Multi-Robot Object SLAM Using Distributed Variational Inference

Hanwen Cao, Sriram Shreedharan, Nikolay Atanasov

***Abstract*—Multi-robot simultaneous localization and mapping (SLAM) enables a robot team to achieve coordinated tasks by relying on a common map of the environment. Constructing a map by centralized processing of the robot observations is undesirable because it creates a single point of failure and requires pre-existing infrastructure and significant communication throughput. This paper formulates multi-robot object SLAM as a variational inference problem over a communication graph subject to consensus constraints on the object estimates maintained by different robots. To solve the problem, we develop a distributed mirror descent algorithm with regularization enforcing consensus among the communicating robots. Using Gaussian distributions in the algorithm, we also derive a distributed multi-state constraint Kalman filter (MSCKF) for multi-robot object SLAM. Experiments on real and simulated data show that our method improves the trajectory and object estimates, compared to individual-robot SLAM, while achieving better scaling to large robot teams, compared to centralized multi-robot SLAM.**

***Index Terms*—Multi-Robot SLAM, Distributed Robot Systems, Probability and Statistical Methods**

I. INTRODUCTION

SIMULTANEOUS localization and mapping (SLAM) [1] is a fundamental problem for enabling mobile robot to operate autonomously in unknown unstructured environments. In robotics applications, such as transportation, warehouse automation, and environmental monitoring, a team of collaborating robots can be more efficient than a single robot. However, effective coordination in robot teams requires a common frame of reference and a common understanding of the environment [2]. Traditionally, these requirements have been approached by relying on a central server or lead robot [3], [4], which communicates with other robots to receive sensor measurements and update the locations and map for the team. However, communication with a central server requires pre-existing infrastructure, introduces delays or potential estimation inconsistency, e.g., if the server loses track of synchronous data streams, and creates a single point of failure in the robot team. Hence, developing distributed techniques for multi-robot SLAM is an important and active research direction. A fully decentralized SLAM system enables robots to communicate opportunistically with connected peers in an ad-hoc network, removing the need for multi-hop communication protocols and centralized computation infrastructure. It allows flexible addition or removal of robots in the team and, by extending the scalability of the algorithm, enables coverage of larger areas with improved localization and map accuracy.

Manuscript received: May 1st, 2024; Revised: July 24th, 2024; Accepted: August 19th, 2024.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments.

We gratefully acknowledge support from NSF FRR CAREER 2045945 and ARL DCIST CRA W911NF-17-2-0181.

The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA, e-mails: {h1cao,sshreedharan,natanasov}@ucsd.edu.

Digital Object Identifier (DOI): see top of this page.

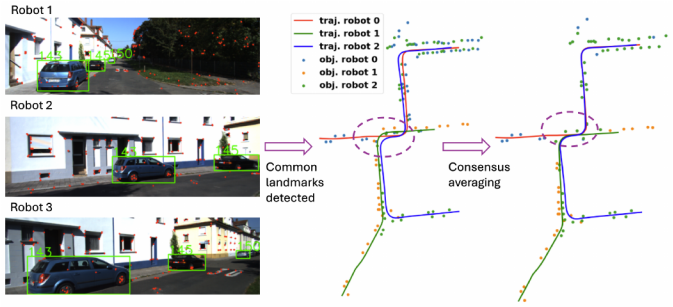

Fig. 1: Illustration of multi-robot object SLAM via distributed multi-state constraint Kalman filtering. The left images are inputs for the robots, where the red are geometric features (extracted by FAST [5]) and the green are object detections (by YOLOv6 [6]). The geometric features and object bounding box centroids are used as observations. When common objects are observed by communicating robots, a consensus averaging step is performed to align the estimated robot trajectories and object positions.

This paper considers a multi-robot landmark-based SLAM problem. We develop an approach for distributed Bayesian inference over a graph by formulating a mirror descent algorithm [7] in the space of probability density functions and introducing a regularization term that couples the estimates of neighboring nodes. Our formulation allows joint optimization of common variables (e.g., common landmarks among the robots) and local optimization of others (e.g., private robot trajectories). As a result, each node keeps a distribution only over its variables of interest, enabling both efficient storage and communication. By using Gaussian distributions in the mirror descent algorithm, we derive a distributed version of the widely used multi-state constraint Kalman filter (MSCKF) [8] with an additional averaging step to enforce consensus for the common variables. We apply our distributed MSCKF algorithm to collaborative object SLAM using only stereo camera observations at each robot. In the prediction step, each robot estimates its trajectory locally using visual odometry. In the update step, the robots correct their trajectory estimates using both visual features and object detections. As common in the MSCKF, we avoid keeping visual landmarks in the state using a null-space projection step. However, object landmarks are kept as a map representation for each robot and are shared among the robots during the consensus averaging step to collaboratively estimate consistent object maps. In short, each robot estimates its trajectory and an object map locally

but communicates with its neighbors to reach agreement on the object maps across the robots. Our contributions are summarized as follows.

- We formulate multi-robot landmark SLAM as a variational inference problem over a communication graph with a consensus constraint on the landmark variables.
- We develop a distributed mirror descent algorithm with a regularization term that couples the marginal densities of neighboring nodes.
- Using mirror descent with Gaussian distributions, we obtain a distributed version of the MSCKF algorithm.
- We demonstrate multi-robot object SLAM using stereo camera measurements for odometry and object detection on both real and simulated data to show that our method improves the overall accuracy of the trajectories and object maps of robot teams, compared to individual-robot SLAM, while achieving better scaling to large robot teams, compared to centralized multi-robot SLAM[1].

II. RELATED WORK

SLAM is a broad research area including a variety of estimation methods [8]–[10] as discussed in [1], [11]–[14]. Performing SLAM with multiple collaborating robots improves efficiency but also introduces challenges related to distributed storage, computation, and communication. This section reviews recent progress in multi-robot SLAM.

A. Multi-robot factor graph optimization

Factor graph methods formulate SLAM as an optimization problem over a bipartite graph of variables to be estimated and factors relating variables and measurements via error functions. Tian et al. [15] propose a certifiably correct pose graph optimization (PGO) method with a novel Riemannian block coordinate descent (RBCD) that operates in a distributed setting. Cunningham et al. [16], [17] extend smoothing and mapping (SAM) [18] by introducing a constrained factor graph that enforces consistent estimates of common landmarks among robots. Choudhary et al. [19] developed a two-stage approach using successive over-relaxation and Jacobi over-relaxation to split the computation among the robots. MR-iSAM2 [20] extends incremental smoothing and mapping (iSAM2) [10] by introducing a novel data structure called mult-root Bayes tree. Tian et al. [21] investigate the relation between Hessians of Riemannian optimization and Laplacians of weighted graphs and design a communication-efficient multi-robot optimization algorithm performing approximate second-order optimization.

Recent SLAM systems utilize the theoretical results of the above works to achieve efficient multi-robot operation. Kimera-Multi [22], a fully distributed dense metric-semantic SLAM system, uses a two-stage optimization method built upon graduated non-convexity [23] and RBCD [15]. DOOR-SLAM [24] uses [19] as a back-end and pairwise consistency maximization [25] for identifying consistent measurements across robots. Xu et al. [26] develop a distributed visual-inertial SLAM combining collaborative visual-inertial odometry with an alternating direction method of multipliers (ADMM) algorithm and asynchronous distributed pose graph optimization [27]. Andersson et al. [3] design a multi-robot SLAM system built upon square-root SAM [18] by utilizing rendezvous-measurements.

[1]Code is at https://github.com/ExistentialRobotics/distributed_msckf.

B. Multi-robot filtering

Filtering methods, such as the Kalman filter, offer a computationally lightweight alternative to factor graph optimization by performing incremental prediction and update steps that avoid a large number of iterations. Roumeliotis and Bekey [28] showed that the Kalman filter equations can be written in decentralized form, allowing decomposition into smaller communicating filters at each robot. Thrun et. al [29] presented a sparse extended information filter for multi-robot SLAM, which actively removes information to ensure sparseness at the cost of approximation. With nonlinear motion and observation models, a decentralized extended Kalman filter (EKF) has an observable subspace of higher dimension than the actual nonlinear system and generates unjustified covariance reduction [30]. Huang et al. introduced observability constraints in EKF [30] and unscented smoothing [31] algorithms to ensure consistent estimation. Gao et al. [32] use random finite sets to represent landmarks at each robot and maintain a probability hypothesis density (PHD). The authors prove that geometric averaging of the robot PHDs over one-hop neighbors leads to convergence of the PHDs to a global Kullback-Leibler average, ensuring consistent maps across the robots. Zhu et al. [33] propose a distributed visual-inertial cooperative localization algorithm by leveraging covariance intersection to compensate for unknown correlations among the robots and deal with loop-closure constraints.

Our contribution is to derive a fully distributed filter for object SLAM from a constrained variational inference perspective. Our formulation makes a novel connection to distributed mirror descent and enables robots to achieve landmark consensus efficiently with one-hop communication only and without sharing private trajectory information.

III. PROBLEM STATEMENT

Consider n robots seeking to collaboratively construct a model of their environment represented by a variable $\mathbf{y}$, e.g., a vector of landmark positions. Each robot i also aims to estimate its own state $\mathbf{x}_{i,t}$, e.g., pose, at time t. The combined state of robot i is denoted as $\mathbf{s}_{i,t} = [\mathbf{x}_{i,t}^\top \ \mathbf{y}^\top]^\top$ and evolves according to a known Markov motion model:

$$\mathbf{s}_{i,t+1} \sim f_i(\cdot \mid \mathbf{s}_{i,t}, \mathbf{u}_{i,t}), \tag{1}$$

where $\mathbf{u}_{i,t}$ is a control input and f_i is the probability density function (PDF) of the next state $\mathbf{s}_{i,t+1}$. Each robot receives observations $\mathbf{z}_{i,t}$ according to a known observation model:

$$\mathbf{z}_{i,t} \sim h_i(\cdot \mid \mathbf{s}_{i,t}), \tag{2}$$

where h_i is the observation PDF.

The robots communicate over a network represented as a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V} = \{1, \dots, n\}$ corresponding to the robots and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

specifying robot pairs that can exchange information, e.g. $(i,j) \in \mathcal{E}$ indicates that robot i and j can exchange information. Let $A \in \mathbb{R}^{n\times n}$ be a doubly stochastic weighted adjacency matrix of $\mathcal{G}$ such that $A_{ij} > 0$ if $(i,j) \in \mathcal{E}$ and $A_{ij} = 0$ otherwise. Also, let $\mathcal{N}_i := \{j \in \mathcal{V} | (j,i) \in \mathcal{E}\} \cup \{i\}$ denote the set of *one-hop neighbors* of robot i. and includes node i itself. We consider the following problem.

Problem 1. Given control inputs $\mathbf{u}_i = [\mathbf{u}_{i,0}^\top, \dots, \mathbf{u}_{i,T-1}^\top]^\top$ and observations $\mathbf{z}_i := [\mathbf{z}_{i,0}^\top, \dots, \mathbf{z}_{i,T}^\top]^\top$, each robot i aims to estimate the robot states $\mathbf{s}_i := [\mathbf{s}_{i,0}^\top, \dots, \mathbf{s}_{i,T}^\top]^\top$ collaboratively by exchanging information only with one-hop neighbors $j \in \mathcal{N}_i$ in the communication graph $\mathcal{G}$.

IV. Distributed Variational Inference

We approach the collaborative estimation problem using variational inference. We develop a distributed mirror descent algorithm to estimate a (variational) density of the states $\mathbf{s}_{i,t}$ with regularization that enforces consensus on the estimates of the common landmarks $\mathbf{y}$ among the robots.

A. Variational inference

As shown in [34], a Kalman filter/smoother can be derived by minimizing the Kullback-Leibler (KL) divergence between a variational density $q_i(\mathbf{s}_i)$ and the true Bayesian posterior $p_i(\mathbf{s}_i|\mathbf{u}_i, \mathbf{z}_i)$. Adopting a Bayesian perspective, the posterior is proportional to the joint density, which factorizes into products of motion and observation likelihoods due to the Markov assumptions in the models (1), (2):

$$\begin{aligned} p_i(\mathbf{s}_i|\mathbf{u}_i, \mathbf{z}_i) &\propto p_i(\mathbf{s}_i, \mathbf{u}_i, \mathbf{z}_i) \\ &\propto p_i(\mathbf{s}_{i,0}) \prod_{t=0}^{T-1} f_i(\mathbf{s}_{i,t+1}|\mathbf{s}_{i,t}, \mathbf{u}_{i,t}) \prod_{t=0}^{T} h_i(\mathbf{z}_{i,t}|\mathbf{s}_{i,t}). \end{aligned} \tag{3}$$

The KL divergence between the variational density $q_i(\mathbf{s}_i)$ and the true posterior $p_i(\mathbf{s}_i|\mathbf{u}_i, \mathbf{z}_i)$ can be decomposed as:

$$\begin{aligned} \mathrm{KL}(q_i||p_i) &= \mathbb{E}_{q_i}[-\log p_i(\mathbf{s}_i, \mathbf{u}_i, \mathbf{z}_i)] \\ &\quad - \underbrace{\mathbb{E}_{q_i}[-\log q_i(\mathbf{s}_i)]}_{\text{entropy}} + \underbrace{\log p_i(\mathbf{u}_i, \mathbf{z}_i)}_{\text{constant}}, \end{aligned} \tag{4}$$

where for simplicity of notation q_i without input arguments refers to $q_i(\mathbf{s}_i)$. Dropping the constant term, leads to the following optimization problem at robot i:

$$\min_{q_i \in \mathcal{Q}_i} c_i(q_i) := \mathbb{E}_{q_i}[-\log p_i(\mathbf{s}_i, \mathbf{z}_i, \mathbf{u}_i) + \log q_i(\mathbf{s}_i)], \tag{5}$$

where $\mathcal{Q}_i$ is a family of admissible variational densities.

B. Distributed mirror descent

We solve the variational inference problem in (5) using the mirror descent algorithm [7]. We use mirror descent because it includes an explicit (Bregman divergence) regularization term in the objective function that can incorporate information from one-hop neighbors [35]. This allows us to formulate a distributed version of mirror descent that enforces agreement among the landmark estimates of different robots with convergence guarantees. Mirror descent is a generalization of projected gradient descent that performs projection using a generalized distance (Bregman divergence), instead of the usual Euclidean distance, to respect the geometry of the constraint set $\mathcal{Q}_i$. Since $\mathcal{Q}_i$ is a space of PDFs, a suitable choice of Bregman divergence is the KL divergence. Starting with a prior PDF $q_i^{(0)}(\mathbf{s}_i)$, the mirror descent algorithm performs the following iterations:

$$q_i^{(k+1)} \in \arg\min_{q_i \in \mathcal{Q}_i} \mathbb{E}_{q_i}\left[\frac{\delta c_i}{\delta q_i}(q_i^{(k)})\right] + \frac{1}{\alpha_k}\mathrm{KL}(q_i||q_i^{(k)}), \tag{6}$$

where $\delta c_i / \delta q_i(q_i^{(k)})$ is the Fréchet derivative of $c_i(q_i)$ with respect to q_i evaluated at $q_i^{(k)}$ and $\alpha_k > 0$ is the step size.

Note that the optimizations (6) at each robot i are completely decoupled and, hence, each robot would be estimating its own density over the common landmarks $\mathbf{y}$. To make the estimation process collaborative, the regularization term $\mathrm{KL}(q_i||q_i^{(k)})$ in (6) should require that the PDF q_i of robot i is also similar to the priors $q_j^{(k)}$ of its neighbors $\mathcal{N}_i$ rather than its own prior $q_i^{(k)}$ alone. In our case, the PDFs $q_j^{(k)}(\mathbf{s}_j) = q_j^{(k)}(\mathbf{x}_j, \mathbf{y})$ are not defined over the same set of variables since each robot j is estimating its own private state $\mathbf{x}_j$ as well. Inspired by but different from [35], to enforce consensus only on the common state $\mathbf{y}$, the KL divergence term in (6) can be decomposed as a sum of marginal and conditional terms:

$$\begin{aligned} \mathrm{KL}(q_i(\mathbf{x}_i, \mathbf{y})||q_i^{(k)}(\mathbf{x}_i, \mathbf{y})) &= \mathrm{KL}(q_i(\mathbf{y})||q_i^{(k)}(\mathbf{y})) \\ &\quad + \mathrm{KL}(q_i(\mathbf{x}_i|\mathbf{y})||q_i^{(k)}(\mathbf{x}_i|\mathbf{y})). \end{aligned} \tag{7}$$

Hence, we can regularize only the marginal density $q_i(\mathbf{y})$ of the common environment state $\mathbf{y}$ to remain similar to the marginal densities $q_j^{(k)}(\mathbf{y})$ of the one-hop neighbors by using a weighted sum of KL divergences. This leads to the following optimization problem at robot i:

$$q_i^{(k+1)} \in \arg\min_{q_i \in \mathcal{Q}_i} g_i(q_i) \tag{8}$$

$$\begin{aligned} g_i(q_i) := \mathbb{E}_{q_i}\left[\frac{\delta c_i}{\delta q_i}(q_i^{(k)})\right] &+ \frac{1}{\alpha^{(k)}}\mathrm{KL}(q_i(\mathbf{x}_i|\mathbf{y})||q_i^{(k)}(\mathbf{x}_i|\mathbf{y})) \\ &+ \frac{1}{\alpha^{(k)}}\sum_{j\in\mathcal{N}_i} A_{ij}\,\mathrm{KL}(q_i(\mathbf{y})||q_j^{(k)}(\mathbf{y})), \end{aligned}$$

where $\mathcal{Q}_i = \{q_i \mid \int q_i = 1\}$ is the feasible set and A_{ij} are the elements of the adjacency matrix with $\sum_{j\in\mathcal{N}_i} A_{ij} = 1$. We derive a closed-form expression for the optimizer in the following proposition.

Proposition 1. *The optimizers of* (8) *satisfy:*

$$\begin{aligned} q_i^{(k+1)}(\mathbf{x}_i, \mathbf{y}) \propto [p_i(\mathbf{x}_i, \mathbf{y}, \mathbf{z}_i, \mathbf{u}_i)/q_i^{(k)}(\mathbf{x}_i, \mathbf{y})]^{\alpha_k} \\ q_i^{(k)}(\mathbf{x}_i|\mathbf{y}) \prod_{j\in\mathcal{N}_i} [q_j^{(k)}(\mathbf{y})]^{A_{ij}}. \end{aligned} \tag{9}$$

Proof. See [36, Appendix A]. □

C. Linear Gaussian case

In this section, we consider linear Gaussian models and obtain an explicit form of the distributed variational inference

update in (9). Suppose each robot i has the following motion and observation models:

$$\begin{aligned} \mathbf{s}_{i,t+1} &= F_i \mathbf{s}_{i,t} + G_i \mathbf{u}_{i,t} + \mathbf{w}_{i,t}, \quad & \mathbf{w}_{i,t} &\sim \mathcal{N}(\mathbf{0}, W_i), \\ \mathbf{z}_{i,t} &= H_i \mathbf{s}_{i,t} + \mathbf{v}_{i,t}, & \mathbf{v}_{i,t} &\sim \mathcal{N}(\mathbf{0}, V_i). \end{aligned} \tag{10}$$

Let the prior density of $\mathbf{s}_{i,0}$ be $\mathcal{N}(\boldsymbol{\mu}_{i,0}, \Sigma_{i,0})$ and considering all timesteps, we can write the models in lifted form as

$$\begin{aligned} \mathbf{s}_i &= \bar{F}_i(\bar{G}_i \mathbf{u}_i + \bar{\mathbf{w}}_i), \ \bar{\mathbf{w}}_i \sim \mathcal{N}(\mathbf{0}, \bar{W}_i), \\ \mathbf{z}_i &= \bar{H}_i \mathbf{s}_i + \bar{\mathbf{v}}_i, \ \mathbf{v}_i \sim \mathcal{N}(\mathbf{0}, \bar{V}_i), \end{aligned} \tag{11}$$

with the lifted terms defined as below

$$\begin{aligned} &\mathbf{s}_i = [\mathbf{s}_{i,0}^\top \ \cdots \ \mathbf{s}_{i,T}^\top]^\top, \ \ \mathbf{u}_i^\top = [\boldsymbol{\mu}_{i,0}^\top \ \mathbf{u}_{i,0} \ \cdots \ \mathbf{u}_{i,T}^\top]^\top, \\ &\mathbf{z}_i = [\mathbf{z}_{i,0}^\top \ \cdots \ \mathbf{z}_{i,T}^\top]^\top, \ \ \bar{H}_i = I_{T+1} \otimes H_i, \\ &\bar{V}_i = I_{T+1} \otimes V_i, \bar{W}_i = \begin{bmatrix} \Sigma_{i,0} & 0 \\ 0 & I_T \otimes W_i \end{bmatrix}, \\ &\bar{F}_i = \begin{bmatrix} I & 0 & \cdots & 0 \\ F & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ F^T & F^{T-1} & \cdots & I \end{bmatrix}, \bar{G}_i = \begin{bmatrix} I & 0 \\ 0 & I_T \otimes G_i \end{bmatrix}, \end{aligned} \tag{12}$$

where $\otimes$ is the Kronecker product. Denoting the density and distribution at iteration k as $q_i^{(k)}(\mathbf{s}_i)$ and $\mathcal{N}(\boldsymbol{\mu}_{i,(k)}, \Sigma_{i,(k)})$, the distributed variational inference update in (9) is computed in the following propositions.

Proposition 2. *Consider a joint Gaussian distribution*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}^{\mathbf{x}} \\ \boldsymbol{\mu}^{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \Sigma^{\mathbf{x}} & \Sigma^{\mathbf{xy}} \\ \Sigma^{\mathbf{xy}\top} & \Sigma^{\mathbf{y}} \end{bmatrix} \right). \tag{13}$$

If the marginal distribution over $\mathbf{y}$ *changes to* $\mathcal{N}(\bar{\boldsymbol{\mu}}^{\mathbf{y}}, \bar{\Sigma}^{\mathbf{y}})$*, the new joint distribution* $\mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\Sigma})$ *of* $(\mathbf{x}, \mathbf{y})$ *will be*

$$\begin{aligned} &\bar{\boldsymbol{\mu}} = \begin{bmatrix} A\bar{\boldsymbol{\mu}}^{\mathbf{y}} + \mathbf{b} \\ \bar{\boldsymbol{\mu}}^{\mathbf{y}} \end{bmatrix}, \ \bar{\Sigma} = \begin{bmatrix} A\bar{\Sigma}^{\mathbf{y}} A^\top + P & A\bar{\Sigma}^{\mathbf{y}} \\ \bar{\Sigma}^{\mathbf{y}} A^\top & \bar{\Sigma}^{\mathbf{y}} \end{bmatrix}, \\ &A = \Sigma^{\mathbf{xy}} \Sigma^{\mathbf{y}-1}, \qquad \mathbf{b} = \boldsymbol{\mu}^{\mathbf{x}} - \Sigma^{\mathbf{xy}} \Sigma^{\mathbf{y}-1} \boldsymbol{\mu}^{\mathbf{y}}, \\ &P = \Sigma^{\mathbf{x}} - \Sigma^{\mathbf{x}} \Sigma^{\mathbf{y}-1} \Sigma^{\mathbf{xy}\top}. \end{aligned} \tag{14}$$

Proposition 3. *In the linear Gaussian case, the distributed variational inference update in* (9) *can be obtained by first averaging marginal densities of the common state* $\mathbf{y}$ *across the neighbors* $\mathcal{N}_i$ *of robot* i*:*

$$\bar{\Sigma}_{i,(k)}^{\mathbf{y}-1} = \sum_{j \in \mathcal{N}_i} A_{ij} \Sigma_{j,(k)}^{\mathbf{y}-1}, \ \bar{\Sigma}_{i,(k)}^{\mathbf{y}-1} \bar{\boldsymbol{\mu}}_{i,(k)}^{\mathbf{y}} = \sum_{j \in \mathcal{N}_i} A_{ij} \Sigma_{j,(k)}^{\mathbf{y}-1} \boldsymbol{\mu}_{j,(k)}^{\mathbf{y}}, \tag{15}$$

then constructing a new joint distribution $\mathcal{N}(\bar{\boldsymbol{\mu}}_{i,(k)}, \bar{\Sigma}_{i,(k)})$ *according to Proposition 2, and finally updating the density using the motion and observation models in* (11)*:*

$$\begin{aligned} &\Sigma_{i,(k+1)}^{-1} = \bar{\Sigma}_{i,(k)}^{-1} + \alpha_k (\bar{F}_i^{-\top} \bar{W}_i^{-1} \bar{F}_i^{-1} + \bar{H}_i^\top \bar{V}_i^{-1} \bar{H}_i - \Sigma_{i,(k)}^{-1}), \\ &\Sigma_{i,(k+1)}^{-1} \boldsymbol{\mu}_{i,(k+1)} = \bar{\Sigma}_{i,(k)}^{-1} \bar{\boldsymbol{\mu}}_{i,(k)} \\ &\quad + \alpha_k (\bar{F}_i^{-\top} \bar{W}_i^{-1} \bar{G}_i \mathbf{u}_i + \bar{H}_i^\top \bar{V}_i^{-1} \mathbf{z}_i - \Sigma_{i,(k)}^{-1} \boldsymbol{\mu}_{i,(k)}). \end{aligned} \tag{16}$$

Proof. See [36, Appendix B]. □

In the above proposition, the averaging over the marginal densities in (15) comes from the term $\prod_{j \in \mathcal{N}_i} [q_j^{(k)}(\mathbf{y})]^{A_{ij}}$ in (9), which enforces consensus among the robots over the common variables $\mathbf{y}$. If the robots are in consensus, i.e., $\boldsymbol{\mu}_{i,(k)}^{\mathbf{y}} = \boldsymbol{\mu}_{j,(k)}^{\mathbf{y}}, \Sigma_{i,(k)}^{\mathbf{y}} = \Sigma_{j,(k)}^{\mathbf{y}}, \ \forall j \in \mathcal{N}_i$, (16) with $\alpha_k = 1$ converges in just one step,

$$\begin{aligned} \Sigma_i^{-1} &= \bar{F}_i^{-\top} \bar{W}_i^{-1} \bar{F}_i^{-1} + \bar{H}_i^\top \bar{V}_i^{-1} \bar{H}_i, \\ \Sigma_i^{-1} \boldsymbol{\mu}_i &= \bar{F}_i^{-\top} \bar{W}_i^{-1} \bar{G}_i \mathbf{u}_i + \bar{H}_i^\top \bar{V}_i^{-1} \mathbf{z}_i. \end{aligned} \tag{17}$$

As shown in [37, Ch. 3.3], considering only two consecutive time steps in the lifted form in (11) leads to a Kalman filter. Using the result in Proposition 3, we obtain a distributed Kalman filter that incorporates the consensus averaging step in (15). To allow correlation between the motion and measurement noise, we follow Crassidis and Junkins [38, Ch. 5] and obtain a correlated Kalman filter in [36, Appendix C].

V. Distributed MSCKF

In this section, we use Proposition 3 to derive a distributed version of the MSCKF algorithm [8], summarized in Algorithm 1. Each step of the algorithm is described in the following subsections.

Algorithm 1 Distributed Multi-State Constraint Kalman Filter

Input: Prior mean and covariance $(\boldsymbol{\mu}_{i,t-1}, \Sigma_{i,t-1})$, control input $\mathbf{u}_{i,t-1}$, and measurements $\mathbf{z}_{i,t}^g$, $\mathbf{z}_{i,t}^o$.
Output: Posterior mean and covariance $(\boldsymbol{\mu}_{i,t}, \Sigma_{i,t})$
1: **Consensus averaging**: (19), (21) in Sec. V-B
2: **State propagation**: (22), (23) in Sec. V-C
3: **State update**: (26), (28) in Sec. V-D
4: **Feature initialization**: (30) in Sec. V-E

A. State and observation description

The state $\mathbf{s}_{i,t}$ of robot i at time t contains a sequence of c historical camera poses $\mathbf{x}_{i,t}$ and a set of m_t landmarks $\mathbf{y}_{i,t}$:

$$\begin{aligned} \mathbf{s}_{i,t} &= (\mathbf{x}_{i,t}, \mathbf{y}_{i,t}), \\ \mathbf{x}_{i,t} &= (T_{i,t-c+1}, \ldots, T_{i,t}), \ \ T_{i,k} \in SE(3), \ \forall k, \\ \mathbf{y}_{i,t} &= [\mathbf{p}_{i,1}^\top \ \cdots \ \mathbf{p}_{i,m_t}^\top]^\top, \quad \mathbf{p}_{i,k} \in \mathbb{R}^3, \ \forall k. \end{aligned} \tag{18}$$

Besides the joint mean of the historical camera poses $\mathbf{x}_{i,t}$ and landmarks $\mathbf{y}_{i,t}$, each robot also keeps track a joint covariance $\Sigma_{i,t} \in \mathbb{R}^{(6c+3m_t)\times(6c+3m_t)}$. Each robot obtains observations $\mathbf{z}_{i,t}^o$ of persistent features, e.g., object detections, and observations $\mathbf{z}_{i,t}^g$ of opportunistic features, e.g., image keypoints or visual features, as illustrated in Fig. 1. We use point observations and the pinhole camera model for both feature types. Only the landmarks associated with persistent features are initialized and stored in the state while the landmarks associated with opportunistic features are used for structureless updates as in the MSCKF algorithm [8].

B. Consensus averaging

Each robot i communicates with its neighbors $\mathcal{N}_i$ to find out common landmarks. Then each robot sends the mean and covariance of the common landmarks $\boldsymbol{\mu}_{i,t-1}^{\mathbf{y}}, \Sigma_{i,t-1}^{\mathbf{y}}$ to its neighbors and receives $\boldsymbol{\mu}_{j,t-1}^{\mathbf{y}}, \Sigma_{j,t-1}^{\mathbf{y}}$, $j \in \mathcal{N}_i \backslash \{i\}$. The consensus averaging step is carried out by averaging the marginal distributions of the common landmarks:

$$\begin{aligned} \bar{\Sigma}_{i,t-1}^{\mathbf{y}-1} &= \sum_{j \in \mathcal{N}_i} A_{ij} \Sigma_{j,t-1}^{\mathbf{y}-1}, \\ \bar{\Sigma}_{i,t-1}^{\mathbf{y}-1} \bar{\boldsymbol{\mu}}_{i,t-1}^{\mathbf{y}} &= \sum_{j \in \mathcal{N}_i} A_{ij} \Sigma_{j,t-1}^{\mathbf{y}-1} \boldsymbol{\mu}_{j,t-1}^{\mathbf{y}}, \end{aligned} \tag{19}$$

which is the same as (15) except that we only consider one time step t here. Then, we need to reconstruct the new joint distribution $\mathcal{N}(\bar{\boldsymbol{\mu}}_{i,t-1}, \bar{\Sigma}_{i,t-1})$. Since we store the historical camera poses as $SE(3)$ matrices, Proposition 2 can not be applied directly. Following [37, Ch. 7.3.1], we define a Gaussian distribution over a historical camera pose $\underline{T}_{i,k},\ k = t-c, \cdots, t-1$ by adding a perturbation $\epsilon_{i,k}$:

$$\underline{T}_{i,k} = T_{i,k} \exp(\epsilon_{i,k}^{\wedge}),\ \epsilon_{i,k} \sim \mathcal{N}(\mathbf{0}_6, \Sigma_{i,t-1}^{\mathbf{x}_k}), \tag{20}$$

where $(\cdot)^{\wedge}$ defined in [37, Ch. 7.1.2] converts from $\mathbb{R}^6$ to a $\mathbb{R}^{4\times4}$ twist matrix. The estimated covariance $\Sigma_{i,t-1}$ already takes account of both the poses and landmarks. For consistency of notation with Proposition 2, we denote the mean of the pose perturbation as $\boldsymbol{\mu}_{i,t-1}^{\mathbf{x}} = \mathbf{0}_{6c}$. After averaging and reconstructing the new joint distribution, $\bar{\boldsymbol{\mu}}_{i,t-1}^{\mathbf{x}}$ may be non-zero, so we need to correct the camera poses as follows:

$$\bar{T}_{i,k} = T_{i,k} \exp(\bar{\boldsymbol{\mu}}_{i,t-1}^{\mathbf{x}_k \wedge}),\ \bar{\boldsymbol{\mu}}_{i,t-1}^{\mathbf{x}_k} \in \mathbb{R}^6, k = t-c, \cdots, t-1. \tag{21}$$

C. State propagation

We derive a general odometry propagation step for the MSCKF algorithm, thus not necessarily requiring IMU measurements and enabling vision-only propagation. We assume an odometry algorithm (e.g., libviso2 [39]) provides relative pose measurements $\delta T_{i,t-1}$ between the frame at time $t-1$ and that at time t. The state of robot i is propagated as:

$$\begin{aligned} \mathbf{x}_{i,t}^{+} &= (\bar{T}_{i,t-c+1}, \ldots, \bar{T}_{i,t-1}, \bar{T}_{i,t-1}\delta T_{i,t-1}), \\ \mathbf{s}_{i,t}^{+} &= (\mathbf{x}_{i,t}^{+}, \bar{\boldsymbol{\mu}}_{i,t-1}^{\mathbf{y}}), \end{aligned} \tag{22}$$

where the terms $\bar{(\cdot)}$ are obtained from the consensus averaging step. The state covariance is propagated as follows:

$$\begin{aligned} \Sigma_{i,t}^{+} &= \begin{bmatrix} A & 0 \\ J_t & 0 \\ 0 & I_{3m_{t-1}} \end{bmatrix} \bar{\Sigma}_{i,t-1} \begin{bmatrix} A & 0 \\ J_t & 0 \\ 0 & I_{3m_{t-1}} \end{bmatrix}^{\top} + \operatorname{diag}(\mathbf{e}_{6n}) \otimes W_i, \\ A &= \left[0_{6(c-1)\times 6} | I_{6(c-1)}\right], J_t = \left[0_{6\times 6(c-1)}\ Ad(\delta T_{t-1}^{-1})\right], \end{aligned} \tag{23}$$

where $Ad(\cdot)$ is the adjoint of an $SE(3)$ matrix [37, Chapter 7.1.4], $\mathbf{e}_{6n} \in \mathbb{R}^{6n+3m_t}$ is a vector with the $6n$-th element as 1 and the rest as 0, and $W_i \in \mathbb{R}^{6\times6}$ is the odometry measurement covariance.

D. State update

The MSCKF update step follows prior work [40]. The camera pose residual is a perturbation $\underline{\epsilon}_{i,k}$ that transforms the estimated pose $T_{i,k}$ to the true pose $\underline{T}_{i,k}$, i.e. $\underline{T}_{i,k} = T_{i,k} \exp(\underline{\epsilon}_{i,k}^{\wedge})$. The landmark residual is the difference between true position $\underline{\mathbf{p}}$ and the estimated position $\mathbf{p}$, i.e., $\tilde{\mathbf{p}} = \underline{\mathbf{p}} - \mathbf{p}$, where $\mathbf{p}$ is the position mean if the landmark is in the state or the result of triangulation if it is not. When robot i receives the k-th geometric feature observation at time t, denoted as $\mathbf{z}_{i,t,k}^{g}$, we linearize the observation model around the current error state $\tilde{\mathbf{s}}_{i,t,k}$ (composed of both pose and landmark residuals) and the feature position residual $\tilde{\mathbf{p}}_{i,k}^{g}$:

$$\mathbf{r}_{i,t,k}^{g} = \mathbf{z}_{i,t,k}^{g} - \hat{\mathbf{z}}_{i,t,k}^{g} = H_{i,t,k}^{\mathbf{s},g} \tilde{\mathbf{s}}_{i,t,k} + H_{i,t,k}^{\mathbf{p},g} \tilde{\mathbf{p}}_{i,k}^{g} + \mathbf{v}_{i,t,k}^{g},$$

where $\hat{\mathbf{z}}_{i,t,k}^{g}$ is the predicted observation, $H_{i,t,k}^{\mathbf{s},g}$ and $H_{i,t,k}^{\mathbf{p},g}$ are Jacobians, and $\mathbf{v}_{i,t,k}^{g} \sim \mathcal{N}(\mathbf{0}, V_i^{g})$ is the geometric observation noise. Then, we left-multiply by the nullspace $N_{i,t,k}$ of $H_{i,t,k}^{\mathbf{p},g}$ to remove the effect of $\tilde{\mathbf{p}}_{i,k}^{g}$:

$$\mathbf{r}_{i,t,k}^{g,0} = N_{i,t,k}^{\top} \mathbf{r}_{i,t,k}^{g} = N_{i,t,k}^{\top} H_{i,t,k}^{\mathbf{s},g} \tilde{\mathbf{s}}_{i,t,k} + N_{i,t,k}^{\top} \mathbf{v}_{i,t,k}^{g}. \tag{24}$$

Since we allow general odometry in the propagation step, potentially obtained from visual features, there may be correlation between the motion noise and the observation noise. The correlation is denoted as

$$S_{i,t,k} = \mathbb{E}[\mathbf{w}_{i,t-1} \mathbf{v}_{i,t,k}^{\top}],\ \ \mathbf{w}_{i,t-1} \sim \mathcal{N}(\mathbf{0}, W_i). \tag{25}$$

Concatenating $\mathbf{r}_{i,t,k}^{g,0}$, $N_{i,t,k}^{\top} H_{i,t,k}^{\mathbf{s},g}$, $N_{i,t,k}^{\top} V_i^{g} N_{i,t,k}$, and $S_{i,t,k} N_{i,t,k}$ for all k appropriately, we get an overall geometric feature residual $\mathbf{r}_{i,t}^{g}$, Jacobian $H_{i,t}^{g}$, noise covariance $V_{i,t}^{g}$ and correlation $S_{i,t}$. Similarly, we linearize the observation model for the object $\mathbf{z}_{i,t,k}^{o}$:

$$\mathbf{r}_{i,t,k}^{o} = \mathbf{z}_{i,t,k}^{o} - \hat{\mathbf{z}}_{i,t,k}^{o} = H_{i,t,k}^{\mathbf{s},o} \tilde{\mathbf{s}}_{i,t} + \mathbf{v}_{i,t,k}^{o},\ \mathbf{v}_{i,t,k}^{o} \sim \mathcal{N}(\mathbf{0}, V_i^{o}),$$

and concatenate $\mathbf{r}_{i,t,k}^{o}$, $H_{i,t,k}^{\mathbf{s},o}$ and V_i^{o} for all k to get an overall object observation residual $\mathbf{r}_{i,t}^{o}$, Jacobian $H_{i,t}^{o}$, and noise covariance $V_{i,t}^{o}$.

Finally, by concatenating the residuals and Jacobians of both geometric and object features, we get the overall residual $\mathbf{r}_{i,t} = [\mathbf{r}_{i,t}^{g\top}\ \mathbf{r}_{i,t}^{o\top}]^{\top}$ and Jacobian $H_{i,t} = [H_{i,t}^{g\top}\ H_{i,t}^{o\top}]^{\top}$. As shown in [36, Appendix C], [38, Ch. 5], the Kalman gain is:

$$K_{i,t} = (\Sigma_{i,t}^{+} H_{i,t}^{\top} + S_{i,t})(H_{i,t} \Sigma_{i,t}^{+} H_{i,t}^{\top} + V_{i,t})^{-1}, \tag{26}$$

$$V_{i,t} = \operatorname{blkdiag}(V_{i,t}^{g} + H_{i,t}^{g} S_{i,t} + S_{i,t}^{\top} H_{i,t}^{g\top}, V_{i,t}^{o}), \tag{27}$$

where $S_{i,t}$ only appears for the geometric features, $\Sigma_{i,t}^{+}$ is from the prediction step, and $K_{i,t}$ can be split to $K_{i,t}^{\mathbf{x}}$ and $K_{i,t}^{\mathbf{y}}$ related to $\mathbf{x}_{i,t}$ and $\mathbf{y}_{i,t}$ respectively. The landmark mean, camera poses, and the entire covariance are updated as:

$$\begin{aligned} \boldsymbol{\mu}_{i,t}^{\mathbf{y}} &= \boldsymbol{\mu}_{i,t}^{\mathbf{y}+} + K_{i,t}^{\mathbf{y}} \mathbf{r}_{i,t}, \\ T_{i,k} &= T_{i,k}^{+} \exp(K_{i,t}^{\mathbf{x}_k} \mathbf{r}_{i,t}),\ K_{i,t}^{\mathbf{x}_k} \mathbf{r}_{i,t} \in \mathbb{R}^6,\ k = t-c+1, \cdots, t, \\ \Sigma_{i,t} &= (I - K_{i,t} H_{i,t}) \Sigma_{i,t}^{+} \end{aligned} \tag{28}$$

where the terms $(\cdot)^{+}$ are from the propagation step.

E. Feature initialization

The feature initialization is the same as [41]. To initialize an object landmark, we first linearize the observation model

$$\tilde{\mathbf{z}}_{i,t,k}^{o} = H_{i,t,k}^{\mathbf{s}} \tilde{\mathbf{s}}_{i,t} + H_{i,t,k}^{\mathbf{p}} \tilde{\mathbf{p}}_{i,k}^{o} + \mathbf{v}_{i,t,k}^{o},\ \mathbf{v}_{i,t,k}^{o} \sim \mathcal{N}(\mathbf{0}, V_i^{o}),$$

where $\tilde{\mathbf{z}}_{i,t,k}^{o}$, $\tilde{\mathbf{s}}_{i,t}$ and $\tilde{\mathbf{p}}_{i,k}^{o}$ are the residuals of the observation, current state, and new landmark respectively. Then, QR decomposition is performed to separate the linearized observation model into two parts: one that depends on the new landmark and another that does not:

$$\begin{bmatrix} \tilde{\mathbf{z}}_{i,t,k}^{o,1} \\ \tilde{\mathbf{z}}_{i,t,k}^{o,2} \end{bmatrix} = \begin{bmatrix} H_{i,t,k}^{\mathbf{s},1} & H_{i,t,k}^{\mathbf{p},1} \\ H_{i,t,k}^{\mathbf{s},2} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{s}}_{i,t} \\ \tilde{\mathbf{p}}_{i,k}^{o} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{i,t,k}^{o,1} \\ \mathbf{v}_{i,t,k}^{o,2} \end{bmatrix}. \tag{29}$$

Thus, we can augment the current state and covariance:

$$\begin{aligned} \mathbf{p}_{i,k}^{o} &= \hat{\mathbf{p}}_{i,k}^{o} + H_{i,t,k}^{\mathbf{s},1-1} \tilde{\mathbf{z}}_{i,t,k}^{o,1}, \\ \Sigma_{i,t,k}^{\mathbf{sp}} &= -\Sigma_{i,t} H_{i,t,k}^{\mathbf{s},1\top} H_{i,t,k}^{\mathbf{p},1-\top}, \\ \Sigma_{i,t,k}^{\mathbf{p}} &= H_{i,t,k}^{\mathbf{p},1-1} (H_{i,t,k}^{\mathbf{s},1} \Sigma_{i,t} H_{i,t,k}^{\mathbf{s},1\top} + V_i^{o,1}) H_{i,t,k}^{\mathbf{p},1-\top}, \end{aligned} \tag{30}$$

where $V_i^{o,1}$ is the covariance of noise $\mathbf{v}_{i,t,k}^{o,1}$, $\Sigma_{i,t,k}^{\mathbf{sp}}$ is the cross-correlation term between the current state and new landmark, and $\Sigma_{i,t,k}^{\mathbf{p}}$ is the covariance of the new landmark.

VI. Evaluation

We implemented the distributed MSCKF using only stereo camera observations and evaluated it on the KITTI dataset [42] and on a simulated dataset with a larger number of robots. All experiments were carried out on a laptop with i9-11980HK@2.60 CPU, 16 GB RAM, and RTX 3080 GPU.

A. KITTI dataset

The KITTI dataset [42] is an autonomous driving dataset that provides stereo images, LiDAR point clouds, and annotated ground-truth robot trajectories. We provide details about the data processing and evaluation results below.

1) Sequences and splits: We chose long sequences in the KITTI odometry dataset with loop closures and a sufficient number of cars, used as object landmarks, namely, sequences 00, 05, 06, and 08. Each sequence is split into 3 sub-sequences representing 3 different robots. The sequence splits are as follows: sequence 00: $[0, 2000]$, $[1500, 3500]$, $[2500, 4540]$; sequence 05: $[0, 1200]$, $[800, 2000]$, $[1560, 2760]$; sequence 06: $[0, 700]$, $[200, 900]$, $[400, 1100]$; sequence 08: $[0, 2000]$, $[1000, 3000]$, $[2000, 4070]$. We used a fully connected graph and the adjacency matrix $A \in \mathbb{R}^{3\times3}$ has all elements as $\frac{1}{3}$.

2) Geometric features: We extract geometric features using the FAST corner detector [5]. The KLT optical flow algorithm [43] is used to track the features across stereo images. Outlier rejection is performed using 2-point RANSAC for temporal tracking and the known essential matrix for stereo matching. Finally, circular matching similar to [44] is performed to further remove outliers.

3) Object features: We utilize YOLOv6 [6] to detect object bounding boxes and compute the centers as our object observations. Since our work does not focus on object tracking, we directly use the instance ID annotations in SemanticKITTI [45] for data association. The instance annotations are provided for LiDAR point clouds and we associate them with the bounding boxes by projecting the LiDAR point clouds onto the image plane and checking the dominant instance points inside each bounding box.

4) Odometry: The relative pose $\delta T_{i,t}$ between consecutive camera frames is obtained by libviso2 [39].

5) Results and analysis: We found empirically that setting the correlation matrix (25) to zero gives the best results. We assume that this is because libviso2 [39] uses SURF features [46], while the update step is performed using FAST features [5] and the correlation is negligible. Qualitative results from three-robot collaborative object SLAM on the KITTI dataset are shown in Fig. 2a. We show the root mean square error (RMSE) of the robot trajectory estimates in Table I and the mean distances between estimated object positions and the ground truth in Table II. We do not use alignment for the trajectory RMSE [47] because trajectory transformations affect the object mapping errors. Some ways to mitigate the effect of bad estimates include resilient consensus [48] or adaptive

TABLE I: Trajectory RMSE in meters on KITTI sequences. Separate and consensus correspond to without/with the consensus averaging step in Sec. V-B.

	Robot 1	Robot 2	Robot 3	Avg	Max
libviso2 [39]	14.30	13.73	12.65	13.56	14.30
00 Separate	**12.47**	7.55	12.42	10.81	**12.47**
00 Consensus	12.51	**7.13**	**8.73**	**9.45**	12.51
05 libviso2 [39]	5.36	6.42	11.57	7.78	11.57
05 Separate	7.18	10.03	**7.87**	8.36	10.03
05 Consensus	**4.69**	**7.75**	9.56	**7.33**	**9.56**
06 libviso2 [39]	5.45	6.89	5.21	5.85	6.89
06 Separate	4.23	**5.60**	4.86	4.90	**5.60**
06 Consensus	**4.23**	5.61	**4.76**	**4.87**	5.61
08 libviso2 [39]	9.17	21.05	11.37	13.86	21.05
08 Separate	15.08	24.28	9.18	16.18	24.28
08 Consensus	**13.89**	**12.71**	**9.18**	**11.93**	**13.89**

TABLE II: Object estimation errors in meters on KITTI sequences. Separate and consensus correspond to without/with the consensus averaging step in Sec. V-B.

	Robot 1	Robot 2	Robot 3	Avg	Max
00 Separate	**8.76**	7.61	8.70	8.36	**8.76**
00 Consensus	9.30	**6.74**	**7.16**	**7.73**	9.30
05 Separate	6.08	8.40	6.92	7.14	**8.40**
05 Consensus	**4.56**	**7.51**	8.54	**6.87**	8.54
06 Separate	**3.43**	5.92	4.64	4.66	5.92
06 Consensus	3.78	**5.63**	**4.37**	**4.59**	**5.63**
08 Separate	12.14	21.91	**8.19**	14.08	21.91
08 Consensus	**12.11**	**13.71**	9.21	**11.68**	**13.71**

TABLE III: Object position differences in meters across different robots on KITTI sequences. Separate and consensus correspond to without/with the consensus averaging step in Sec. V-B.

	Robot 1	Robot 2	Robot 3	Avg	Max
00 Separate	9.69	10.35	8.35	9.46	10.35
00 Consensus	**5.95**	**8.62**	**5.11**	**6.56**	**8.62**
05 Separate	7.25	10.20	15.74	11.06	15.74
05 Consensus	**1.50**	**5.15**	**10.17**	**5.61**	**10.17**
06 Separate	5.56	4.97	4.79	5.12	5.56
06 Consensus	**5.01**	**4.68**	**4.43**	**4.71**	**5.01**
08 Separate	14.61	20.24	23.81	19.55	23.81
08 Consensus	**6.50**	**9.48**	**11.36**	**9.12**	**11.36**

adjacency weights A_{ij} depending on the robots' measurement accuracy. Although consensus averaging can harm the estimation accuracy for some robots compared to running individual MSCKF algorithms for each robot, it helps improve the overall team performance in both localization and object mapping. The separate MSCKF sometimes perform worse than libviso2 [39]. This is because the object observation is too noisy. Updating with only object features can give an error up to 10 times as in Table I. The distributed MSCKF achieves better agreement in the map estimates among the robots. We compare the object position differences with and without averaging in Table III to quantify the reduction in disagreement. We also claim that the consensus averaging step does not add much time overhead because the robots communicate only common landmarks, meaning that the corresponding covariance $\Sigma_{i,t}^{\mathbf{y}}$ in (19) is small, and only perform averaging with one-hop neighbors. The computation time used by different components in the algorithm is shown in Fig. 3. Consensus averaging takes a small portion of time compared with the MSCKF update.

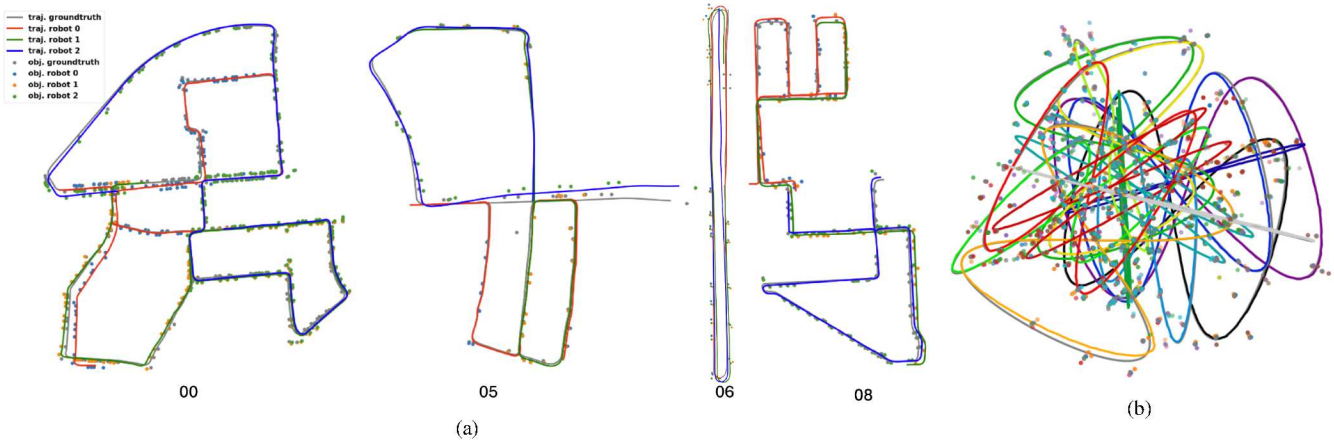

Fig. 2: Trajectory and object estimates of (a) 3 robots on KITTI sequences 00, 05, 06 and 08, (b) 15 robots in simulation.

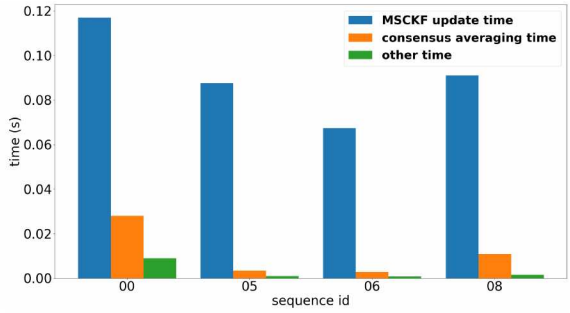

Fig. 3: Time consumed by different components per robot per frame, including MSCKF update, consensus averaging, and other (prediction and landmark initialization).

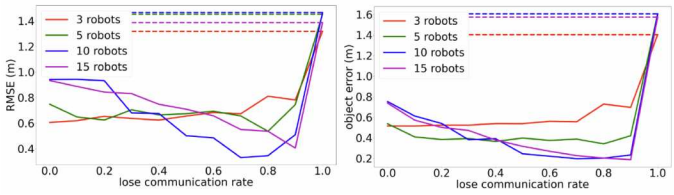

Fig. 4: Analysis of the effect of the robot network connectivity. The dashed lines show RMSE without averaging.

B. Simulated data

To test our algorithm with increasing numbers of robots, we generated simulated data for 3-15 robots.

1) Data generation: Each robot moves along a Lissajous curve and odometry measurements are generated by adding perturbations to the relative transformation between consecutive poses. The landmarks, both geometric and objects, are generated by randomly sampling from Gaussian distributions centered at each trajectory point. There are 210 objects in the scene with different numbers of robots. The observations are then generated by projecting the corresponding landmarks onto the image plane and adding noise.

2) Results: The results are visualized in Fig. 2b. The quantitative results from the simulations with a fully connected graph are shown in Table IV. Our algorithm scales efficiently with an increasing number of robots while continuing to outperform decoupled MSCKF algorithms for each robot. We also analyze the effect of connectivity in Fig. 4. In the experiment,

TABLE IV: Trajectory, object errors in meters and consensus averaging time per robot per timestep in seconds in simulation with different numbers of robots. Separate and consensus correspond to without/with the consensus averaging step in Sec. V-B.

Number of robots	3	5	10	15
Separate trajectory RMSE (m)	1.318	1.452	1.464	1.385
Consensus trajectory RMSE (m)	**0.605**	**0.748**	**0.941**	**0.933**
Separate object error (m)	1.402	1.604	1.605	1.573
Consensus object error (m)	**0.514**	**0.536**	**0.752**	**0.737**
Consensus averaging time (s)	0.021	0.022	0.026	0.028

each robot loses communication with each neighbor according to rate r, i.e., each edge in the graph at each time step is removed independently with probability r. We see that our algorithm is robust to communication loss rate up to $r = 0.9$. With small numbers of robots (3 and 5), the errors oscillate as the loss rate increases but with relatively large numbers of robots (10 and 15), the errors seem to decrease as the loss rate increases. Analyzing the effect of randomly connected graphs, e.g., broad gossip [49], will be considered in future work.

VII. CONCLUSION

We developed a distributed vision-only filtering approach for multi-robot object SLAM. Our experiments demonstrate that the method improves both localization and mapping accuracy while achieving agreement among the robots on a common object map. Since the algorithm is fully distributed, it allows efficient scaling of the number of robots in the team. Having a common object map is useful for collaborative task planning, which we plan to explore in future work.

ACKNOWLEDGMENTS

The authors are grateful to Shubham Kumar, Shrey Kansal, and Kishore Nukala from University of California San Diego for technical discussions and help with dataset preparation.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] J. Kinnari, A. Thomas, P. Lusk, K. Kondo, and J. P. How, "SOS-SLAM: Segmentation for Open-Set SLAM in Unstructured Environments," *arXiv preprint arXiv:2401.04791*, 2024.

[3] L. A. A. Andersson and J. Nygards, "C-SAM: Multi-robot slam using square root information smoothing," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2798–2805, 2008.

[4] I. Deutsch, M. Liu, and R. Siegwart, "A framework for multi-robot pose graph SLAM," in *IEEE International Conference on Real-time Computing and Robotics*, pp. 567–572, 2016.

[5] M. Trajković and M. Hedley, "Fast corner detection," *Image and vision computing*, vol. 16, no. 2, pp. 75–87, 1998.

[6] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, *et al.*, "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[7] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003.

[8] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *IEEE International Conference on Robotics and Automation*, pp. 3565–3572, 2007.

[9] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Transactions on Robotics (T-RO)*, vol. 24, no. 6, pp. 1365–1378, 2008.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 2, pp. 216–235, 2012.

[11] G. Huang, "Visual-inertial navigation: A concise review," in *International Conference on Robotics and Automation (ICRA)*, pp. 9572–9582, 2019.

[12] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2502–2509, 2018.

[13] S. Mokssit, D. B. Licea, B. Guermah, and M. Ghogho, "Deep Learning Techniques for Visual SLAM: A Survey," *IEEE Access*, vol. 11, pp. 20026–20050, 2023.

[14] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual SLAM," *Expert Systems with Applications*, vol. 205, p. 117734, 2022.

[15] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2137–2156, 2021.

[16] A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully distributed slam using constrained factor graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3025–3030, 2010.

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

[18] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[19] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 12, pp. 1286–1311, 2017.

[20] Y. Zhang, M. Hsiao, J. Dong, J. Engel, and F. Dellaert, "Mr-isam2: Incremental smoothing and mapping with multi-root bayes tree for multi-robot slam," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8671–8678, 2021.

[21] Y. Tian and J. P. How, "Spectral sparsification for communication-efficient collaborative rotation and translation estimation," *IEEE Transactions on Robotics*, vol. 40, pp. 257–276, 2024.

[22] Y. Tian, Y. Chang, F. H. 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 (TRO)*, vol. 38, no. 4, 2022.

[23] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[24] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "Door-SLAM: Distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.

[25] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2916–2923, 2018.

[26] H. Xu, P. Liu, X. Chen, and S. Shen, "D^2SLAM: Decentralized and Distributed Collaborative Visual-inertial SLAM System for Aerial Swarm," *arXiv preprint:2211.01538*, 2022.

[27] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: An Algorithmic Framework for Asynchronous Parallel Coordinate Updates," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. A2851–A2879, 2016.

[28] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[29] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *The International Journal of Robotics Research (IJRR)*, vol. 23, no. 7-8, pp. 693–716, 2004.

[30] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "On the consistency of multi-robot cooperative localization," in *Robotics: Science and Systems (RSS)*, 2009.

[31] G. Huang, M. Kaess, and J. J. Leonard, "Consistent unscented incremental smoothing for multi-robot cooperative target tracking," *Robotics and Autonomous Systems*, vol. 69, pp. 52–67, 2015.

[32] L. Gao, G. Battistelli, and L. Chisci, "Random-Finite-Set-Based Distributed Multirobot SLAM," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1758–1777, 2020.

[33] P. Zhu, P. Geneva, W. Ren, and G. Huang, "Distributed visual-inertial cooperative localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8714–8721, 2021.

[34] T. D. Barfoot, J. R. Forbes, and D. J. Yoon, "Exactly sparse gaussian variational inference with application to derivative-free batch nonlinear state estimation," *The International Journal of Robotics Research*, vol. 39, no. 13, pp. 1473–1502, 2020.

[35] P. Paritosh, N. Atanasov, and S. Martinez, "Distributed Bayesian estimation of continuous variables over time-varying directed networks," *IEEE Control Systems Letters*, vol. 6, pp. 2545–2550, 2022.

[36] H. Cao, S. Shreedharan, and N. Atanasov, "Multi-Robot Object SLAM using Distributed Variational Inference," *arXiv preprint arXiv:2404.18331*, 2024.

[37] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[38] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. Chapman and Hall/CRC, 2004.

[39] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[40] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.

[41] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4666–4672, 2020.

[42] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012.

[43] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, pp. 674–679, 1981.

[44] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *IEEE Intelligent Vehicles Symposium*, pp. 486–492, 2010.

[45] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of lidar sequences," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9297–9307, 2019.

[46] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*, pp. 404–417, Springer, 2006.

[47] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[48] D. Saldaña, A. Prorok, S. Sundaram, M. F. M. Campos, and V. Kumar, "Resilient consensus for time-varying networks of dynamic agents," in *American Control Conference (ACC)*, pp. 252–258, 2017.

[49] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal processing*, vol. 57, no. 7, pp. 2748–2761, 2009.