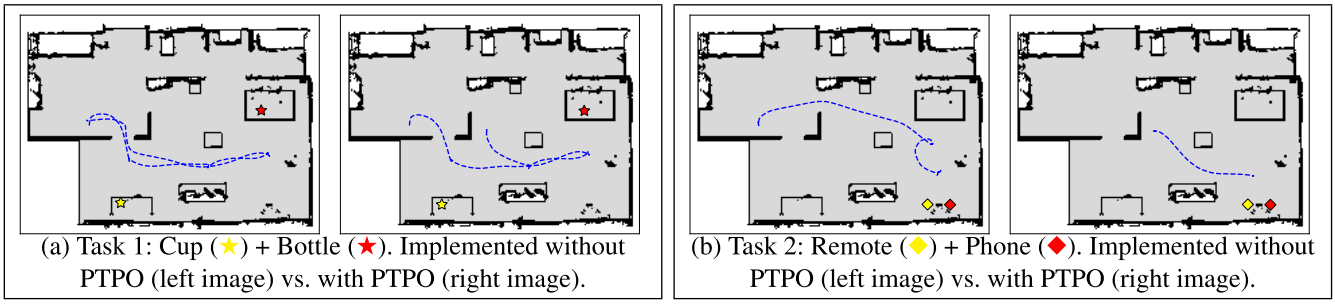
where, $\delta(.)$ is the Kronecker-Delta Function
U: Utility of Position (Q)
Dist(A,Q): Distance to Position (Q) from robot (A)

The above equation guides the robot to navigate to positions closer to high-likelihood landmarks while penalizing incremental travel to reach there. Additionally, a scaling factor has been introduced to discourage room transitions during the optimization.

Fig. 7 details the qualitative difference due to PTPO in successive tasks. The robot is initially tasked with searching for the Cup and Bottle. After the robot completes the task, a follow-up request for searching the Remote and Phone is given to the robot. The overall distance travelled for the first task, i.e., Cup + Bottle, is less when using the PTPO module (Fig. 7a vs 7b). Since the robot utilizes the information gained during the first task, the second task is also accomplished quickly. The quantitative impact of PTPO can be found in the Ablation Study (Sec. V-C).

V. EXPERIMENTAL STUDIES

We conducted numerous experiments to understand the influence of initial conditions, our proposed heuristic, and



(a) Task 1: Cup (★) + Bottle (★). Implemented without PTPO (left image) vs. with PTPO (right image).

(b) Task 2: Remote (◆) + Phone (◆). Implemented without PTPO (left image) vs. with PTPO (right image).

FIGURE 7. Trajectory followed with and without Post-task Position Optimization (PTPO). PTPO results in reduced time, distance, and room transitions during successive object search tasks. Quantitative results are given in Table 1 of Sec. V-C.

navigation strategies. Each data point shown in the next subsection is obtained after averaging the results of five tests conducted in every scenario. Averaging eliminates any bias due to the slight randomness of the path planners and the minor differences in the starting positions (< 5cm) in each run. The following section describes the experimental details, starting with hardware configuration, followed by an ablation study, and finally comparative results and analysis.

A. HARDWARE CONFIGURATION

We use a Turtlebot2 platform with Kobuki base for our experiments. The onboard sensors include an RGB-D camera (Azure Kinect) and a laser range scanner (RPLIDAR S2). The encoder information from the robot base is used to compute the odometry. The data acquisition from sensors and command relay to the robot is performed on the NVIDIA Jetson AGX Xavier as the client. The backend computations as well as the frontend visualization were carried out on a server CPU with NVIDIA 3090RTX graphics unit with an i9-12900K processor. The ROS distributed computing network ensured time-synchronized communication between the server and the client.

B. LIVING LAB- SIMULATED INDOOR ENVIRONMENT

All tests were performed in a simulated indoor environment called the 'Aobayama Living Lab' [2] at Tohoku University. The goal is to create a concept for future welfare facilities, as shown in Fig. 8. The Living Lab included household objects such as tables, chairs, sofas, beds, TVs, lamps, and cabinets. The facility emulates various areas, including toilets, bathrooms, and kitchens as well as an outdoor environment with stairs, slopes, and rough terrain. The dataset generated from the Living Lab will be used to facilitate long-term navigation for service robots.

C. ABLATION STUDY

We conduct an ablation study to compare the relative impact of different modules of our framework described till now. Specifically, we quantify the impact of Post-task Position Optimization (PTPO) and Observable Regions (OR) in our Object Search task. We removed from our overall framework, the PTPO and OR modules and treated it as

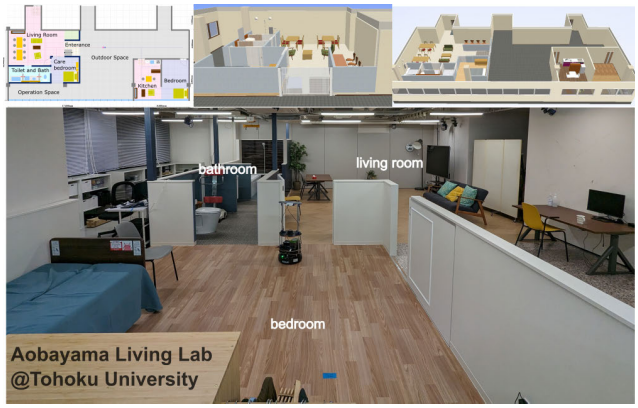


FIGURE 8. Aobayama Living lab (Tohoku University): Indoor test-bed environment for experimenting with robots.

the Baseline. The data with the Baseline framework was compared to adding PTPO and OR to the Baseline. For this study, the robot performed two tasks consecutively. Each task involved searching for two objects from the same start location. One set will be defined as two such tasks. For e.g., as shown in Fig. 7, Task 1 (Cup + Bottle) and Task 2 (Remote+Phone) would comprise one set. We used cumulative Time to search (Time(s)), Distance to search (Dist.(m)), and Room Transitions (RTs) during a set, as the parameters for evaluation. RT is defined as the number of times the robot moved from one room to the other during the task. The RT parameter was manually determined by observing the trajectories followed by the robot. However, this can be automated by amending the state vector of the robot to indicate the current room it is present in. With the RTs determined, the average results of the experiments are given in Table 1 below.

TABLE 1. Ablation study with PTPO and OR.

Modules			Metrics (Avg.)		
Baseline	PTPO	OR	Time(s) ↓	Dist.(m) ↓	RTs ↓
✓	×	×	105.07	21.09	7
✓	×	✓	91.62	19.92	7
✓	✓	×	78.67	13.01	**4**
✓	✓	✓	**70.31**	**12.45**	**4**

We can see from the Ablation Study that the best results are observed when the OR and PTPO modules are used in tandem. The cumulative effect is a reduction in the overall time by 33.08% and overall distance by 40.96% compared to the Baseline. Incorporating just the OR module has a minor impact on the overall distance, however, a considerable time reduction (12.80%) compared to the Baseline is found. This supports our hypothesis that ORs help eliminate the need for cumbersome manoeuvres due to erroneous planning or overshoots. The Post-task Position Optimization (PTPO) drives a reduction in time (25.12%) as well as distance (38.31%). Notably, the overall Room transitions (RTs) for the tasks are also reduced.

D. COMPARATIVE RESULTS AND ANALYSIS

The results are divided into three groups. In the first group, we compare the performance of our novel heuristic with other approaches for multi-object search. Secondly, we look at the influence of the start position on the search task. Lastly, we compare the differences in object search when incorporating the priorities set by the user.

BASELINE COMPARISON

While keeping all other parameters, such as local path planners, object detectors, and map conditions same, we benchmark our framework against the following methods:

- Random: The robot randomly chooses a landmark, navigates to that location, and searches for the object.
- Probabilistic Greedy: The robot traverses greedily while searching for the object. We define the greedy heuristic as follows:

$$\mathbf{H}_1 = P_{env}(T_1|L) + P_{env}(T_2|L) \tag{14}$$

- Distance-TSP (D-TSP): Object search is formulated as a Travelling Salesperson Problem (TSP) to determine the sequence of locations to visit. This strategy only considers optimizing the distance travelled. The TSP is solved with the Google-OR Tools' Routing solver [50].
- Scene Graph Object Search (SGOS): [38] is a State-of-the-Art framework. The authors proposed a method to incrementally build and update scene graphs of the environment as the robot searches for an object. The scene graphs were then used in conjunction with a long-horizon planning strategy to optimize distance travelled for search. Since this method is primarily for a single object search, the multi-object search task is adapted as sequential single object searches.

It is also important to consider the metrics for evaluation. We list the different metrics for evaluation below:

- Distance: The overall distance from the start of the search task to the end of the search task.
- Landmarks: The number of landmarks visited during the task.
- Time: The total time required for completion of the task.
- Coverage Probability (CP): The cumulative probability from the start of the task to finding the object. This is calculated as follows:

$$CP = \sum_{i=1}^{n-1} P_{env}(T|L) \tag{15}$$

- Probability Weighted Success (PWS): For a total of N runs, where object locations are varied with probabilities P_i and the corresponding distance to search is D_i, PWS is defined as:

$$PWS = \frac{\sum_{i=1}^{N} P_i(T|L)D_i(T,A)}{N} \tag{16}$$

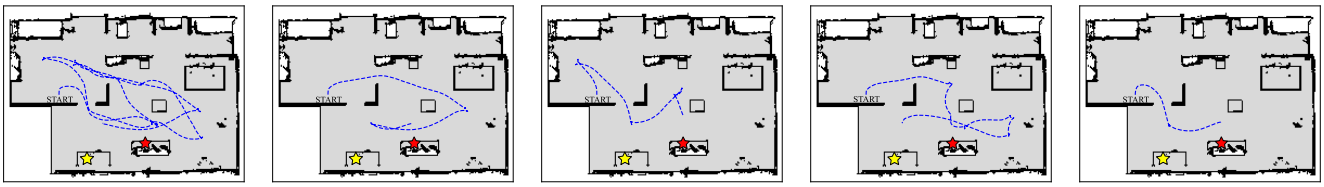
Apart from the metrics, we also consider the computational load of different methods. Our method along with D-TSP and SGOS, uses the A* planner for estimating the distance to be travelled. For comparison between these three we need not consider the load due to A* planner. Besides A*, the computationally expensive step in our framework is that of finding the maximum of the heuristic list (step 9 in Alg. 2). This has a computational load of O(n). The D-TSP and SGOS methods have a very variable computational load and in worst-cases may have $O(n^k)$ and O(n!) load, respectively. The Random search has O(1) (i.e., constant time) load but has a poor performance in every other metric. The Greedy method uses Quicksort algorithm (O(nlogn)) but without the A* planner. However, in real-world scenarios, the computational time in our experiments, hovers in the 1-2% range of the total search time.

In Table 2, we show results after comparing our method with the baselines based on the metrics described above. The starting position of the robot was kept the same in all runs. The OR module was applied for all the methods in the task. However, the PTPO module was not applicable since the objective is to assess the performance in each task separately. Three tasks (Cup+Phone, Bottle+Backpack, Remote+Phone) were assigned and runs were carried out with all methods. The trajectories followed by the robot for each method for the task (Cup+Phone) are shown in Fig. 9 below. For each method, the objects were kept at two different locations for each task. Thus, each data point in the table below is an average of twelve runs (3 tasks, 2 objects per task, and 2 locations per object: $3 \times 2 \times 2 = 12$).

TABLE 2. Comparisons with other methods.

Method	Metrics (Avg.)				
	Time(s) ↓	Ldmk ↓	Dist. (m) ↓	PWS ↓	CP ↑
Random	112.6	5.5	24.03	9.28	0.36
Greedy	55.92	**2.17**	11.62	4.21	0.43
D-TSP	69.45	4.83	10.88	4.85	0.49
SGOS	61.91	3.45	10.26	3.96	**0.54**
Ours	**55.11**	3	**8.83**	**3.32**	0.52

From our comparison, it is clear that the proposed method outperforms other methods. Specifically, the method is significantly superior to the Random and D-TSP methods by 51.6% and 20.6% respectively, in terms of time.



(a) Random (b) Greedy (c) Distance-TSP (d) SGOS (e) Ours

FIGURE 9. In this scenario, the robot is tasked with searching for a Cup (★) and Phone (★) from the same start location. The trajectories followed by the robot according to the labelled inferencing schema are shown in these figures.

Correspondingly, it is also much better in terms of number of landmarks visited (45.5%, 37.9%) and distance to search (63.2%, 18.9%).

Amongst these methods, only SGOS and Greedy method have comparable results. Greedy method has fewer landmarks visits on an average (2.17) compared to ours (3). However, a more nuanced look reveals that the time required for search is almost the same (≈1%) while the distance to search is almost 31% less for our method. Additionally, the CP value is higher for our method by 20%. This outperformance can be solely attributed to our proposed heuristic which considers minimizing the distance while maximizing the chances of finding the object simultaneously. On the other hand, SGOS has a minor advantage in CP value (0.54) over ours (0.52). However, on every other metric, our method has better results.

The Random has the highest variance as compared to other methods for obvious reasons. D-TSP follows a deterministic path since it is purely distance-dependent. Due to this, it has a relatively higher CP value. Compared to D-TSP, SGOS improves on the CP metric since it does consider the probabilistic priors in its planning.

START POSITION ANALYSIS

A key aspect missing in the previous analysis is the influence of the starting position. We compare our approach with the Probabilistic Greedy baseline when tasked with searching for the Cup from five different locations within the environment.

Fig. 10 shows the results of this comparative study. Compared to the baseline, the proposed heuristic led to an average reduction of 31.67% in the time taken, and 40.5% reduction in the distance travelled to find a Cup. Similar reductions of 26.35% and 29.3% were found in time required and distance travelled for searching the Remote. Thus, our heuristic is significantly superior to probabilistic baselines for multi-object search.

USER PREFERENCE ANALYSIS

Next, we compare the differences in object search with respect to the priorities set by the user. For this analysis, the robot is tasked with searching for the Cup and Remote. The search is initialized from ten random locations in the indoor environment. From each location three different priorities are set:

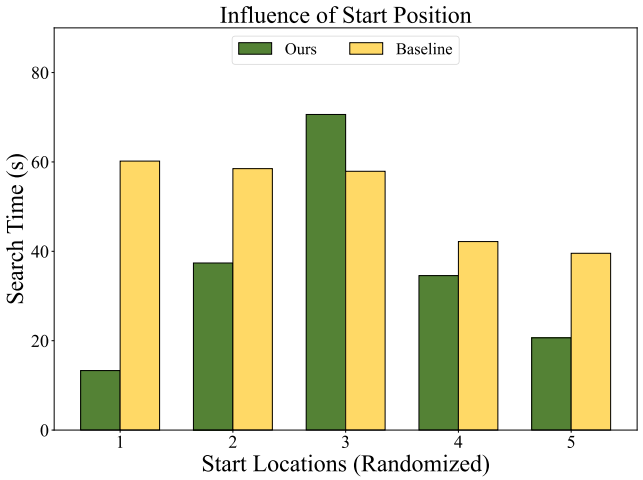


FIGURE 10. Comparison against baselines for search object: Cup. Similar results were observed for Remote as the object.

- Prioritizing Cup
- Equal Priority
- Prioritizing Remote.

A total of 30 experiments (3 × 10) were conducted. The results are summarized in Figure. 11.

Our studies show that on average a time reduction of 33.5% is observed when the user prioritizes finding the Cup. In the case of prioritizing Remote, a time reduction of 26.5% is observed compared to an equal priority search. When the user prioritizes Cup, the first-hit (i.e., success of finding the object at the first landmark visited) percentage was 20% higher than for an equal-priority search. The cumulative time spent increased by 8.94s when prioritizing the Remote and by 4.13s when prioritizing the Cup as compared to equal priority search. Additionally, if the robot were to prioritize finding the Remote, it would take 25.8s more than the equal priority search to find the Cup. Thus, it can be inferred that if the user intends to save cumulative time rather than find one object at the earliest, an equal priority directive should be given.

As discussed above, incorporating user preferences comes at the cost of an increase in cumulative time. To better assess the distance efficiency of our search strategy, we use the Success weighted by Path Length (SPL) metric. SPL is determined as follows:

$$\mathbf{SPL} = \frac{1}{N}\sum_{i=1}^{N}\frac{L_{opt}}{max(L, L_{opt})} \tag{17}$$

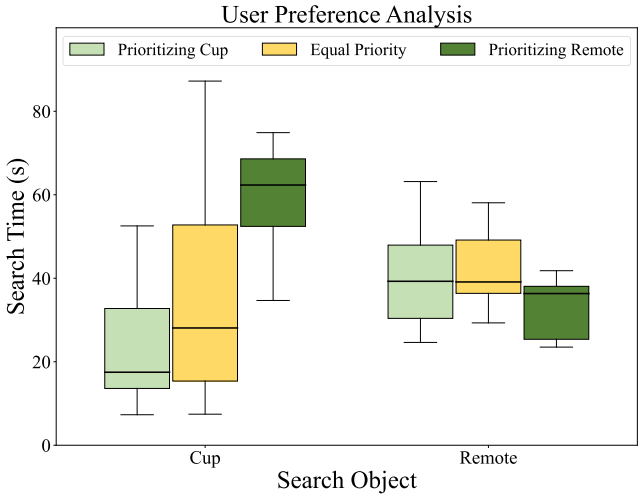


FIGURE 11. The search times for each object from ten different locations with three different initial conditions (in terms of user preferences).

where, L_{opt} = Optimal distance between Robot and Object
L = Length of path followed by Robot

The optimal distance is defined as the distance the robot would have to traverse if the object location were known to the robot. Figure 12 shows the SPL for object search (for both objects) depending on the priority set by the user.

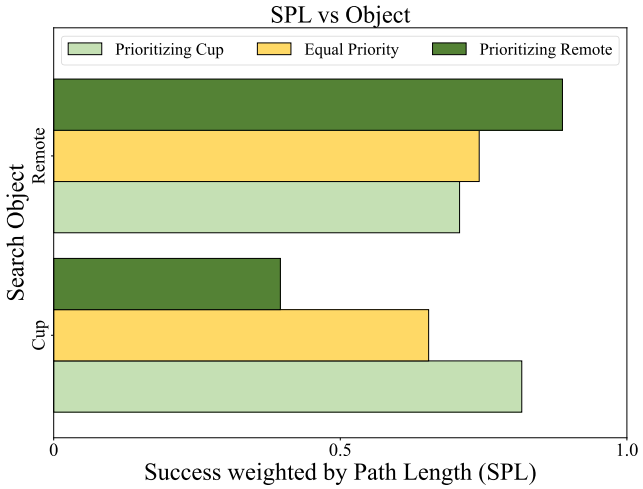


FIGURE 12. SPL values with respect to user preferences.

It is observed that the SPL increases by 24.84% when the user tasks the robot to search for a cup with priority, as compared to an equal priority search. An increase of 19.5% in the SPL is observed when prioritizing the Remote in the object search. In a cross-analysis, it is seen that the SPL decreases by 20.2% while searching for the Remote and 51.5% while searching for the Cup if the user prioritizes searching for the other object. The sharper SPL decline for the Cup can be attributed to its real location being closer to multiple landmarks as opposed to the Remote.

VI. CONCLUSION AND DISCUSSIONS

In this study, we have successfully demonstrated the feasibility of multi-object search in indoor environments by leveraging a semantic map. The first novelty of our approach lies in the extraction of 'Environment-specific' priors from the more generalized COCO dataset. Our extraction model was able to logically utilize the environment topology for fusing Ontologies of various types. We proposed a novel heuristic capable of incorporating user preferences while locating multiple objects. Our proposed approach included a reliable region-to-region navigation strategy that is efficient in terms of time. We also presented a novel Post-task Position Optimization strategy that enhances the performance over successive tasks. Our system can perform these tasks in real-time which makes it suitable for small to medium-sized indoor spaces such as homes and offices. Finally, we compared our approach to different baseline and State-of-the-Art strategies on several metrics which established the overall improvement due to our framework.

The system can be improved using decision-making strategies that optimize long-horizon navigation planners. A future non-trivial extension of this work is to add a manipulator to the system that can perform high-level 'Tidy Up' tasks and to improve its robustness to accommodate cluttered and challenging environments.

REFERENCES

[1] D. Belanche, L. V. Casaló, C. Flavián, and J. Schepers, "Service robot implementation: A theoretical framework and research agenda," *Service Industries J.*, vol. 40, nos. 3–4, pp. 203–225, Mar. 2020.

[2] A. A. Ravankar, S. A. Tafrishi, J. V. Salazar Luces, F. Seto, and Y. Hirata, "CARE: Cooperation of ai robot enablers to create a vibrant society," *IEEE Robot. Autom. Mag.*, vol. 30, no. 1, pp. 8–23, Mar. 2023.

[3] M. Kim, S. Kim, S. Park, M.-T. Choi, M. Kim, and H. Gomaa, "Service robot for the elderly," *IEEE Robot. Autom. Mag.*, vol. 16, no. 1, pp. 34–45, Mar. 2009.

[4] R. Martins, D. Bersan, M. F. M. Campos, and E. R. Nascimento, "Extending maps with semantic and contextual object information for robot navigation: A learning-based framework using visual and depth cues," *J. Intell. Robotic Syst.*, vol. 99, nos. 3–4, pp. 555–569, Sep. 2020.

[5] M. Hayat, S. H. Khan, M. Bennamoun, and S. An, "A spatial layout and scale invariant feature representation for indoor scene classification," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4829–4841, Oct. 2016.

[6] N. Zimmerman, T. Guadagnino, X. Chen, J. Behley, and C. Stachniss, "Long-term localization using semantic cues in floor plan maps," *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 176–183, Jan. 2023.

[7] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs," *Int. J. Robot. Res.*, vol. 40, nos. 12–14, pp. 1510–1546, Dec. 2021.

[8] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[10] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "TidyBot: Personalized robot assistance with large language models," *Auto. Robots*, vol. 47, no. 8, pp. 1087–1102, Dec. 2023.

[11] D. Fernandez-Chaves, J.-R. Ruiz-Sarmiento, N. Petkov, and J. Gonzalez-Jimenez, "*ViMantic*, a distributed robotic architecture for semantic mapping in indoor environments," *Knowl.-Based Syst.*, vol. 232, Nov. 2021, Art. no. 107440.

[12] S. Hasegawa, A. Taniguchi, Y. Hagiwara, L. El Hafi, and T. Taniguchi, "Inferring place-object relationships by integrating probabilistic logic and multimodal spatial concepts," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2023, pp. 1–8.

[13] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 2168–2173.

[14] C. McCormick. (Apr. 2016). *Word2vec Tutorial: The Skip-Gram Model*. [Online]. Available: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model

[15] Y. Guo, Y. Xie, Y. Chen, X. Ban, B. Sadoun, and M. S. Obaidat, "An efficient object navigation strategy for mobile robots based on semantic information," *Electronics*, vol. 11, no. 7, p. 1136, Apr. 2022.

[16] M. Zhang, G. Tian, Y. Cui, Y. Zhang, and Z. Xia, "Hierarchical semantic knowledge-based object search method for household robots," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 1, pp. 930–941, Feb. 2024.

[17] Y. Zhang, G. Tian, X. Shao, M. Zhang, and S. Liu, "Semantic grounding for long-term autonomy of mobile robots toward dynamic object search in home environments," *IEEE Trans. Ind. Electron.*, vol. 70, no. 2, pp. 1655–1665, Feb. 2023.

[18] M. Mantelli, F. M. Noori, D. Pittol, R. Maffei, J. Torresen, and M. Kolberg, "Semantic temporal object search system based on heat maps," *J. Intell. Robotic Syst.*, vol. 106, no. 4, p. 69, Dec. 2022.

[19] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat. (CIRA) Towards New Comput. Princ. Robot. Automat.*, Sep. 1997, pp. 146–151.

[20] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Auto. Robots*, vol. 33, no. 4, pp. 427–444, Nov. 2012.

[21] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "ObjectNav revisited: On evaluation of embodied agents navigating to objects," 2020, *arXiv:2006.13171*.

[22] K. Yadav, S. K. Ramakrishnan, J. Turner, A. Gokaslan, O. Maksymets, R. Jain, R. Ramrakhya, A. X. Chang, A. Clegg, M. Savva, E. Undersander, D. S. Chaplot, and D. Batra. (2022). *Habitat Challenge 2022*. [Online]. Available: https://aihabitat.org/challenge/2022/

[23] M. Zhu, B. Zhao, and T. Kong, "Navigating to objects in unseen environments by distance prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 10571–10578.

[24] W.-C. Lee and H.-L. Choi, "Complex semantic-spatial relation aided indoor target-directed exploration," *IEEE Access*, vol. 9, pp. 167039–167053, 2021.

[25] A. Milas, A. Ivanovic, and T. Petrovic, "ASEP: An autonomous semantic exploration planner with object labeling," *IEEE Access*, vol. 11, pp. 107169–107183, 2023.

[26] K. Chen, J. Liu, Q. Chen, Z. Wang, and J. Zhang, "Accurate object association and pose updating for semantic SLAM," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25169–25179, Dec. 2022.

[27] T. Ran, L. Yuan, J. Zhang, L. He, R. Huang, and J. Mei, "Not only look but infer: Multiple hypothesis clustering of data association inference for semantic SLAM," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[28] K. Doherty, D. Fourie, and J. Leonard, "Multimodal semantic SLAM with probabilistic data association," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2419–2425.

[29] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic SLAM based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75545–75559, 2018.

[30] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5079–5085.

[31] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, "Place categorization and semantic mapping on a mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5729–5736.

[32] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–9.

[33] L. Holzherr, J. Förster, M. Breyer, J. Nieto, R. Siegwart, and J. J. Chung, "Efficient multi-scale POMDPs for robotic object search and delivery," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 6585–6591.

[34] K. Zheng, Y. Sung, G. Konidaris, and S. Tellex, "Multi-resolution POMDP planning for multi-object search in 3D," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 2022–2029.

[35] K. Zheng, R. Chitnis, Y. Sung, G. Konidaris, and S. Tellex, "Towards optimal correlational object search," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 7313–7319.

[36] C. Wang, J. Cheng, W. Chi, T. Yan, and M. Q.-H. Meng, "Semantic-aware informative path planning for efficient object search using mobile robot," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 5230–5243, Aug. 2021.

[37] T. Choi and G. Cielniak, "Adaptive selection of informative path planning strategies via reinforcement learning," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Aug. 2021, pp. 1–6.

[38] F. Zhou, H. Liu, H. Zhao, and L. Liang, "Long-term object search using incremental scene graph updating," *Robotica*, vol. 41, no. 3, pp. 962–975, Mar. 2023.

[39] A. C. Hernandez, E. Derner, C. Gomez, R. Barber, and R. Babuška, "Efficient object search through probability-based viewpoint selection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 6172–6179.

[40] A. Chikhalikar, A. A. Ravankar, J. V. S. Luces, S. A. Tafrishi, and Y. Hirata, "An object-oriented navigation strategy for service robots leveraging semantic information," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2023, pp. 1–6.

[41] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "A hybrid topological mapping and navigation method for large area robot mapping," in *Proc. 56th Annu. Conf. Soc. Instrum. Control Engineers Jpn. (SICE)*, Sep. 2017, pp. 1104–1107.

[42] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, vol. 3, Kobe, Japan, 2009, p. 5.

[43] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough Terrain navigation," in *Robot Operating System (ROS)—The Complete Reference*, vol. 1. A. Koubaa, Ed., Springer, 2016, ch. 5.

[44] *Azure Kinect DK Sensor SDK*. Accessed: Aug. 13, 2022. [Online]. Available: https://docs.microsoft.com/en-us/azure/Kinect-dk/sensor-sdk-download

[45] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 2014.

[46] M. Labbé and F. Michaud, "RTAB-map as an open-source LiDAR and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, Mar. 2019.

[47] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.

[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*. Switzerland: Springer, 2014, pp. 740–755.

[49] R. Bormann, F. Jordan, W. Li, J. Hampp, and M. Hägele, "Room segmentation: Survey, implementation, and analysis," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1019–1026.

[50] V. Furnon and L. Perron. (May 2024). *OR-Tools Routing Library*. [Online]. Available: https://developers.google.com/optimization/routing/

AKASH CHIKHALIKAR (Student Member, IEEE) received the B.Tech. and M.Tech. degrees in mechanical engineering from the Indian Institute of Technology, Bombay, India, in 2019. He is currently pursuing the Ph.D. degree with the Department of Robotics, School of Engineering, Tohoku University, Japan. From 2019 to 2021, he was a Scientist with the Indian Space Research Organization (ISRO). Since 2021, he has been affiliated with the Smart Robots Design Laboratory, Tohoku University. His research interests include semantic SLAM, task and motion planning (TAMP), and open-vocabulary 3-D scene graphs (3DSG). He was a recipient of the Pioneering Research Support Fellowship from the Japan Science and Technology Agency (JST).

ANKIT A. RAVANKAR (Member, IEEE) received the M.Eng. and Ph.D. degrees in human mechanical systems and design engineering from Hokkaido University, Japan, in 2012 and 2015, respectively. He was an Assistant Professor with the School of Engineering, Hokkaido University, from 2015 to 2021. Currently, he holds the position of a Project Associate Professor with the Department of Robotics, Tohoku University, Japan. He received the MEXT Scholarship from the Government of Japan for his graduate studies. His research interests include planning and decision-making for robot sensing under uncertainty, mobile robot navigation, service robots, SLAM, path planning, computer vision, rehabilitation robotics, and multi-robot systems. He has received several awards, including the Best Journal Paper Award from the Society of Instrumentation and Control Engineers (SICE), in 2021; the International Award at IEEE SICE, in 2022; the Best Paper Award at the International Symposium on Artificial Life and Robotics (AROB), in 2020; the Young Author Award from the International Society of Artificial Life and Robotics (ISAROB), in 2017; the Best Paper Award at the IEEE ICCAS, in 2015; and the Young Author Award at the IEEE/SICE SII, in 2011.

JOSE VICTORIO SALAZAR LUCES (Member, IEEE) received the bachelor's degree in systems engineering with a minor in computer science from the Universidad de los Andes, in 2010, and the master's and Ph.D. degrees in bioengineering and robotics from Tohoku University. He is an Assistant Professor with the Smart Robots Design Laboratory, Tohoku University. He received the MEXT Scholarship, in 2013. He has published several articles in top-tier journals and conferences. His research interests include haptics and assistive robotics. He is a member of professional organizations, including RSJ and JSME. He was a recipient of the JSME Young Engineers Award, in 2019.

YASUHISA HIRATA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in mechanical engineering from Tohoku University, Japan, in 1998, 2000, and 2004, respectively. He is a Professor with the Department of Robotics, Tohoku University; and the Project Manager with the Moonshot Research and Development Program, Japan. Formerly, he was a Research Associate and an Associate Professor with Tohoku University. He was also a Visiting Researcher with the Universite de Versailles Saint-Quentin-en-Yvelines, France, from 2006 to 2012. For more than 20 years, he has been doing research on the control of multiple mobile robots in coordination, human–robot cooperation systems, assistive robots, haptics, and industrial robots. He has over 200 technical publications in the area of robotics. He served as an AdCom Member and the Vice President for TAB in IEEE RAS. He received the Nagamori Award; and the Best Paper Awards in *Advanced Robotics*, JSME Journal, RSJ Journal, and Fanuc FA Foundation.