Toward Decentralized Heterogeneous Multi-Robot SLAM and Target Tracking

Ofer Dagan[1], Tycho L. Cinquini[1], Luke Morrissey[2], Kristen Such[2], Nisar R. Ahmed[1], Christoffer Heckman[2]

I. INTRODUCTION

In many robotics problems, there is a significant gain in collaborative information sharing between multiple robots, for exploration, search and rescue, tracking multiple targets, or mapping large environments. In particular, there are different approaches to solving multi-robot state estimation problems: they can be solved in a centralized manner, where all robots send their data to a fusion center that serves all robots; or they can be solved in a distributed or decentralized manner using algorithms such as consensus [1] or decentralized data fusion (DDF) [2]. To the best of our knowledge, all of these approaches have a common assumption – that each robot implements the same (homogeneous) underlying estimation algorithm. However, in practice, we want to allow collaboration between robots possessing different capabilities and that therefore must rely on heterogeneous algorithms.

We present a system architecture and the supporting theory to enable collaboration in a decentralized network of robots, where each robot relies on different estimation algorithms. To develop our approach, we focus on the application to multi-robot simultaneous localization and mapping (SLAM) with multi-target tracking. Our theoretical framework builds on our idea of exploiting the conditional independence structure inherent to many robotics applications [3] to separate between each robot's local inference (estimation) tasks and fuse only relevant parts of their non-equal, but overlapping probability density function (pdfs). We rely on the widely used factor framework [4], and its advantages in representing conditional independence to develop easily scalable, statistically consistent heterogeneous fusion algorithms [5], [6]. These algorithms significantly reduce local communication and computation requirements and enable robots to share information for SLAM with dense and metric-semantic maps while tracking dynamic targets and using different sensor suites. The ideas and work presented here demonstrate progress to allow a network of robots to collaboratively share information, irrespective of their local SLAM and tracking algorithms.

In multi-robot SLAM, most works are based on sparse landmark SLAM where robots share and fuse sub-maps e.g., [7], [8], [9], [10]. In [7], Nettleton *et al.* selects informative parts of the map to communicate between robots. They use a hybrid channel filter (CF) – covariance intersection (CI) algorithm to ensure that previously communicated map data is counted not more than once. Reece and Roberts [8] improve upon this work by replacing the CI with a less conservative fusion rule called Bounded Covariance Inflation. In [9] Julier and Uhlmann use different variations of the CI algorithm [11] to solve the Full Covariance SLAM problem. Their approach suggests building three types of maps: main, relative, and local, using different instantiations of a Kalman filter (KF) and then fusing them together using CI to account for unknown dependencies between the estimates. In [12], Tchiev and Indelman consider the problem of semantic distributed SLAM with sparse landmarks, where robots fuse belief over both the map and classes of objects. To avoid double counting previously communicated data and simplify bookkeeping, they assume the communication network topology is undirected and acyclic.

While for sparse landmark SLAM it is possible to share parts of the map, i.e. coordinates of landmarks, this approach is less feasible when considering dense maps [13], [14] and metric-semantic SLAM algorithms [15]. For that reason, recent approaches either use a centralized server to merge maps [16], [17], or choose to collaboratively optimize the robots' trajectories (instead of the map) using pose graph optimization (PGO) techniques [18]. The improved trajectory estimate is then fed back to correct the local map estimate.

The problem of collaborative tracking and SLAM has gotten much less attention in the literature than collaborative SLAM. In [19] Wang *et al.* formulates the SLAM and detection and tracking of moving objects (DATMO) problem in Bayesian terms. For a single robot problem, they suggest splitting the SLAM and DATMO problems into two separate solvers but assume that data regarding the moving object (target) does not carry any information concerning the map and robot pose. Moratuwage *et al.* [20] use random finite sets (RFS) for collaborative multi-vehicle SLAM with moving object tracking. Their work solves for sparse landmark representation of the map and is not fully decentralized, as the posterior pdf is based on all measurements and trajectories.

In this paper, we present a new decentralized graph-based approach to the multi-robot SLAM and tracking problem. We leverage factor graphs to split between different parts of the problem for efficient data sharing between robots in the network while enabling robots to use different local sparse landmark/dense/metric-semantic SLAM algorithms. The paper is organized as follows: Sec. II defines the decentralized heterogeneous multi-robot SLAM and tracking

[1] Ofer Dagan, Tycho L. Cinquini, and Nisar R. Ahmed are with the Smead Aerospace Engineering Sciences Department, University of Colorado Boulder, Boulder, CO 80309 USA ofer.dagan@colorado.edu; nisar.ahmed@colorado.edu [2] Luke Morrissey, Kristen Such, and Christoffer Heckman are with the Computer Science Department, University of Colorado Boulder, Boulder, CO 80309 USA kristen.such@colorado.edu; christoffer.heckman@colorado.edu

problem, Sec. III presents our factor graph-based technical approach, and Sec. IV describes our planned simulation study and a roadmap for future hardware experiments.

II. Problem Statement

Consider a network of $n_r = |N_r|$ robots in an unknown joint environment. Each robot reasons about the environment and tries to gain situational awareness using its local, potentially different, SLAM algorithm. In addition, the robots are tasked with jointly inferring (estimating) the unknown position of a subset of $n_t = |N_t|$ static or dynamic targets at time step k. In a probabilistic approach, the uncertainty over full system multi-robot SLAM and multi-target tracking problem can be described by,

$$p(\chi|Z_k) = p(X_{k:0}, M, T_k|Z_k). \tag{1}$$

Here, $p(\chi|Z_k)$ is the pdf over the full system set of random variables (rvs), conditioned on the data set $Z_k = \bigcup_{i \in N_r} Z_k^i$ gathered by all robots. $X_{k:0}$ is the set of rvs describing the uncertain poses of all n_r robots at time steps 0 to k, T_k is the set of rvs describing the uncertain position of all n_t targets at time step k, and M is a set of rvs representing the map. Note that we defined M rather ambiguously, so as to enable heterogeneity in the system. This allows each robot to hold a different representation of the environment, e.g. sparse landmarks, point clouds, and labeled objects.

Decentralized SLAM and Tracking:

In a decentralized formulation of the problem, each robot i is tasked with inferring a subset of the full set of rvs $\chi^i \subseteq \chi$: its own pose $x_{k:0}^i \subset X_{k:0}$, local map $M^i \subseteq M$, and the states T^i of a subset $N_t^i \subseteq N_t$ of the targets. The local inference task of any robot $i \in N_r$ can be described by,

$$p^i(\chi^i|Z_k^i) = p(x_{k:0}^i, M^i, T_k^i|Z_k^i). \tag{2}$$

In decentralized SLAM and tracking, each robot maintains a pdf (2) based on the data it gathers from local sensors and the data it receives from neighboring robots through peer-to-peer communication. Since each robot maintains a different, but overlapping pdf with its neighbors, this is an instance of a heterogeneous fusion problem [3]. In heterogeneous fusion, we split the robot's set of rvs χ^i to common rvs $\chi_C^i = \bigcup_{j \in N_r^i} \chi_C^{ij}$, local rvs χ_L^i which are not monitored by any other robot in the network, and non-mutual rvs $\chi^{i\backslash j} = \chi_L^i \cup \{\chi_C^i \backslash \chi_C^{ij}\}$. In our proposed SLAM and tracking, the common rvs are the common target states T_C^{ij}, the local rvs are the local map, and the robot's pose states; and $T^{i\backslash j}$ is the subset of targets monitored by robot i, and possibly other robots, but not j.

III. Technical Approach

The key part of our approach is exploiting the probabilistic conditional independence structure of the problem to: (i) enable robots to use different SLAM algorithms and only share 'relevant' parts of their pdfs with each other for fusion, and (ii) enable inner-loop robot architecture that separates the SLAM solution, which includes the local map and ego pose, from the target tracking solution. First, let us look at the example factor graph in Fig. 1 showing the dependency structure of a two-robot problem.

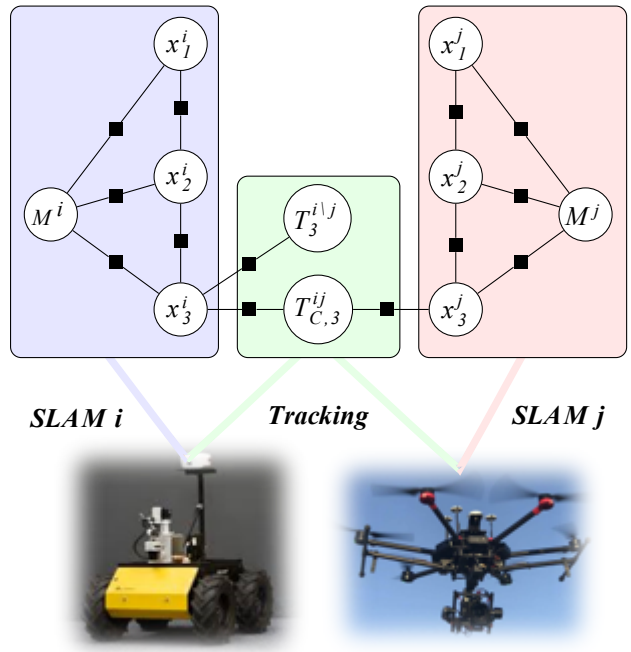

Fig. 1: Factor graph representation of a 2-robot SLAM and target tracking application. Both robots i and j are building local maps M^i and M^j while cooperatively tracking targets in common T_C^{ij} (observed at time step 3). The data fused regarding common targets will indirectly update the map and the estimate of non-mutual targets $T^{i\backslash j}$.

A factor graph [4] is an undirected bipartite graph $\mathcal{F} = (U, V, E)$ that represents a function, proportional to the joint pdf over all random variable nodes $v_m \in V$, and factorized into smaller functions given by the factor nodes $f_l \in U$. An edge $e_{lm} \in E$ in the graph only connects a factor node l to a variable node m. The joint distribution over the graph is then proportional to the global function $f(V)$:

$$p(V) \propto f(V) = \prod_l f_l(V_l), \tag{3}$$

where $f_l(V_l)$ is a function of only those variables $v_m \in V_l$ connected to the factor l. That enables the graph to explicitly express the conditional independence structure of the problem and makes factor graphs an attractive representation for decentralized inference problems [5].

The scenario depicted in the graph (Fig. 1) shows two robots i and j, tasked with a SLAM problem – estimating their pose $x_{k:1}^i$ ($x_{k:1}^j$) and the local map M^i (M^j). At the same time, the two robots are tasked with estimating the state of a set of common targets T_C^{ij} and a set of targets, only tracked by robot i, $T^{i\backslash j}$. From the graph, we see that non-mutual variables (pose, map, exclusive targets) of each robot are conditionally independent given the common target variables T_C^{ij}. With this conditional independence structure, we can use the heterogeneous fusion rule developed in [3],

to perform a peer-to-peer fusion of data regarding common targets,

$$p_f^i(\chi^i|Z_k^{i,+}) \propto$$
$$p^i(x_{k:0}^i, M^i, T^{i\setminus j}|T_{C,k}^{ij}, Z_k^{i,-}) \cdot \frac{p^i(T_{C,k}^{ij}|Z_k^{i,-})p^j(T_{C,k}^{ij}|Z_k^{j,-})}{p_c^{ij}(T_{C,k}^{ij}|Z_k^{i,-} \cap Z_k^{j,-})}. \tag{4}$$

Here $Z_k^{i,-}$ and $Z_k^{i,+}$ are the data sets available for agent i at time k prior and post fusion, respectively.

Second, we can again exploit conditional independence, now in the inner-robot level, to separate the map M^i from the targets T^i. Looking at the two left boxes (purple and green) in Fig. 1, the map and tracking variables are conditionally independent given the ego poses in which the targets were observed, i.e. in this example $M^i \perp T^i|x_3^i$. With this conditional independence, drawing inspiration from the *NET-DDF* architecture suggested in [21], we can separate between the SLAM and tracking solutions. As shown in Fig. 2, we suggest using two modules - a tracking module, to handle the robot's tracking assignment, and a SLAM module, to solve for the robot's pose and the local map. This architecture contains a filter and a channel filter (CF) [22] to explicitly track the data transferred between the local filter, responsible for tracking, and the SLAM module. The architecture expresses the following factorization of (2) and data fusion between the two modules,

$$p^i(\chi^i|Z_k^i) =$$
$$\frac{\overbrace{p(M^i|x_{k:0}^i, Z_k^{i,s})p(x_{k:0}^i|Z_k^{i,s})}^{\text{SLAM Module}} \cdot \overbrace{p(x_K^i|Z_k^{i,t})p(T_k^i|x_K^i, Z_k^{i,t})}^{\text{Tracking Module}}}{\underbrace{p(x_K^i|Z_k^{i,s} \cap Z_k^{i,t})}_{\text{CF}}}. \tag{5}$$

Where $Z_k^{i,s}$ and $Z_k^{i,t}$ are the data sets gathered by the SLAM and tracking modules, respectively, and K describes a vector of time steps in which the targets were observed.

The pipeline then works in the following way:

1) **Initialization:** Each robot i instantiates (i) a local SLAM engine (purple in Fig. 2), (ii) a tracking filter (green in Fig. 2) to estimate the set of N_t^i targets, represented by a factor graph, (iii) a stack of CFs, one for the SLAM-Tracking data tracking (pink in Fig. 2) and another one for each communication link with neighboring robots (not shown), to track common data regarding common targets [5], [6].
2) **Run SLAM:** Each robot i uses its' local SLAM algorithm to maintain a map of the environment and an ego pose estimate (Fig 2(a)).
3) **Request pose estimate:** When the robot observes a target $t \in T^i$, the tracking module 'asks' the SLAM module for a pose estimate.
4) **Send pose estimate:** The SLAM module sends a factor (or a set of factors) proportional to the marginal pose estimate, e.g., the blue factor in Fig. 2(a) and (c).
5) **Receive pose estimate:** After removing common data by the CF, a new factor is integrated into both the CF and the tracking module's graphs.
6) **Peer-to-peer fusion:** The tracking module can now communicate with a neighboring robot over common targets according to Fig. 1, and using the following version of the heterogeneous fusion rule (4),

$$p_f^i(x_K^i, T^i|Z_k^{i,t,+}) \propto p^i(x_K^i, T^{i\setminus j}|T_{C,k}^{ij}, Z_k^{i,t,-}) \times \frac{p^i(T_{C,k}^{ij}|Z_k^{i,t,-})p^j(T_{C,k}^{ij}|Z_k^{j,t,-})}{p_c^{ij}(T_{C,k}^{ij}|Z_k^{i,t,-} \cap Z_k^{j,t,-})}. \tag{6}$$

7) **Send pose back to SLAM:** The tracking module removes common data using the CF and sends the SLAM module marginal ego pose factors (orange factor in 2(b) and (d)), as it now has new data to send, both from the robot-target measurement and indirectly from fusion with a neighboring robot j.
8) **Add factors to SLAM:** SLAM module adds new data from the tracking module to the SLAM graph (orange factor in 2(b) and (d)).
9) **Repeat:** Process repeats recursively.

IV. Simulation Plan and Experimental Roadmap

To test our approach, we will simulate $n_r \geq 2$ robots moving in a joint Gazebo environment [23], tracking $n_t \geq 5$ moving targets. The robots estimate the $2D$ position and velocity of each target $t_k^t = [X_k^t, \dot{X}_k^t, Y_k^t, \dot{Y}_k^t]^T$. Targets are assumed to be moving according to the following linear kinematics model,

$$t_{k+1}^t = Ft_k^t + Gu_k^t + \omega_k, \quad \omega_k \sim \mathcal{N}(0, 0.08 \cdot I_{n_x \times n_x}),$$
$$F = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ 0 & \Delta t \end{bmatrix}. \tag{7}$$

Where Δt is the time step, u_k^t is the control input of target t, and ω_k is zero-mean additive white Gaussian (AWGN) noise. When a target t is within robot i's range, it can take a relative position measurement,

$$y_k^{i,t} = \begin{bmatrix} X_k^i - X_k^t \\ Y_k^i - Y_k^t \end{bmatrix} + v_k^i, \quad v_k^i \sim \mathcal{N}(0, R_k^i), \tag{8}$$

where v_k^i is zero-mean AWGN noise.

To test our system, we need a dataset that includes data describing (i) multiple robots in a joint environment, and (ii) different measurement modalities, e.g., LiDAR and camera. To the best of our knowledge, such a dataset does not exist. For example, [24] contains several sequences that can be used in parallel to simulate collaborative SLAM, but all sequences contain only visual-inertial data. On the other hand, [25] provides data from multiple sensors such as IMU, LiDAR, and Camera, but it only includes one sequence per scenario, which is not applicable for a multi-robot scenario. Thus, to rigorously build and test our system, our plan is to start with Gazebo-based simulations before moving to hardware experiments. We designed a simple Gazebo

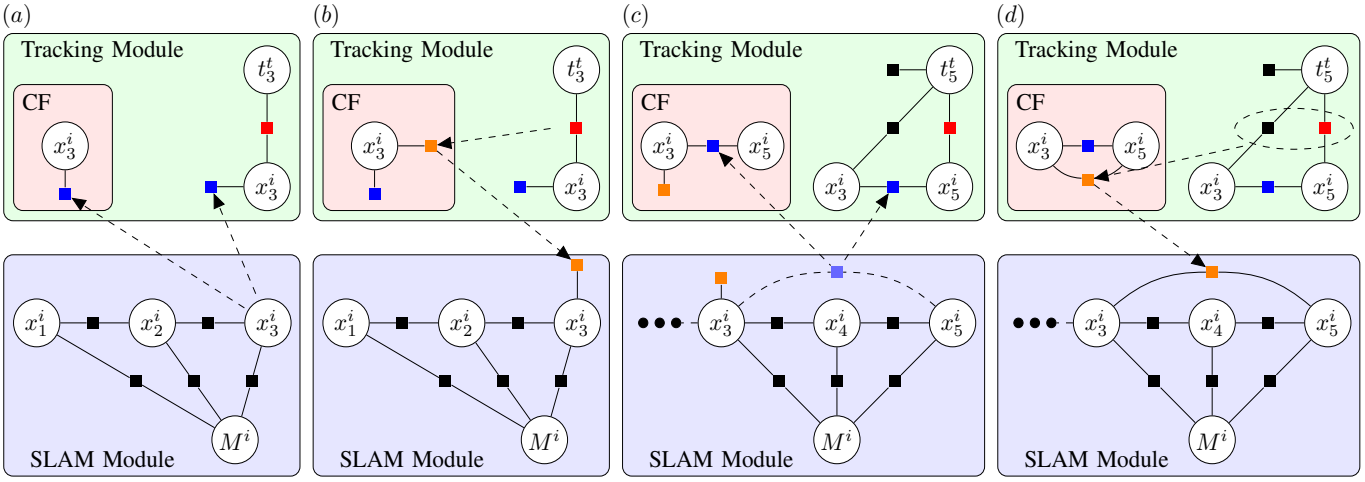

Fig. 2: Robot inner-loop architecture demonstrated with robot i. Each robot has a SLAM module (purple), a Tracking module (green), and a CF between the SLAM and tracking modules (pink). (a) and (c): When the robot measures the target, e.g., time steps 3 and 5, it requests and receives a pose estimate from the SLAM module (blue factors). (b) and (d): New data from the tracking module, due to target measurement or communication with neighboring robots, flows back to the SLAM module via the orange factors.

environment, including Clearpath Jackal unmanned ground vehicles (UGV) that can simulate the SLAM and tracking robots, as well as the targets. The preliminary study aims to test the architecture of the system, we will use LIO-SAM [13], a factor graph-based lidar-inertial odometry system, as the SLAM engine. When the tracking module requests a pose estimate from the SLAM module (see III), we query the LIO-SAM factor graph for the marginal distribution over the requested time steps, represented by the information vector and matrix. These are then integrated as a new factor to the tracking module factor graph and CF, as shown in Fig. 2(a) and (c). When the SLAM module receives a new pose factor back from the tracking algorithm, as shown in Fig. 2(b) and (d), it adds it to the LIO-SAM graph as a GPS or loop closure type factors [13].

When successful, the rest of the test plan is as follows: (i) Gazebo simulation with two different SLAM algorithms, e.g., LIO-SAM [13] - LiDAR-inertial, and Kimera [15] - visual-inertial. (ii) hardware experiments using 2-3 Clearpath Huskey UGVs and varying number of maneuvering and static targets in an outdoor environment.

V. Summary

The primary goal of this paper is to report progress toward a heterogeneous robotic system, where robots are able to collaborate despite differences in their onboard algorithms. We use factor graphs to analyze and exploit the conditional independence structure inherent in the decentralized multi-robot heterogeneous SLAM and tracking problem. We are then able to design a system architecture that separates the SLAM and tracking solutions of the robot and between the SLAM systems used by neighboring robots. While this paper focuses on a SLAM and tracking problem, this approach can be extended to other robotic applications such as cooperative localization and navigation [26], [27], and terrain height mapping [28].

References

[1] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004, conference Name: IEEE Transactions on Automatic Control.

[2] C. Y. Chong, E. Tse, and S. Mori, "Distributed estimation in networks," in *1983 American Control Conference (ACC)*, Jun. 1983, pp. 294–300.

[3] O. Dagan and N. R. Ahmed, "Exact and Approximate Heterogeneous Bayesian Decentralized Data Fusion," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1136–1150, Apr. 2023.

[4] B. J. Frey, F. R. Kschischang, H.-A. Loeliger, and N. Wiberg, "Factor graphs and algorithms," in *In Proc. 35th Allerton Conf. Communications, Control, and Computing*, 1997, pp. 666–680.

[5] O. Dagan and N. R. Ahmed, "Factor graphs for heterogeneous Bayesian decentralized data fusion," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, Nov. 2021, pp. 1–8.

[6] ——, "Conservative Filtering for Heterogeneous Decentralized Data Fusion in Dynamic Robotic Systems," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 5840–5847, iSSN: 2153-0866. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9981414

[7] E. Nettleton, S. Thrun, H. Durrant-Whyte, and S. Sukkarieh, "Decentralised SLAM with low-bandwidth communication for teams of vehicles," in *Field and Service Robotics*, vol. 24. Springer, 2006, pp. 179–188.

[8] S. Reece and S. Roberts, "Robust, low-bandwidth, multi-vehicle mapping," in *2005 7th International Conference on Information Fusion*, vol. 2, Jul. 2005, pp. 8 pp.–.

[9] S. J. Julier and J. K. Uhlmann, "Using covariance intersection for SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 3–20, Jan. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889006001436

[10] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: consistent distributed smoothing and mapping," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 5220–5227, iSSN: 1050-4729.

[11] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (ACC)*, vol. 4, Jun. 1997, pp. 2369–2373 vol.4.

[12] V. Tchuiev and V. Indelman, "Distributed Consistent Multi-Robot Semantic Localization and Mapping," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4649–4656, Jul. 2020, conference Name: IEEE Robotics and Automation Letters.

[13] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5135–5142, iSSN: 2153-0866.

[14] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, conference Name: IEEE Transactions on Robotics.

[15] 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," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, Dec. 2021. [Online]. Available: https://doi.org/10.1177/02783649211056674

[16] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[17] D. Van Opdenbosch and E. Steinbach, "Collaborative Visual SLAM Using Compressed Feature Exchange," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 57–64, Jan. 2019, conference Name: IEEE Robotics and Automation Letters.

[18] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, Aug. 2022.

[19] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 1, Sep. 2003, pp. 842–849 vol.1, iSSN: 1050-4729.

[20] D. Moratuwage, B.-N. Vo, and D. Wang, "Collaborative Multi-vehicle SLAM with moving object tracking," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5702–5708, iSSN: 1050-4729.

[21] I. Loefgren, "Scalable Navigation-Integrated Decentralized Data Fusion in Communication-Limited Environments," Master's thesis, University of Colorado at Boulder, United States – Colorado, 2020. [Online]. Available: https://www.proquest.com/docview/2411078422/abstract/F6EEA5BE4A58494APQ/1

[22] S. Grime and H. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, Oct. 1994.

[23] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2149–2154 vol.3.

[24] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364915620033

[25] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A Multi-sensor and Multi-scenario SLAM Dataset for Ground Robots," Dec. 2021, arXiv:2112.13659 [cs]. [Online]. Available: http://arxiv.org/abs/2112.13659

[26] S. J. Dourmashkin, N. R. Ahmed, D. M. Akos, and W. W. Whitacre, "GPS-limited cooperative localization using scalable approximate decentralized data fusion," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Apr. 2018, pp. 1473–1484.

[27] I. Loefgren, N. Ahmed, E. Frew, C. Heckman, and S. Humbert, "Scalable event-triggered data fusion for autonomous cooperative swarm localization," in *2019 22th International Conference on Information Fusion (FUSION)*, Jul. 2019, pp. 1–8.

[28] J. R. Schoenberg and M. Campbell, "Distributed terrain estimation using a mixture-model based algorithm," in *2009 12th International Conference on Information Fusion (FUSION)*, Jul. 2009, pp. 960–967.