

Cooperative swarm localization and mapping with inter-agent ranging

Young-Hee Lee¹, Chen Zhu², Gabriele Giorgi³, and Christoph Günther⁴

Abstract—Compared to a single robot, a swarm system can conduct a given task in a shorter time, and it is more robust to system failures of each agent. To successfully execute cooperative missions with multiple agents, accurate relative positioning is important. If global positioning (e.g. with a GNSS-based positioning) is available, we can easily compute relative positions. In environments where a global positioning system is unreliable or unavailable, visual odometry can be applied for estimating each agent’s egomotion, by exploiting onboard cameras. Using these self-localization results, relative positions between agents can be estimated, once the relative geometry between agents is initialized. However, since visual odometry is a dead-reckoning process, the estimation errors accumulate inherently without bounds. We propose a cooperative localization method using visual odometry and inter-agent range measurements. Using the proposed method, we can reduce the drifts in position estimates with very modest requirements on the communication channel between agents.

I. INTRODUCTION

To successfully execute missions with a swarm system, accurate relative positioning is important to avoid collision between agents and properly merge the data acquired by each platform. In [1] and [2] Schumuck *et al.* propose centralized collaborative simultaneous localization and mapping (SLAM) with inter-agent loop closures to reduce inherent drifts in relative positioning. Lie *et al.* [3] propose a similar method, extending the state-of-the-art SLAM algorithm for a single agent, ORB-SLAM2 [4]. Additionally, Mohanarajah *et al.* [5] propose centralized cooperative SLAM with inter-agent map point matches using a cloud server instead of using a central server computer. Although the drifts in relative positioning can effectively corrected exploiting map point matches between agents, these methods might not be suitable for environments where data communication is constrained (i.e. communication is only available with low bandwidth and data rate) since large-sized data, which include feature descriptors of each map points, need to be exchanged between agents and a server computer (or cloud).

In order to reduce the data packet size, Cuning *et al.* [6], [7] propose multi-agent SLAM with a decentralized graph-based data fusion, by applying constraints to the factor graph (DDF-SAM). In [8] Cieslewski *et al.* propose

a fully decentralized visual SLAM with compact feature descriptors generated with NetVLAD [9], improving their previous works in [10] and [11]. Even though the data packet size can be reduced with the proposed methods, the inherent problem of using matching map points remains: robots’ movements are constrained since useful measurements for fusing local maps are only available when agents observe the same features in the explored environments, adding a constraint on the mission planning.

Instead of using inter-agent map point matches, Richardson *et al.* [12] use a known pattern for a multi-UAV system: whenever a UAV detects a known pattern (e.g. checkerboard) on another platform, the relative pose between the agents are corrected using this pattern information. Kim *et al.* [13] propose a graph-based cooperative SLAM using both map point matches and patterns for a swarm system with heterogeneous platforms, as a multi-agent version of iSAM [14]. However, as the approaches with inter-agent map point matches, the movement of agents is still needed to be constrained to observe a pattern on another robot.

We propose a collaborative SLAM method using a combination of visual odometry and range measurements between agents to mitigate relative positioning errors. Unlike the methods exploiting inter-agent map point matches and/or pattern observations, inter-agent ranges can be obtained with time-of-flight measurements even when agents are distributed widely, and observe different features. In addition, our method requires to only exchange poses of one of the agents (6DoF) and range measurements (scalar values), which reduces the data packet size significantly compared to methods using map fusion. We evaluate the proposed algorithm with an existing image dataset and synthetically generated inter-agent range measurements between two platforms. The proposed method could be also applied to a swarm system with an arbitrary robots as well.

II. SYSTEM SETUP

Fig. 1 shows the setup of the proposed collaborative SLAM with two agents. Each agent- k has a monocular camera C_k^t and a ranging tag module T_k^t as on-board sensors. The position vector from a camera to a tag is \mathbf{p}_{CT} , which can be calibrated before conducting the mission. The local reference frame of the agent- k is denoted with L_k whose origin coincides with its on-board camera frame.

To describe each agent’s camera pose with respect to its (static) local frame L_k , we use 3×4 matrices of the SE(3) Lie group [15]. The matrices in this group are denoted with $\mathbf{T}_{C_k L_k}^t = \begin{bmatrix} \mathbf{R}_{C_k L_k}^t & \mathbf{t}_{L_k C_k}^t \\ 0 & 1 \end{bmatrix}$, where $\mathbf{R}_{C_k L_k}^t$ is a rotation matrix

¹Young-Hee Lee is with the Institute for Communications and Navigation, Technical University of Munich, Germany younghee.lee@tum.de

²Chen Zhu and ³Gabriele Giorgi are with the Institute of Communications and Navigation, German Aerospace Center (DLR), Oberpfaffenhofen, Germany chen.zhu@dlr.de, gabriele.giorgi@dlr.de

⁴Christoph Günther is with the Institute for Communications and Navigation, Technical University of Munich and Institute of Communications and Navigation, German Aerospace Center (DLR) christoph.guenther@dlr.de

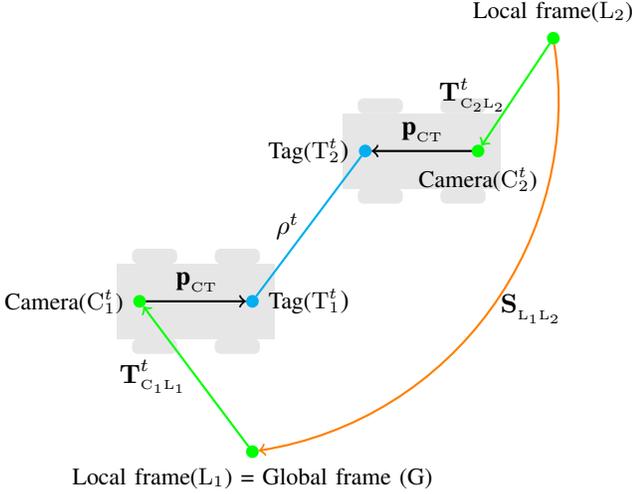


Fig. 1: The system setup with two agents. Each agent has a monocular camera C_k^t and a ranging tag module T_k^t . The local reference frame of the agent- k L_k is defined for each agent's egomotion estimation. The agent-1's local frame L_k is assumed to be the global frame G of the entire system. The agent- k 's camera pose at time t with respect to its local frame is denoted $\mathbf{T}_{C_k L_k}^t$, and \mathbf{p}_{CT} is the position vector from the camera to the tag at time t . The inter-agent range measurement at time t is denoted with ρ^t .

that converts a vector's reference frame from L_k to C_k^t , and $\mathbf{t}_{L_k C_k}^t$ is a translational vector from L_k to C_k^t .

We assume to work with monocular cameras that only allow estimating the agent- k 's trajectory and map point coordinates up to a common global scale factor λ_{GL_k} . In addition, we assume that only agent-1 recovers its absolute global scale λ_{GL_1} , thus setting the local reference frame of the agent-1 L_1 as the global frame G . Since the origin and orientation of the global frame is same as the agent-1's local frame, the relative similarity matrix from L_1 to G , denoted with \mathbf{S}_{GL_1} (in the Sim(3) Lie group), only includes this global scale factor as

$$\mathbf{S}_{GL_1} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0} \\ 0 & \lambda_{GL_1}^{-1} \end{bmatrix}. \quad (1)$$

The agent-2 recovers its global scale indirectly by estimating the relative similarity matrix from L_2 to L_1 :

$$\mathbf{S}_{L_1 L_2} = \begin{bmatrix} \mathbf{R}_{L_1 L_2}^t & \mathbf{t}_{L_2 L_1}^t \\ 0 & \lambda_{L_1 L_2}^{-1} \end{bmatrix} \quad (2)$$

The inter-agent range measurement between the tag modules at time t is described as ρ^t .

III. ALGORITHM OVERVIEW

We illustrate the overview of the proposed collaborative SLAM system with two agents in Fig. 2.

First, each agent estimates local map points and poses with respect to its own local frame using monocular visual odometry. Since a monocular vision-based system is used, the global scales of both agents are inherently ambiguous. The agent-1 recovers the absolute global scale by e.g. observing a known pattern (checkerboard), while the agent-2

indirectly recovers its global scale by estimating the relative similarity matrix from its own system to the agent-1's system by e.g. using map points observed by both agents. Using the estimated global scale of the agent-1's system and the relative similarity matrix between the agents, the locally-referred poses can be converted to the global frame for the proposed data fusion. Note that our method requires to observe a known pattern and matching map points between agents only at the beginning of the mission to properly initialize the global scale and relative pose between agents. These measurements are not needed after the initialization for the proposed data fusion method.

To fuse local data, the agent-2 transmit its poses (6DoF) to the agent-1 after converting the reference frame from L_2 to L_1 , by exploiting the initialized relative similarity between the agents. Then, on the agent-1's onboard computer, the reference frame of all the poses are converted to the global frame. In addition, inter-agent range measurements are obtained using time-of-flight (ToF) measurements between each agent's ranging onboard module (tag).

Before fusing the odometry and ranging data, we synchronize the collected measurements using their timestamps. Then, the collected local odometry data and range measurements are fused to mitigate the drifts in each agent's local pose estimates. After this fusion process, the reference frame of each agent's pose is converted back to its own local frame, and the local map points are updated using the pose differences (before and after the fusion process) of the frame at which map points are first generated.

As shown in Fig. 2, only the agent-2's poses (6DoF) and ranges (scale values) are transmitted between the agents during the entire process, which can significantly reduce the exchanging data size and computational load compared to other collaborative SLAM methods that thoroughly use map point matches between agents.

IV. FUSION OF ODOMETRY AND INTER-AGENT RANGING

In this section, we explain a fusion of odometry and inter-agent range measurements for accurate relative positioning. The data fusion problem is formulated using the factor graph-based method [14].

A. Factorization

First, we formulate a factor graph as shown in Fig.3. In this figure, circular nodes represent the unknowns (state variables), which are the agents' poses converted to the global frame:

$$\mathbf{T}_{ov} = [\mathbf{T}_{C_1 G}^{t_1}, \dots, \mathbf{T}_{C_1 G}^{t_N} | \mathbf{T}_{C_2 G}^{t_1}, \dots, \mathbf{T}_{C_2 G}^{t_N}].$$

The likelihood of the states $\mathbf{T}_{C_k G}^{t_i}$ and $\mathbf{T}_{C_k G}^{t_j}$ given the odometry measurements

$$\mathbf{z}_{vo,k}^{j,i} = \hat{\mathbf{T}}_{C_k L_k}^{t_j} (\hat{\mathbf{T}}_{C_k L_k}^{t_i})^{-1}$$

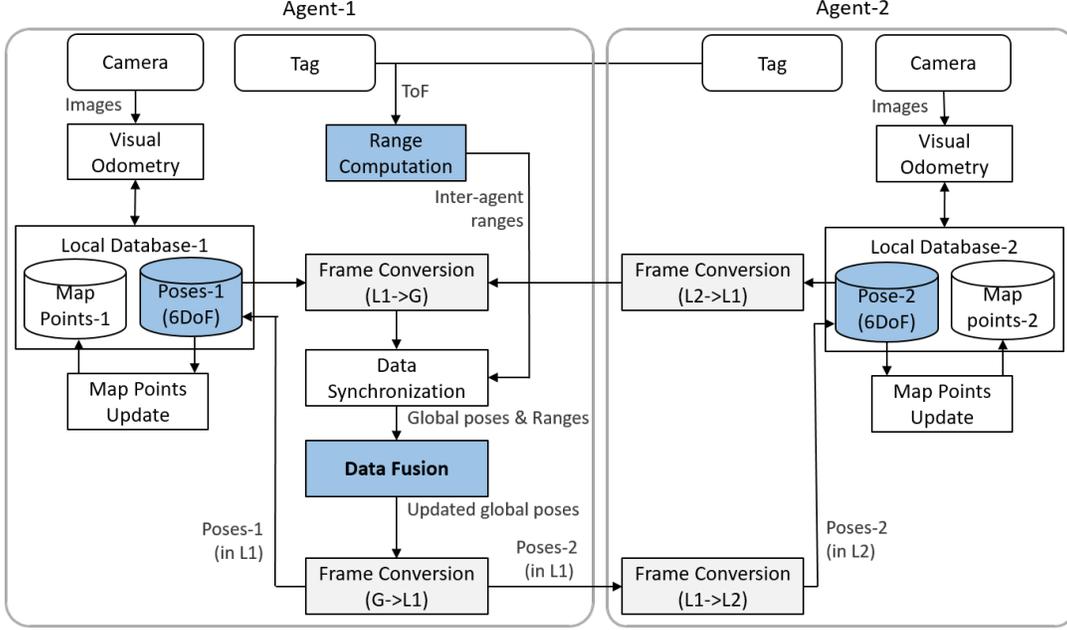


Fig. 2: An overview of the proposed decentralized collaborative SLAM. The local frame of the agent-1 is set as the global frame of the entire system, and the computation processes for fusing data are conducted on the agent-1.

are denoted as the factors between the camera nodes $\phi_{vo,k}^{ji}$. Assuming Gaussian noise in the measurements,

$$\begin{aligned} \phi_{vo,k}^{ji} &= l(\mathbf{T}_{C_k G}^{t_i}, \mathbf{T}_{C_k G}^{t_j} | \mathbf{z}_{vo,k}^{j_i}) \\ &\propto \exp\left\{-\frac{1}{2} \|\mathbf{e}_{vo}^{j_i}(\mathbf{T}_{C_k G}^{t_i}, \mathbf{T}_{C_k G}^{t_j}, \mathbf{z}_{vo,k}^{j_i})\|^2\right\}, \end{aligned} \quad (3)$$

with $\mathbf{e}_{vo}^{j_i}(\mathbf{T}_{C_k G}^{t_i}, \mathbf{T}_{C_k G}^{t_j}, \mathbf{z}_{vo,k}^{j_i})$ denotes the visual odometry error function. Since camera poses are described with matrices in SE(3) of the Lie group, odometry errors cannot be directly computed by subtracting measurements from the model. To compute the errors, first, we compute the error matrix in SE(3) as

$$\mathbf{E}_{vo}(\mathbf{T}_{C_k G}^{t_i}, \mathbf{T}_{C_k G}^{t_j}, \mathbf{z}_{vo,k}^{j_i}) = (\mathbf{T}_{C_k L_k}^{t_j})^{-1} \mathbf{z}_{vo,k}^{j_i} \mathbf{T}_{C_k L_k}^{t_i} \quad (4)$$

Then, this error matrix is converted to vector $\mathbf{e}_{vo}^{j_i}(\mathbf{T}_{C_k G}^{t_i}, \mathbf{T}_{C_k G}^{t_j}, \mathbf{z}_{vo,k}^{j_i})$, which is a 6×1 vector in se(3) of the Lie algebra [15].

In addition, we add factors that represent the likelihood of the states $\mathbf{T}_{C_1 G}^{t_i}$ and $\mathbf{T}_{C_2 G}^{t_i}$ given the inter-agent range measurements \mathbf{z}_r^i as a factor ϕ_r^i between the camera nodes whenever range measurements are available. With a Gaussian noise assumption,

$$\begin{aligned} \phi_r^i &= l(\mathbf{T}_{C_1 G}^{t_i}, \mathbf{T}_{C_2 G}^{t_i} | \mathbf{z}_r^i) \\ &\propto \exp\left\{-\frac{1}{2} \|\mathbf{e}_r^i(\mathbf{T}_{C_1 G}^{t_i}, \mathbf{T}_{C_2 G}^{t_i}, \mathbf{z}_r^i)\|^2\right\}, \end{aligned} \quad (5)$$

where $\mathbf{e}_r^i(\mathbf{T}_{C_1 G}^{t_i}, \mathbf{T}_{C_2 G}^{t_i}, \mathbf{z}_r^i)$ denotes the ranging error function. Since range measurements are scalar values, the errors can be easily defined as the difference between measurements and the value computed with the model

$$\mathbf{e}_r^i(\mathbf{T}_{C_1 G}^{t_i}, \mathbf{T}_{C_2 G}^{t_i}, \mathbf{z}_r^i) = \mathbf{z}_r^i - \mathbf{h}_r^i(\mathbf{T}_{C_1 G}^{t_i}, \mathbf{T}_{C_2 G}^{t_i}), \quad (6)$$

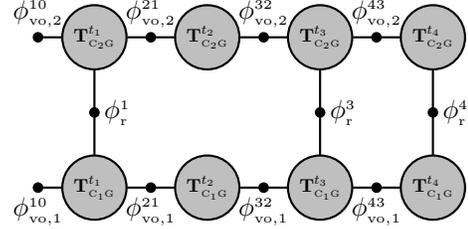


Fig. 3: The formulated factor graph including unknowns $\mathbf{T}_{C_1 G}^{t_i}$ and $\mathbf{T}_{C_2 G}^{t_i}$, odometry factors $\phi_{vo,k}^{ji}$, and ranging factors ϕ_r^i

where $\mathbf{h}_r^i(\mathbf{T}_{C_1 G}^{t_i}, \mathbf{T}_{C_2 G}^{t_i})$ is ranging measurement model function:

$$\begin{aligned} \mathbf{h}_r^i(\mathbf{T}_{C_1 G}^{t_i}, \mathbf{T}_{C_2 G}^{t_i}) &= \|\mathbf{g}_{\mathbf{p}_{T_1 T_2}^{t_i}}\| \\ &= \|\mathbf{p}_{GC_2}^{t_i} - \mathbf{p}_{GC_1}^{t_i} + (\mathbf{R}_{GC_2}^{t_i} - \mathbf{R}_{GC_1}^{t_i}) \mathbf{p}_{CT}\| \\ &= \| -(\mathbf{R}_{C_2 G}^{t_i})^T \mathbf{t}_{GC_2}^{t_i} + (\mathbf{R}_{C_1 G}^{t_i})^T \mathbf{t}_{GC_1}^{t_i} + (\mathbf{R}_{GC_2}^{t_i} - \mathbf{R}_{GC_1}^{t_i}) \mathbf{p}_{CT} \|^2 \end{aligned} \quad (7)$$

B. Least-Squares Estimation

Assuming Gaussian noise in measurements, the optimal solution that maximizes the multiplication of all the factors is the same as the solution that minimizes the summation of the squared errors between measurements and models:

$$\begin{aligned} \hat{\mathbf{T}}_{ov} &= \underset{\mathbf{T}_{ov}}{\operatorname{argmax}} \prod_k \prod_{i,j} \phi_{vo,k}^{ji} \prod_i \phi_r^i \\ &= \underset{\mathbf{T}_{ov}}{\operatorname{argmin}} \sum_k \sum_{i,j} (\mathbf{e}_{vo,k}^{j_i})^T \boldsymbol{\Omega}_{vo,k} \mathbf{e}_{vo,k}^{j_i} + \sum_l (\mathbf{e}_r^l)^T \boldsymbol{\Omega}_r \mathbf{e}_r^l, \end{aligned} \quad (8)$$

where the inversed measurement covariance matrices are denoted with $\boldsymbol{\Omega}$.

To solve the least-squares problem (8), we apply the Levenberg-Marquardt algorithm. This algorithm iteratively updates the state with $\Delta \hat{\mathbf{T}}_{\text{ov}}^*$ computed by solving

$$(\mathbf{H} + \lambda \mathbf{I}) \Delta \hat{\mathbf{T}}_{\text{ov}}^* = -\mathbf{b}, \quad (9)$$

where $\mathbf{H} = \sum_k \mathbf{J}_k^T \Omega_k \mathbf{J}_k$ and $\mathbf{b} = \sum_k \mathbf{J}_k^T \Omega_k \mathbf{e}_k$ with Jacobian matrices \mathbf{J}_k . For the iterative computations, $g2o$ [16] is used.

C. Local Poses Recovery and Local Map Point Update

Finally, we recover each agent's local poses $\hat{\mathbf{T}}_{\text{C}_k \text{L}_k}^{t_i}$ using the agents' global pose estimates $\hat{\mathbf{T}}_{\text{C}_k \text{G}}^{t_i}$ and the initially recovered similarity matrices (1) and (2):

$$\hat{\mathbf{T}}_{\text{C}_1 \text{L}_1}^{t_i} = \begin{bmatrix} \hat{\mathbf{R}}_{\text{C}_1 \text{G}}^{t_i} & \lambda_{\text{GL}_1}^{-1} \hat{\mathbf{t}}_{\text{GC}_1}^{t_i} \\ 0 & 1 \end{bmatrix}$$

$$\hat{\mathbf{T}}_{\text{C}_2 \text{L}_2}^{t_i} = \begin{bmatrix} \hat{\mathbf{R}}_{\text{C}_2 \text{G}}^{t_i} \mathbf{R}_{\text{L}_1 \text{L}_2} & \lambda_{\text{GL}_1}^{-1} \lambda_{\text{L}_1 \text{L}_2}^{-1} \hat{\mathbf{t}}_{\text{GC}_2}^{t_i} - \hat{\mathbf{R}}_{\text{C}_2 \text{G}}^{t_i} (\lambda_{\text{L}_1 \text{L}_2}^{-1} \mathbf{t}_{\text{L}_2 \text{L}_1}^{t_i}) \\ 0 & 1 \end{bmatrix}$$

Then, local map points $\mathbf{p}_{\text{L}_k \text{M}_n}$ are updated on each agent's on-board computer, using the previous pose estimate $\tilde{\mathbf{T}}_{\text{C}_k \text{L}_k}^{t_i}$ (before the fusion) and updated ones $\hat{\mathbf{T}}_{\text{C}_k \text{L}_k}^{t_i}$ (after the fusion) of the camera frame at which each map point is first generated:

$$\hat{\mathbf{p}}_{\text{L}_k \text{M}_n} = (\hat{\mathbf{T}}_{\text{C}_k \text{L}_k}^{t_i})^{-1} \tilde{\mathbf{T}}_{\text{C}_k \text{L}_k}^{t_i} \tilde{\mathbf{p}}_{\text{L}_k \text{M}_n}.$$

V. EVALUATIONS

First, we show the reduced exchanging data size for the proposed collaborative SLAM method compared to the approaches that use map fusion. Then, we evaluate the proposed method in terms of relative positioning accuracy with a public dataset.

A. Analysis on Data Packet Size

Table I shows the required transmitting data size for collaborative SLAM using map matches (both centralized and decentralized systems) and inter-agent range measurements (decentralized system, our method) for two agents. We assume that the binary robust independent elementary feature (BRIEF) [17] descriptors are used to describe the characteristics of feature points.

For decentralized SLAM with map fusion, one of the agents needs to transmit its own poses (6 floats for each pose) and local map points (3 floats for each point), as well as detected feature point positions in image frames and their descriptors, to the other agent. When agent- k estimates its poses at N_k timestamps, generates total M_k map points, and i -th map point is detected at $m_{k,i}$ frames, then the agent needs to transfer

$$(N_k \times 192) + (M_k \times 96) + \sum_{i=1}^{M_k} m_{k,i} (64 + 128) [\text{bits}].$$

TABLE I: The required exchanging data packet for collaborative SLAM with inter-agent map matches and ranging. The number of the agent- k 's poses is denoted with N_k . M_k denotes the number of local map points estimated with the agent- k , and M_k^0 are the number of local map points required to initialize relative geometry between the agents. $m_{k,i}$ are the number of time instances at which the i -th map point is observed, and n_ρ is the number of time-of-flight measurements (ranging). In the last row, the total required data size is presented in KB for a simply formulated collaborative SLAM problem: $N_1 = N_2 = 10$, $M_1 = M_2 = 100$, $M_2^0 = 5$, and $m_{1,i} = m_{2,i} = 2$.

Data	Map Matches		Ranges
	Cen.	Decen.	(Decen., Ours)
Poses1	$6N_1 f$		
Map Points1	$3M_1 f$		
Features+ Descriptors2	$\sum_{i=1}^{M_1} m_{1,i} \times (2f + 128b)$		
Poses2	$6N_2 f$	$6N_2 f$	$6N_2 f$
Map Points2	$3M_2 f$	$3M_2 f$	$3M_2^0 f$
Features+ Descriptors2	$\sum_{i=1}^{M_2} m_{2,i} \times (2f + 128b)$	$\sum_{i=1}^{M_2} m_{2,i} \times (2f + 128b)$	$\sum_{i=1}^{M_2^0} m_{2,i} \times (2f + 128b)$
Time-of-flight			$n_\rho f$
Total [KB]	12.48	6.24	2.67

To use a central server, both agents need to send this information to the server.

However, the proposed method only requires to send this visual information only for initializing the relative geometry between the agents. After the initialization, we only need to transmit the data packet without visual data:

$$(N_k \times 192) + (n_\rho \times 32) [\text{bits}],$$

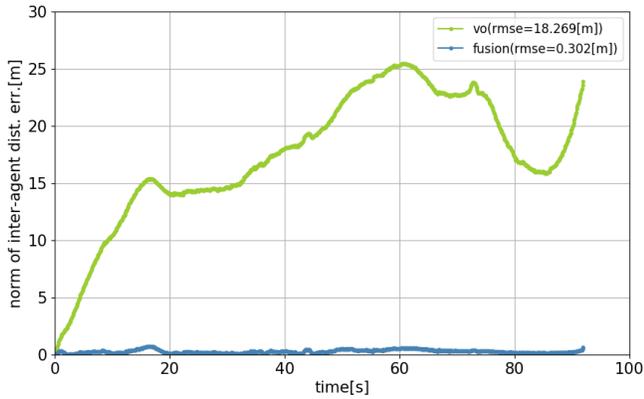
where n_ρ is the number of timestamps at which inter-agent ranging is available.

The last row of Table I shows the total required data size in kilo bytes [KB] when each vehicle estimates its poses at 10 timestamps, generates 100 local map points, 5 map points are needed for the initialization, and each map point is observed at two image frames. Compared to the approaches with map fusion, the proposed method shows a largely reduced amount of the required data transmission. Considering $N_k \ll M_k$ in practice, our method can be useful when a swarm system in an environment where communication capabilities are very limited.

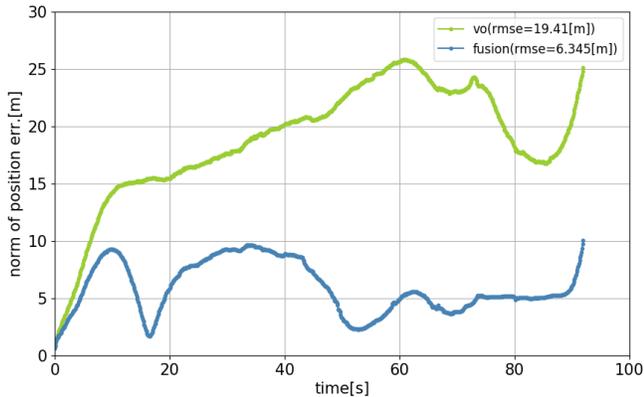
B. Analysis on Positioning Accuracy with a Public Dataset

We evaluate the relative positioning accuracy of the proposed collaborative SLAM using the image sequence 00 of the KITTI dataset [18]. This is a image dataset acquired with a car, so first we use two parts of the entire dataset, and assume that each part is traveled by one agent. Since inter-agent ranges are not included in the dataset, we generated synthetic measurements using the ground truth trajectory and artificially added Gaussian noise.

The errors in relative distances between two agents over time are shown in Fig. 4a. The green line represents the errors over time when only with visual odometry is used.



(a) The absolute relative distance errors of positions estimated using only visual odometry and the proposed method with inter-agent range measurements over time.



(b) The errors of relative positions estimated using only visual odometry and the proposed method with inter-agent range measurements over time.

Fig. 4: The errors in relative distances and positions over time

The blue line is the errors over time when the proposed data fusion is applied using perfectly accurate inter-agent range measurements (no noise). This figure shows that the proposed method can reduce the relative distance errors between agents, which is helpful for the swarm system to e.g. avoid collisions.

Fig. 4b compares the error magnitude of the relative positions estimated only with visual odometry (green) to the proposed method with inter-agent range measurements (blue). This figure shows that the errors in relative positioning are reduced with the proposed method. In Table II, we analyze the accuracy of relative positioning in terms of ranging noise magnitude. In the first row, the root mean square error (RMSE) of the estimates obtained with only visual odometry are presented. From the second row, the RMSE of the estimates obtained by using the proposed method with various noise level on inter-agent range measurements are given. As range measurements becomes inaccurate with a larger standard deviation (σ_ρ), relative positioning becomes inaccurate, and the RMSE of the relative distances between the agents becomes larger. However, we can observe that our method outperforms visual odometry in all the cases.

TABLE II: The RMSE of the relative distances and position estimates obtained by using only visual odometry and the proposed method with various standard deviation (σ_ρ) of Gaussian noise on inter-agent range measurements

Setting	RMSE [m]	
	Relative dist.	Relative position
Visual odometry	18.29	19.410
No noise	0.302	6.345
$\sigma_\rho=0.1\text{m}$	0.311	6.346
$\sigma_\rho=0.5\text{m}$	0.346	6.347
$\sigma_\rho=1.0\text{m}$	0.477	6.350

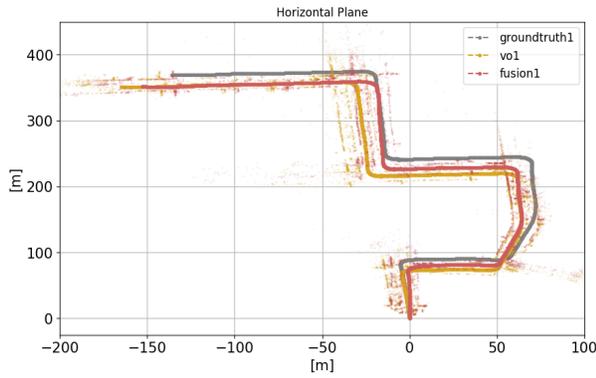
Fig. 5 illustrates the trajectory estimates with generated map points in horizontal and vertical planes. Fig. 5a and Fig. 5b show the trajectory and map points estimates in the horizontal and vertical plane of the agent-1 in its local frame. Fig. 5c and 5d show the agent-2's trajectory estimates in the horizontal and vertical plane in its own local frame. The combined trajectories and map points in the global frame are illustrated in Fig. 5e and 5f. In these figure, we cannot see a large improvement of absolute positioning with the proposed method, which is expected since our approach exploits inter-agent range measurements that only provide relative information to the system. To mitigate the drifts in global position estimates, we need to use absolute measurements, such as absolute position measurements obtained with GNSS-based systems [19] and ranging links to a base station (anchor point) as proposed in [20] and [21].

VI. CONCLUSION

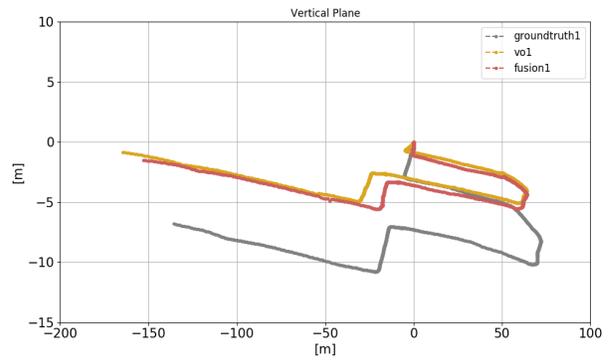
We propose an accurate cooperative positioning method by exploiting inter-agent range measurements for swarm systems. After the initialization, the proposed method only requires to exchange the poses of one of the agents with time-of-flight observations. With a simply formulated cooperative localization problem, we showed the reduced size of exchanging data packets, compared to methods that use map fusions to mitigate the estimation errors. Then, improved accuracy of relative positioning is shown with simulation results obtained with real images and synthetically generated inter-agent range measurements. Since the proposed method only reduces relative positioning errors, absolute positions of agents could be shifted or rotated. To mitigate global positioning errors, the proposed method needs to exploit additional absolute measurements, such as absolute position measurements and ranging links to a base station.

REFERENCES

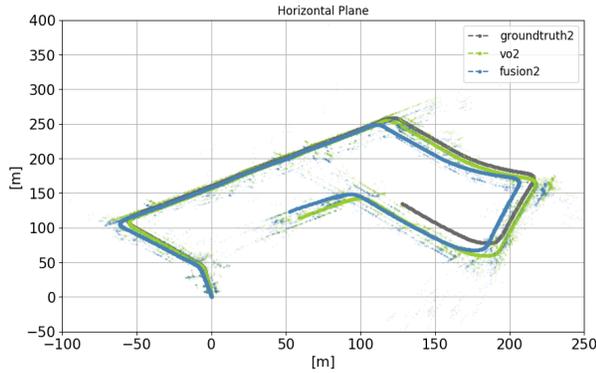
- [1] P. Schmuck and M. Chli, "Multi-uav collaborative monocular slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3863–3870.
- [2] —, "Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.
- [3] F. Li, S. Yang, X. Yi, and X. Yang, "Corb-slam: a collaborative visual slam system for multiple robots," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer, 2017, pp. 480–490.
- [4] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.



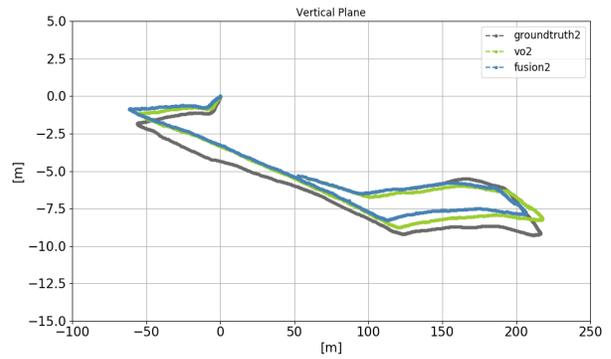
(a) Trajectory and map point estimates in the horizontal plane of L_1 obtained with the agent-1.



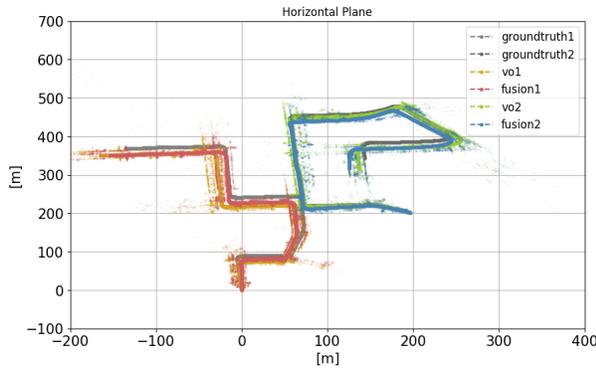
(b) Trajectory estimates in the vertical plane of L_1 obtained with the agent-1.



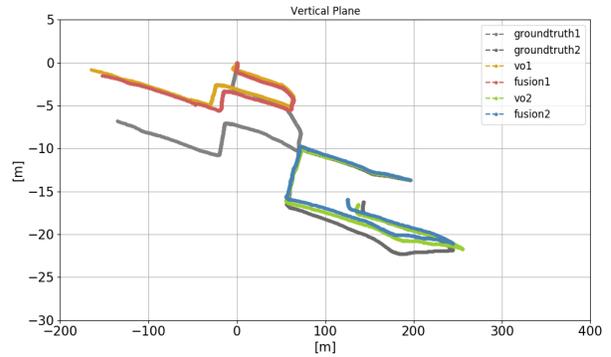
(c) Trajectory and map point estimates in the horizontal plane of L_2 obtained with the agent-2.



(d) Trajectory estimates in the vertical plane of L_2 obtained with the agent-2.



(e) Trajectory and map point estimates in the horizontal plane of G obtained with the agent-1 and agent-2.



(f) Trajectory estimates in the vertical plane of G obtained with the agent-1 and agent-2.

Fig. 5: Trajectories and map point estimates of each agent’s local estimates, and combined global estimates represented in the global frame.

- [5] G. Mohanarajah, V. Usenko, M. Singh, R. D’Andrea, and M. Waibel, “Cloud-based collaborative 3d mapping in real-time with low-cost robots,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 423–431, 2015.
- [6] A. Cunningham, M. Paluri, and F. Dellaert, “Ddf-sam: Fully distributed slam using constrained factor graphs,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3025–3030.
- [7] 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*. IEEE, 2013, pp. 5220–5227.
- [8] T. Cieslewski, S. Choudhary, and D. Scaramuzza, “Data-efficient decentralized visual slam,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2466–2473.
- [9] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn architecture for weakly supervised place recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.
- [10] T. Cieslewski and D. Scaramuzza, “Efficient decentralized visual place recognition from full-image descriptors,” in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2017, pp. 78–82.
- [11] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, “Distributed trajectory estimation with privacy and com-

- munication constraints: a two-stage distributed gauss-seidel approach,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5261–5268.
- [12] T. S. Richardson, C. G. Jones, A. Likhoded, E. Sparks, A. Jordan, I. Cowling, and S. Willcox, “Automated vision-based recovery of a rotary wing unmanned aerial vehicle onto a moving platform,” *Journal of Field Robotics*, vol. 30, no. 5, pp. 667–684, 2013.
- [13] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, “Multiple relative pose graphs for robust cooperative mapping,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3185–3192.
- [14] M. Kaess, A. Ranganathan, and F. Dellaert, “isam: Incremental smoothing and mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [15] B. Hall, *Lie groups, Lie algebras, and representations: an elementary introduction*. Springer, 2015, vol. 222.
- [16] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, “g2o: A general framework for (hyper) graph optimization,” *Tech. Rep.*, 2011.
- [17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *European conference on computer vision*. Springer, 2010, pp. 778–792.
- [18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [19] R. Mascaró, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, “Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1421–1428.
- [20] Y.-H. Lee, C. Zhu, G. Giorgi, and C. Günther, “Fusion of monocular vision and radio-based ranging for global scale estimation and drift mitigation,” *arXiv preprint arXiv:1810.01346*, 2018.
- [21] —, “Mitigation of odometry drift with a single ranging link in gnss-limited environments,” *Proceedings of ION ITM 2020 (to be published)*, 2020.