LIVE: Lidar Informed Visual Search for Multiple Objects with Multiple Robots

Ryan Gupta*, Minkyu Kim†, Juliana T Rodriguez†, Kyle Morgenstein* and Luis Sentis*

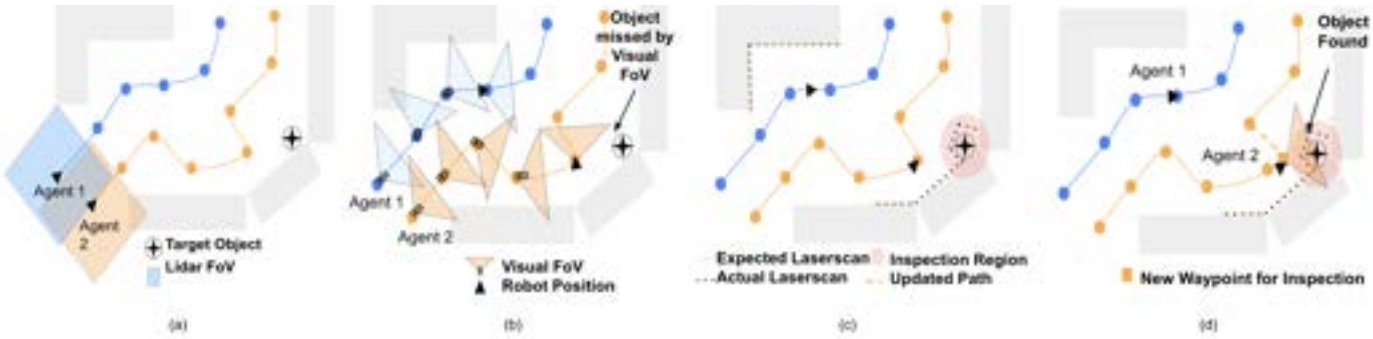

Fig. 1: An overview of LIVE. In (a), global path plans are generated for Lidar CPP. (b) While following the global paths without LIVE, the agent may miss the object of interest due to a differently sized camera and lidar FoV. (c) During localization, if an unmapped object, represented by a black circle with a cross, is discovered using lidar segmentation, it is labeled as a inspection region. (d) The Waypoint Manager (Sec. II-C) generates an intermediate waypoint for visual inspection of the unexpected object.

***Abstract*—This paper introduces LIVE: Lidar Informed Visual Search focused on the problem of multi-robot (MR) planning and execution for robust visual detection of multiple objects. We perform extensive real-world experiments with a two-robot team in an indoor apartment setting. LIVE acts as a perception module that detects unmapped obstacles, or Short Term Features (STFs), in Lidar observations. STFs are filtered, resulting in regions to be visually inspected by modifying plans online. Lidar Coverage Path Planning (CPP) is employed for generating highly efficient global plans for heterogeneous robot teams. Finally, we present a data model and a demonstration dataset, which can be found by visiting our project website https://sites.google.com/view/live-iros2023/home.**

I. Introduction

Autonomous planning and real-world execution for multi-robot (MR) CPP, Active Object Search, and Exploration are receiving significant attention from the robotics community due to their relevance in several real-world scenarios including cleaning, lawn mowing, inspection, surveillance, and SAR [19]. This paper addresses efficient and robust path planning for real-world MR teams performing multi-object visual detection by combining Coverage Path Planning (CPP) with active sensing. The goal in CPP is to generate path plans such that a sensor footprint, or Field of View (FoV), covers the region of interest [10]. Efficiency is commonly measured by coverage time, path length or FoV overlap [19]. Early work [2] consider various cost functions for continuous and varying-rate area sweeping with a single robot and multiple robots [3]. MR CPP offers several benefits including speed and resilience to robot failure and is a common feature in search and rescue (SAR) and other critical applications. State of the art work in multi-aerial vehicle exploration in confined spaces [6] consider the problem of multi-sensor exploration for the purpose of visual mapping or inspection of surfaces. They leverage range scan data for guiding visual surface exploration. Instead, we propose to leverage range scan information for guiding visual search. Active Sensing was first proposed in [5] as a method of providing a control strategy based on current world state, updated by observations, and is frequently employed in information gathering missions. Active sensing has proven to be an effective tool for information gathering missions including SLAM [9], search [4, 21], and object tracking [15]. In the MR setting, Gosrich et al. [12] enable robot teams position themselves to observe events using a Graph Neural Network for non-local information during decision making. In [14], MR multi-object search in unknown environments is cast as a reinforcement learning problem and address non-myopic planning.

While MR CPP offers several benefits, computing optimal paths for multiple agents is NP-Hard [13]. It remains an ongoing problem to improve path efficiency and computational load in MR CPP and exploration [19]. Kim et al. [16] provide high efficiency plans by combining sampling and optimization. However, in visual CPP with a small FoV RGB camera, it becomes impractical to plan online due to increased visitation sites. Furthermore, [11] notes the lack of multi-robot systems deployed for real-world autonomous search, even with increased publications. They indicate a need for research focused on the realistic evaluation of methods for real-world search.

Mobile robots are frequently equipped with Lidar, which cast a significantly wider FoV than an RGB camera. LIVE leverages the wide FoV Lidar sensor to inform visual inspection in real-world experiments. LIVE classifies incoming Lidar observations to remove dynamic features and

*Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, Austin, TX 78712 USA ryan.gupta@utexas.edu
†Department of Mechanical Engineering, University of Texas at Austin, Austin, TX 78712 USA

static map obstacles leaving Short Term Features (STFs), defined as static, unmapped obstacles. Raw STFs are filtered into inspection regions, defined as possible target object locations. Agents select among inspection regions, inspect them visually, then continue along global plans. An overview is shown in Fig. 1. This online modification of global plans enables the fast planning of efficient paths based on wide FoV Lidar scans at the global stage, while still managing robust visual results from the active sensing approach.

The contributions of this work can be summarized as follows:

- Propose a new method for incorporating lidar information to real-world multi-robot multi-object visual search
- Uniquely combine Lidar CPP and Active Sensing for visual object search
- Deploy extensively in a heterogeneous two-robot system for verification and baseline comparison

II. Methods

This paper focuses on efficient and robust multi-robot multi-object detection in indoor environments with static map information known. The approach leverages global multi-robot CPP for efficient global plans and a perception module capable of modifying those plans online. An overview can be seen in Fig. 1. First, global path plans are generated for heterogeneous multi-robot teams [16].

A. Search Map (Entropy Map)

Bayesian filtering is employed to maintain a target estimate over the map. Each cell is assigned a probability of occupancy between 0 and 1. A cell is initialized to 0.5, except those corresponding to the static map, which are assigned 1. Cells that have been visited by the sensor FoV and are free are assigned 0. The search map is updated at each step when the central server receives local costmap observations from each robot that represent sensor FoV. In this work, local costmaps are rectangular, representing Lidar FoV, or triangular, representing visual FoV. Entropy over the map is measured by considering all cells in the 2D global costmap and is computed as

$$H(M_t) = -\sum_{i=1}^{N}(m_t^i \log(m_t^i) + (1-m_t^i)log(1-m_t^i)) \quad (1)$$

where m_t^i is the occupancy variable at time step t and N denotes the total number of cells. For further details refer to [16].

B. Inspection Region Detection

Computing inspection regions begins with the assumption that the objects of interest belong to the set of unmapped obstacles. Incoming Lidar observations are classified based on current robot pose estimate. Each point in the 2D scan is classified as a Long Term Feature (LTF), Short Term Feature (STF), or Dynamic Feature (DF) [8]. LTFs represent the static map obstacles and STFs represent static unmapped obstacles. Let x_i denote the pose of the robot, and s_i denote observation at time step t_i. Each observation s_i consists of

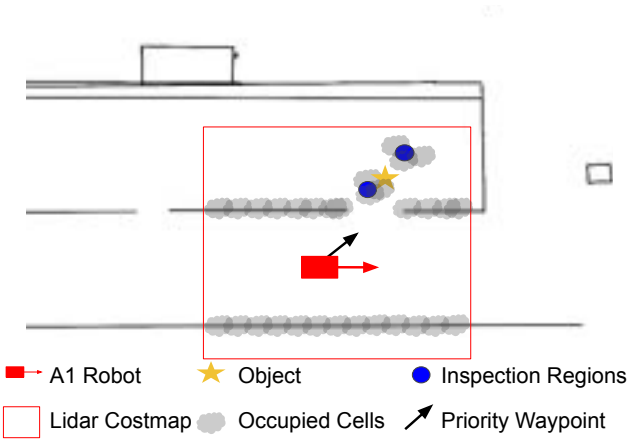

Fig. 2: The lidar costmap is shown over a portion of the map to demonstrate the Lidar FoV detecting unmapped obstacles. Unmapped obstacles, or STFs, are filtered into inspection regions. This figure occurs the moment A1 is given a priority waypoint from LIVE for viewing an inspection region.

n_i 2D points, $s_i = \{p_i^j\}_{j=1:n_i}$. Observations are transformed from robot local frame into the global frame using an affine transformation $T_i \in SE(3)$. Finally, let map M be represented as a set of lines $\{l_i\}_{1:n}$.

1) LTF: First, an analytic ray cast is performed [7] to determine expected laserscan based on map M and current robot position x_i. Given observations, the probability that points correspond to one of the lines of that static map can be written

$$P(p_i^j|x_i, M) = exp\left(-\frac{dist(T_i p_i^j, l_j)^2}{\Sigma_s}\right) \quad (2)$$

where Σ_s is the scalar variance of observations, which comes from sensor accuracy. If Eq. 2 is greater than a threshold, point p_i^j is classified as a LTF.

2) STF: Remaining points will be classified as STF or DF. Observations at current time i, p_i^j, are compared with prior observations at time k, p_k^l to determine correspondence between points in subsequent observations. The likelihood of the remaining points corresponding to the same point as in a previous laserscan is computed as

$$P(p_i^j, p_k^l|x_i, x_k) = exp\left(-\frac{||T_i p_i^j - T_k p_k^l||^2}{\Sigma_s}\right) \quad (3)$$

where p_l^k is the nearest point from p_i^j among points which does not belong to LTF at other timesteps, defined as

$$p_k^l = \arg min||T_i p_i^j - T_k p_k^l|| \quad (4)$$

When Eq. 3 is greater than some threshold, point p_i^j is classified as an STF. Remaining points in s_i are classified as DFs, which are ignored in this study.

3) Inspection Region Selection: STFs obtained in the previous subsection are generated stochastically using the entire set of Lidar points from estimated robot pose. As a result, the set of raw STFs is large and filtering steps are critical. First, the pooling operator is used to reduce duplicates

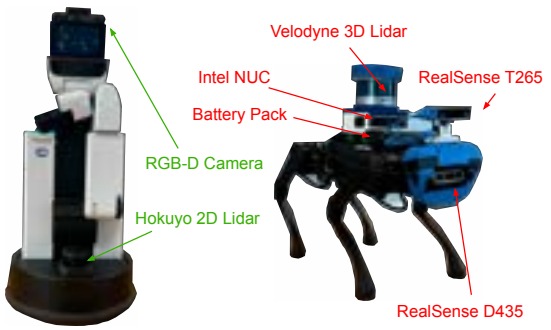

Fig. 3: Figure of two robot team with key components labeled.

within a radius. Second, pooled STFs within a certain distance of LTFs are removed to eliminate false positives caused by localization drift. Finally, points inside visually observed regions of the map are removed. The remaining points are inspection regions. During waypoint generation the nearest inspection region is selected for generating a priority waypoint. The result of this process is shown in Fig. 2.

C. Waypoint Manager

The Waypoint Manager takes global paths as input and acts as a finite state machine to determine when to send the robot an updated navigation waypoint. This node also receives priority waypoints for viewing inspection regions at each localization timestep. Due to the high update frequency and the stochastic nature of inspection regions, exhaustively visiting all priority waypoints is inefficient. As a result, this node incorporates priority waypoints every so often if they are available. The maximum rate of occurrence of priority waypoints is a tunable parameter that will impact robustness and target detection time. After visiting the priority waypoint, the manager will resume along the global path plans. A priority waypoint being selected is shown in Fig. 2.

III. Task Description and Experiment Setup

A team comprised of the Unitree A1 Quadruped [1] and the Toyota HSR [20] must visually detect two static objects of interest. Specifically, robots must find two small suitcases in a 20mx30m apartment setting with an attached hall. The goal of the team is to robustly detect objects where efficiency is measured as path length. Robots with their sensors and instrumentation are described in Fig. 3. The HSR employs Toyota move base to generate movement commands from given waypoints. The A1 leverages a carrot planner to navigate to waypoints. LIVE and waypoint manager nodes run aboard each robot. Two maps are maintained: the first is the Search Map, which is implemented as a 2D global costmap in ROS and the second is a vectormap used for localization [8], shown in Fig 4. Both maps can be found alongside the dataset at the project website. A central search server generates path plans and maintains the search map by sending global plans to robots and receiving position and costmap information back. Communication between the central server and each robot are performed using Robofleet [17]. A local network covers the full region using a ASUS AC1900 WiFi router. The laptop and

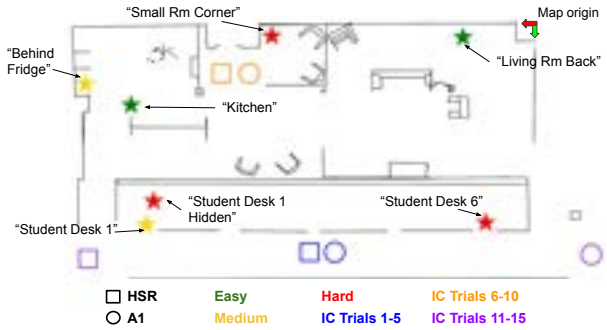

Fig. 4: Top view of the static map with each of the three robot initial conditions (IC) and seven object location labeled. A1 ICs are circles and HSR ICs are squares, color-coordinated for the trials. The object locations are shown by stars whose color indicates difficulty with location names included.

Heuristic Visual CPP			
	HSR Avg.	**A1 Avg.**	**Overall Avg.**
IC1	36.9	46.5	41.7
IC2	15.0	32.2	23.6
IC3	30.2	44.1	37.2
	27.4	40.9	**34.2**

Lidar CPP + LIVE (Our Method)			
	HSR Avg.	**A1 Avg.**	**Overall Avg.**
IC1	19.4	32.9	26.1
IC2	17.0	22.5	19.7
IC3	25.2	46.0	35.6
	20.5	33.8	**27.2**

TABLE I: Table showing average path lengths for each robot and combined in meters as a function of initial condition.

onboard computers run Ubuntu 18.04 and ROS Melodic [18] to implement all of the capabilities. Three planner settings are compared: 1) Lidar CPP, 2) Heuristic Visual CPP, and 3) Lidar CPP + LIVE The three planner settings are described in Section II. In each of the three planning settings, 15 trials performed, for a total of 30 potential objects to be found. The 15 trials are composed of five trials from each of three different initial conditions (IC), depicted on the static map in Fig. 4. Object locations change between trials with varying difficulty. The same five sets of object locations are tested from each IC. There are seven total object locations used, categorized as easy, medium, or hard and they are also shown in Fig. 4.

IV. Results & Discussion

Complete tabular results can be found on the project website, including trial-by-trial results, path lengths compared by initial condition, failure mode analysis, and success rate versus object difficulty. Videos and robot trajectories for all trials can be found via the Google Drive link on the website. Figure 5 shows the success rate for each of the three planner settings as well as the relative frequency of each failure mode. A trial is classified 'Path Failure' if the agents successfully follow the path plans output by the planner and failed to visually detect at least one object. 'HSR Navigation Failure' indicates the HSR bumped into an object and triggered the emergency stop. 'A1 Locomotion Failure' occurs if the A1 falls while walking. 'Object Detection Failure' occurs when at least a portion of the target object is in the RGB camera feed, yet the object detection algorithm misses it. Notably, the occurrence of path failure in the Heuristic Visual CPP is similar to the

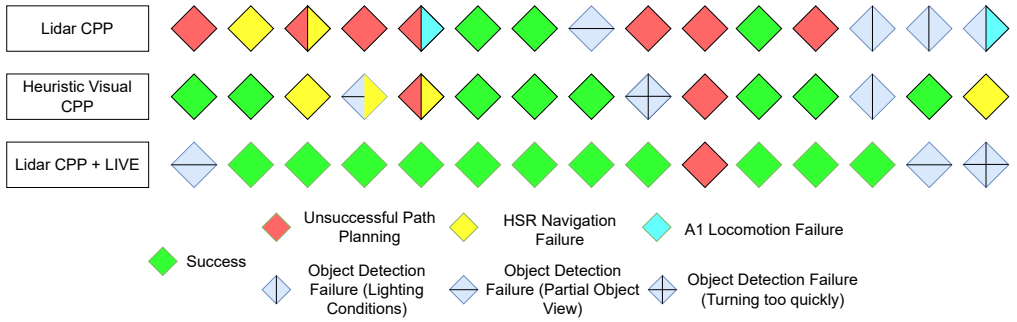

Fig. 5: Success / Failure Modes for each of 15 trials in all three path planning variants. The addition of LIVE significantly improves overall success rate and reduces the frequency of HSR Navigation Failure with more efficient path lengths.

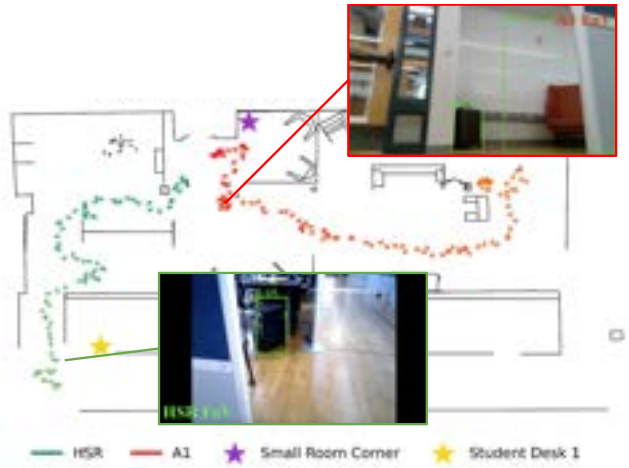

Fig. 6: Top view of the static map with robot trajectories from LIVE trial 9 overlaid. Arrows represent robot pose as recorded using rosbag during execution. Included is the moment of object detection for each of the robots connected with the robot pose at that moment.

proposed method, while Lidar CPP fails at a significantly higher rate, generating insufficient paths 40% of the time. Path lengths are computed in each setting with results from the proposed method and visual CPP in Table I, with Lidar CPP results on the website for brevity. In trials where one or more object is missed, the total lengths of the paths traversed by the robots is considered. While Visual CPP rarely fails due to inadequate global paths, it suffers from 'HSR Navigation Failures' in 16.7% of trials. This can be attributed to the near 50%, increase in path length over the other methods, resulting in higher localization drift and ultimately navigation failure.

While the Heuristic Visual CPP path plans are successful, Lidar CPP + LIVE shows to be more robust in finding the objects in real-world scenarios due to this reduction in path length. Trials throughout all three planning methods suffer from object detection shortcomings. In some cases, the robot turns too quickly for the algorithm to detect the suitcase. In others, detection was impacted by reflections of the sun and partial object views, however, the occurrence is nearly equal over the three planner settings. The results indicate the addition of LIVE improves real-world task performance with a success rate boost of 20% as compared with Heuristic Visual CPP and 50% as compared with the Lidar CPP baselines.

We further inspect success rate for each method based on object difficulty. In each of the three settings, nine objects are located in Easy, 12 in Medium, and nine in Hard locations. Fig. 4 depicts named object locations with color coded difficulty level. Results in tabular form are omitted for brevity but can be found on the project website. Easy object locations are found nearly 100% of the time across all three experiment settings. When object locations are Medium or Hard, however, Lidar CPP baseline performs poorly, with combined success rate of 14.3%. This result is expected given that Lidar CPP does not account for the visual sensor. In particular, Lidar CPP + LIVE detects 100% of Medium objects and 67% of Hard objects compared to Heuristic Visual CPP's 67% and 44%, respectively. Priority waypoints from LIVE are shown to result in successful object detection in trials 4, 5, 8, 9, and 15. Specifically, these trials have objects in Medium and Hard locations. Fig. 6 is a representative trajectory plot from LIVE trial 9. Both objects in trial 9 are found with a priority waypoint and the moment of detection is overlaid on trajectories.

Time data for the experiments can be found on the project website. The data displays a trend that LIVE improves path length efficiency and success rate, but must trade off detection time to explore inspection regions to achieve such success. This can be seen in particular in LIVE trial 13 before the HSR finds the object located 'Behind Fridge.' We note the following on comparing times for the three methods tested: 1) Due to the relative success rates, the data is skewed towards those more difficult trials where the baseline methods are less successful, 2) The variance of time data is high, indicating the initial conditions and object locations are well dispersed, and 3) During some trials there are pauses in robot navigation, confounding the relationship between actual search time and quality of paths generated.

V. Conclusion

We present an algorithm that leverages efficient global path plans, 2D range data and map information to efficiently find objects in known environments. We present results supporting that LIVE is more robust and efficient than global planning methods alone for real-world multi-object search. Ongoing work involves extending this method to unknown environments with supervised learning. Further, incorporation of inspection regions in a utility function for viewpoint selection.

ACKNOWLEDGMENTS

This research was supported in part by NSF Award #2219236 and Living and Working with Robots, a core research project of Good Systems, a UT Grand Challenge. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. Research was in part sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-19-2-0333. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] Unitree-A1. https://www.unitree.com/products/a1/.

[2] Mazda Ahmadi and Peter Stone. Continuous area sweeping: A task definition and initial approach. In *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pages 316–323. IEEE, 2005.

[3] Mazda Ahmadi and Peter Stone. A multi-robot system for continuous area sweeping tasks. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1724–1729. IEEE, 2006.

[4] Alper Aydemir, Andrzej Pronobis, Moritz Göbelbecker, and Patric Jensfelt. Active visual object search in unknown environments using uncertain semantics. *IEEE Transactions on Robotics*, 29(4):986–1002, 2013.

[5] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[6] Graeme Best, Rohit Garg, John Keller, Geoffrey A Hollinger, and Sebastian Scherer. Resilient multi-sensor exploration of multifarious environments with a team of aerial robots. In *Robotics: Science and Systems (RSS)*, 2022.

[7] Joydeep Biswas and Manuela Veloso. Depth camera based indoor mobile robot localization and navigation. In *2012 IEEE International Conference on Robotics and Automation*, pages 1697–1702. IEEE, 2012.

[8] Joydeep Biswas and Manuela M Veloso. Episodic non-markov localization. *Robotics and Autonomous Systems*, 87:162–176, 2017.

[9] Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4791–4798. IEEE, 2015.

[10] Howie Choset, Kevin M Lynch, Seth Hutchinson, George A Kantor, and Wolfram Burgard. *Principles of robot motion: theory, algorithms, and implementations*. MIT press, 2005.

[11] Daniel S Drew. Multi-agent systems for search and rescue applications. *Current Robotics Reports*, 2(2):189–200, 2021.

[12] Walker Gosrich, Siddharth Mayya, Rebecca Li, James Paulos, Mark Yim, Alejandro Ribeiro, and Vijay Kumar. Coverage control in multi-robot systems via graph neural networks. In *2022 International Conference on Robotics and Automation (ICRA)*, page 8787–8793. IEEE Press, 2022. doi: 10.1109/ICRA46639.2022.9811854. URL https://doi.org/10.1109/ICRA46639.2022.9811854.

[13] Teng Guo and Jingjin Yu. Sub-1.5 time-optimal multi-robot path planning on grids in polynomial time. In *Robotics: Science and Systems (RSS)*, 2022.

[14] Conor Igoe, Ramina Ghods, and Jeff Schneider. Multi-agent active search: A reinforcement learning approach. *IEEE Robotics and Automation Letters*, 7(2):754–761, 2021.

[15] Minkyu Kim and Luis Sentis. Active object tracking using context estimation: handling occlusions and detecting missing targets. *Applied Intelligence*, pages 1–12, 2022.

[16] Minkyu Kim, Ryan Gupta, and Luis Sentis. Concerts: Coverage competency-based target search for heterogeneous robot teams. *Applied Sciences*, 12(17), 2022. ISSN 2076-3417. URL https://www.mdpi.com/2076-3417/12/17/8649.

[17] Kavan Singh Sikand, Logan Zartman, Sadegh Rabiee, and Joydeep Biswas. Robofleet: Secure Open Source Communication and Management for Fleets of Autonomous Robots . In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 406–412, 2021. doi: 10.1109/IROS51168.2021.9635830.

[18] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. URL https://www.ros.org.

[19] Chee Sheng Tan, Rosmiwati Mohd-Mokhtar, and Mohd Rizal Arshad. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access*, 2021.

[20] Takashi Yamamoto, Koji Terada, Akiyoshi Ochiai, Fuminori Saito, Yoshiaki Asahara, and Kazuto Murase. Development of human support robot as the research platform of a domestic mobile manipulator. *ROBOMECH journal*, 6(1):1–15, 2019.

[21] Yiming Ye and John K Tsotsos. Sensor planning in 3d object search: its formulation and complexity. In *The 4th International Symposium on Artificial Intelligence and Mathematics, Florida, USA*, 1996.