2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

On Data Sharing Strategy for Decentralized Collaborative Visual-Inertial Simultaneous Localization And Mapping

Rodolphe Dubois[1,2], Alexandre Eudes[1], Vincent Frémont[2]
[1] DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France
[2] Centrale Nantes, LS2N, UMR 6004, Nantes, France
E-mail: [1] firstname.lastname@onera.fr, [2] firstname.lastname@ls2n.fr

***Abstract*— This article introduces and evaluates two decentralized data sharing algorithms for multi-robot visual-inertial simultaneous localization and mapping (VI-SLAM): Factor Sparsification for Visual-Inertial Packets (FS-VIP) and Min-K-Cover Selection for Visual-Inertial Packets (MKCS-VIP). Both methods make robots regularly build and exchange data packets which describe the successive portions of their map, but rely on distinct paradigms. While FS-VIP builds on consistent marginalization and sparsification techniques, MKCS-VIP selects raw visual and inertial information which can best help to perform a faithful and consistent re-estimation while reducing the communication cost. Performances in terms of accuracy and communication loads are evaluated on multi-robot scenarios built on both available (EUROC) and custom datasets (SOTTEVILLE).**

I. INTRODUCTION

A. Considered problem

The use of a fleet of robots is increasingly being considered for applications like the exploration or the inspection of large-scale areas. For such tasks, Simultaneous Localization And Mapping (SLAM) techniques are core algorithmic blocks which support higher-level decision and control algorithms. When performing SLAM, a robot relies on the measurements acquired by its embedded sensors to estimate in real time a model of its environment and its trajectory within it.

While single-robot SLAM has been extensively studied [1], its multi-robot counterpart has introduced new kinds of challenges to cope with when exchanging data between robots. Robots share knowledge by exchanging data packets designed to provide adequate information: i) to enrich the recipient's map by correctly registering the received data w.r.t. it; ii) to refine it by correlating the provided information with its underlying estimation model, such that subsequent re-estimations should enhance the accuracy of the estimates without compromising *their consistency* i.e. the correct assessment of the covariance matrices on the estimates.

Additionally to real-time requirements, robots must face various constraints when building packets, the most of which being communication constraints such as limited communication range and bounded bandwidth. Thus, communication policies should enhance the autonomy of each agent, and the amount of transmitted data should be as flexible as possible depending on those constraints.

B. Related works

1) Centralized multi-robot SLAM*:* Most of the data sharing methods that have been introduced so far are centralized: they only allow data exchanges between single agents and a central server unit. Forster *et al.* [2] developed a centralized monocular visual SLAM framework which assigns low-level tasks such as keyframe selection and processing to the agents while high-level tasks for Collaborative Structure-from-Motion like mapping, loop closing and map merging are performed by a ground station. Karrer *et al.* [3] have recently introduced a similar system for VI-SLAM called CVI-SLAM. However, those methods often require to exchange significant amounts of data that might exceed the available communication bandwidth. This is the case when the whole set of 2D keypoints and visual descriptors of each frame is communicated. Furthermore, each agent must remain within the server's communication range, so it reduces its autonomy. Finally, centralized architectures are critically vulnerable to failures of the central server.

2) Decentralized multi-robot SLAM*:* Because of the previously cited drawbacks, many authors have focused on decentralized SLAM methods. Schuster *et al.* [4] proposed a submap-based approach for visual-stereo SLAM that makes each robot share some local submaps with neighboring robots, which are then aligned via ICP point-cloud registration. Contreras *et al.* [5] also proposed a decentralized method for LiDAR SLAM. To cope with limited bandwidth constraints, other works make robots exchange summarized representations of their map to meet the bandwidth requirements and ease their alignment within the recipient robot's map. Such methods often rely on marginalization and sparsification techniques [6]. For instance, Cunningham *et al.* [7] proposed DDF-SAM in which one robots exchange marginalized maps over some shared variables (a.k.a. separators) like commonly observed landmarks. In a similar way, Lazaro *et al.* [8] make robots exchange condensed measurements computed through marginalization over separators. Paull *et al.* [9] proposed a more robust framework based on the bookkeeping of inter-robot exchanges for acoustic underwater decentralized SLAM. Nonetheless, few methods of that kind have been specifically designed for VI-SLAM.

C. Contributions

In this paper, we propose and evaluate Factor Sparsification for Visual-Inertial Packets (FSVIP) and Min-K-Cover Selection Packets (MKCS-VIP), two decentralized data sharing algorithms for VI-SLAM. The rest of this paper is organized

978-1-7281-4004-9/19/$31.00 ©2019 IEEE

as follows. In section II, the classical landmark-based VI-SLAM problem is formulated within a Maximum Likelihood (ML) estimation scheme. Sections III and IV respectively introduce FSVIP and MKCS-VIP, whose performances are evaluated in section V. Concluding remarks are finally proposed in the last section of the paper.

II. Monocular Visual-Inertial SLAM

In VI-SLAM, each robot relies on the images acquired by its monocular camera, and on the specific acceleration and angular velocity measurements supplied by its Inertial Measurement Unit (IMU) to estimate its own trajectory and to build a map of its environment. A *Front-End* module first interprets those measurements to build online a probabilistic graphical model known as a *factor graph*, as represented in the figure (1), which encodes the measurements likelihood w.r.t. hidden variables purposely introduced, based on the observations, to model the trajectory and the environment.

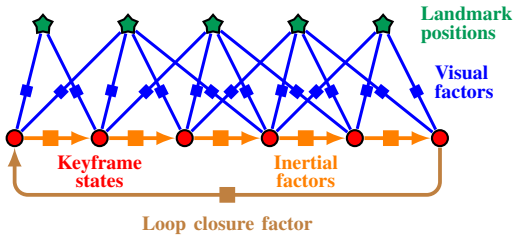


Fig. 1: SLAM graphical probabilistic model

The trajectory is modelled as a discrete set $\mathcal{X}$ of keyframes. Each keyframe $\boldsymbol{x_k}$ is described by the pose of the robot $\boldsymbol{T}_{\mathrm{WB},k} \in \mathrm{SE}(3)$ – where the subscript k denotes time t_k, W the gravity-aligned world frame (whose origin coincides with the position of the first keyframe) and B the body frame – and its inertial state $\boldsymbol{\xi}_k = [{}^{\mathrm{W}}\boldsymbol{v}_k^\top \ {}^{\mathrm{B}}\boldsymbol{b}_{\boldsymbol{a}k}^\top \ {}^{\mathrm{B}}\boldsymbol{b}_{\boldsymbol{\omega}k}^\top]^\top$ whose components respectively denote the velocity, the accelerometer and the gyrometer biases. The environment is sparsely represented as a set $\mathcal{L}$ of triangulated 3D landmarks whose observations have first been tracked through the acquired images. As illustrated in the figure (1), all those variables are mutually constrained by Gaussian factors, which are stochastic constraints derived from the measurements. Each IMU measurement $\boldsymbol{u_k}$ yields the following inertial factor:

$$p\left(\boldsymbol{x_{k+1}}|\boldsymbol{x_k},\boldsymbol{u_k}\right) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{x_{k+1}} \boxminus f_k(\boldsymbol{x_k},\boldsymbol{u_k})\right\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1}}\right) \tag{1}$$

where $\boldsymbol{x}_{(.)} \in \mathcal{X}$, f_k is the integrated discrete-time dynamics of the robot and $\boldsymbol{\Sigma_u}$ the covariance of the residual propagated from the IMU noise model. The $\boxminus$ minus operator returns the residual in a tangent space of the ambient composite Lie group in which one $\boldsymbol{x}_{(.)}$ evolves [10]. Equally, each landmark observation $\boldsymbol{z_k^i}$ yields the following factor:

$$p\left(\boldsymbol{z_k^i}|\boldsymbol{T}_{\mathrm{WB},k},{}^{\mathrm{W}}\boldsymbol{l_i}\right) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{z_k^i} - \pi_{\mathrm{C}}(\boldsymbol{T}_{\mathrm{WB},k},{}^{\mathrm{W}}\boldsymbol{l_i})\right\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{v}}^{-1}}\right) \tag{2}$$

where ${}^{\mathrm{W}}\boldsymbol{l_i} \in \mathcal{L}$, $\boldsymbol{\Sigma_v}$ is the covariance of the visual measurement, and π_{C} is the projection of the landmark to the camera frame C. Finally, the robot can refine its factor graph by closing loops. A robot closes a loop when it knowingly observes an area it has already mapped. It thus derives a relative pose constraint from 2D-3D correspondences, merges the associated landmarks and corrects the inertial drift it had accumulated in the meanwhile.

A *Back-End* module finally infers the map and the trajectory based on the resulting factor graph. Popular approaches rely on Maximum Likelihood (ML) estimators which solve the following graph optimization problem:

$$\{\mathcal{X}^*_{\mathrm{ML}}, \mathcal{L}^*_{\mathrm{ML}}\} = \underset{\mathcal{X},\mathcal{L}}{\arg\min} \ -\log p(\mathcal{Z}|\mathcal{X},\mathcal{L}) \tag{3}$$

where $\mathcal{Z}$ is the set of visual and inertial measurements. $p(\mathcal{Z}|\mathcal{X},\mathcal{L})$ is proportional to the product of the factors defined by (1) and (2). Equation (3) yields a nonlinear least-square optimization problem. Note that in VI-SLAM, the visual observations allow to estimate the trajectory and the 3D structure of the environment up to a scale factor which is made observable by the IMU measurements.

III. Factor Sparsification for Visual-Inertial Packets (FS-VIP)

The method described in this section is transposed from the one introduced in [9] for acoustic underwater SLAM, that we adapted to the specific constraints of VI-SLAM (e.g. the need to properly condense the visual-inertial factors and to provide enough visual information to spot inter-robot loop closures).

A. Overview of the proposed method

An overview of all the processes involved in FS-VIP is given in the figure (2) and detailed in the next paragraphs. While the map is being built as an output of the SLAM algorithm, three additional threads are running to i) precompute the packets that will be sent to the other robots; ii) handle the packet exchange and iii) integrate the received packets.

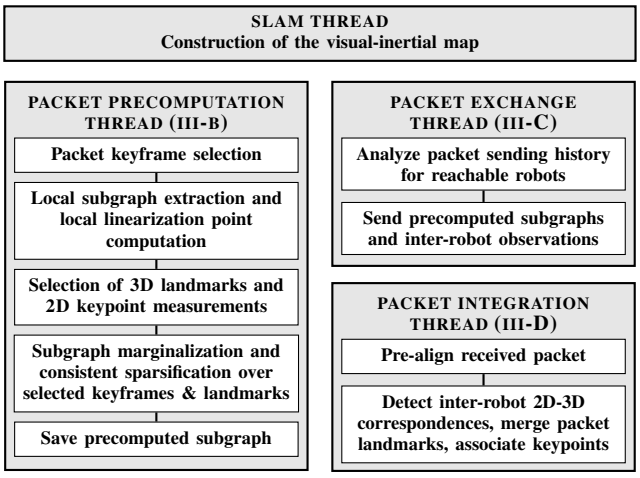
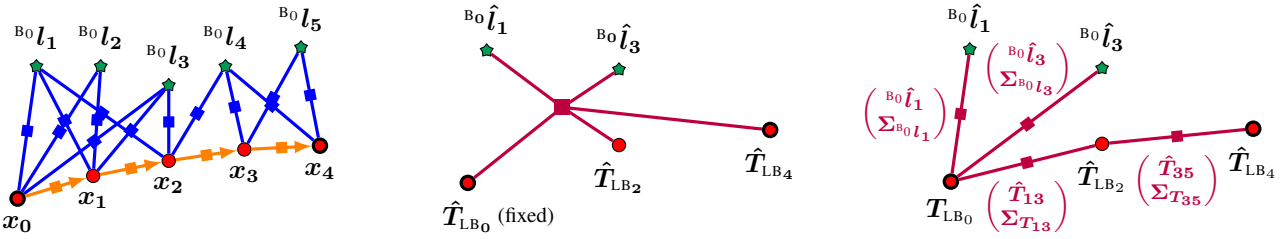


Fig. 2: Overview of FS-VIP

B. Packet precomputation

In parallel to the mapping process, each robot precomputes the summarized representations of the successive portions of its map that it will send to the other robots.



(a) Subgraph extraction and optimization (b) Subgraph marginalization (c) Subgraph sparsification

Fig. 3: Overview of the subgraph extraction, marginalization and sparsification process

1) Keyframe selection: When opening a new packet, the first step is to select online some keyframes along the trajectory, based on covisibility criteria which are looser than the ones used for regular keyframe selection. For each keyframe along the trajectory, we compute the number n_c and the ratio r_c of commonly observed landmarks with the lastly selected keyframe. If either $n_c \leq n_c^{\lim}$ or $r_c \leq r_c^{\lim}$ where $n_c^{\lim}$ and $r_c^{\lim}$ are two user-defined thresholds, then the current keyframe is selected. The aim is to select a minimal set of keyframes which covers the area observed from the sub-trajectory keyframes.

2) Subgraph extraction: We compute a new packet as soon as a new keyframe is selected and an estimated distance $d_{\min}$ has been run since the last packet computation. The corresponding factor graph is then extracted as showed in the figure (3a). It holds all the keyframes and observed landmarks states as well as the visual and inertial factors linked to the extracted sub-trajectory.

3) Local linearization point computation: All the keyframe and landmark states included in the subgraph are then locally inferred by solely optimizing over the extracted factors and fixing the pose of the first keyframe. We thus get a local linearization point which only depends on the information held within the extracted subgraph. Note that the landmark positions should be estimated relatively to the first keyframe, while the keyframe poses can be estimated relatively to any gravity-aligned reference frame L. Another remark is that the subgraph extraction may make some previously well-constrained landmarks badly-constrained (according to the criteria defined in [11][1]). Those should be identified and removed after running the optimization.

4) Visual information selection: Contrary to [9] which relies on nearest landmark association to compute inter-robot loop closures, minimal visual information should be communicated to spot them in a visual-inertial framework. We choose to transmit local visual information on the keyframes selected in III-B.1. For each one, we communicate the $n_{\text{keypoints}}$ keypoints associated with the most observed landmarks in the map along with their full visual descriptor, and among those observed landmarks, we select the $n_{\text{landmarks}}$ $(\leq n_{\text{keypoints}})$ most observed landmarks within the extracted subgraph.

5) Subgraph marginalization: As a fourth step, the local estimate's information matrix is computed as the *Fisher Information Matrix* (FIM) of the subgraph, evaluated at the inferred local estimate:

$$\boldsymbol{\mathcal{I}}_{\boldsymbol{\mathcal{Z}}}(\mathcal{X}^*_{\text{ML}}, \mathcal{L}^*_{\text{ML}}) = \sum_{\boldsymbol{z}\in\mathcal{Z}} \boldsymbol{J}^{\boldsymbol{z}\top}_{\{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{L}}\}} \boldsymbol{\Sigma}_{\boldsymbol{z}}^{-1} \boldsymbol{J}^{\boldsymbol{z}}_{\{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{L}}\}} \tag{4}$$

where $\boldsymbol{J}^{\boldsymbol{z}}_{\{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{L}}\}}$ is the measurement Jacobian matrix of measurement $\boldsymbol{z}$ w.r.t. the estimated states. The computed information matrix is then marginalized to keep only the pose states $\hat{\boldsymbol{T}}_{\text{WI},\boldsymbol{k}}$ of the selected keyframes and the positions $^{\text{W}}\hat{\boldsymbol{l}}_{\boldsymbol{i}}$ of the selected landmarks. The marginalized information matrix is the Schur's complement of the marginalized variables:

$$\boldsymbol{\mathcal{I}}_{\text{D}} = \boldsymbol{\mathcal{I}}_{\boldsymbol{\mathcal{Z}}}(\mathcal{T}_{\text{S}}, \mathcal{L}_{\text{S}}) = \boldsymbol{\mathcal{I}}_{\text{SS}} - \boldsymbol{\mathcal{I}}_{\text{SM}} \boldsymbol{\mathcal{I}}_{\text{MM}}^{-1} \boldsymbol{\mathcal{I}}_{\text{MS}} \tag{5}$$

where the subscripts S and M respectively denote the selected and the marginalized variables, and $\mathcal{T}$ and $\mathcal{L}$ the sets of keyframe poses and landmark positions.

6) Subgraph consistent sparsification: As suggested by the figure (3b), the resulting information matrix $\boldsymbol{\mathcal{I}}_{\boldsymbol{D}}$ is dense since it has been filled in with cross-correlation blocks between the remaining variables. The associated distribution $\mathcal{N}(\hat{\boldsymbol{m}}_{\boldsymbol{S}}, \boldsymbol{\mathcal{I}}_{\boldsymbol{D}}^{-1})$ is far too complex to be communicated as it is. Therefore, we take advantage on sparsification techniques to compute a set of uncorrelated relative pose and relative position Gaussian factors (that we call *virtual measurements*) connecting pose and landmark states in a simpler topology (as represented in figure (3c)), and which yield the same local estimate and the closest state information matrix in terms of information metrics. The virtual measurement information matrix $\boldsymbol{\Omega}_{\boldsymbol{S}}$ is computed as the solution of an optimization problem which minimizes the loss of information under a *consistency* constraint:

$$\begin{aligned} \hat{\boldsymbol{\Omega}}_{\text{S}} &= \underset{\boldsymbol{\Omega}_{\text{S}}\in\mathcal{D}_+}{\arg\min}\ \mathcal{D}_{\text{KL}}\left(\mathcal{N}(\hat{\boldsymbol{m}}_{\text{S}}, (\boldsymbol{J}_{\text{S}}^{\top}\boldsymbol{\Omega}_{\text{S}}\boldsymbol{J}_{\text{S}})^{-1}) || \mathcal{N}(\hat{\boldsymbol{m}}_{\text{S}}, \boldsymbol{\mathcal{I}}_{\text{D}}^{-1})\right) \\ &\text{subjected to } \boldsymbol{\mathcal{I}}_{\text{S}} = \boldsymbol{J}_{\text{S}}^{\top}\boldsymbol{\Omega}_{\text{S}}\boldsymbol{J}_{\text{S}} \leq \boldsymbol{\mathcal{I}}_{\text{D}} \end{aligned} \tag{6}$$

where $\mathcal{D}_{\text{KL}}$ denotes the Kullback-Leibler divergence, $\mathcal{D}_+$ is the set of block diagonal positive definite matrices which match to the desired sparsified topology and $\boldsymbol{\mathcal{I}}_{\text{S}}$ is the state information matrix derived from the computed factors. $\boldsymbol{J}_{\text{S}}$ is the virtual measurement Jacobian matrix:

$$\boldsymbol{J}_{\text{S}} = \frac{\partial\left(\{\boldsymbol{T}_{\text{B}_{i-1}\text{B}_i}\}_{i\in\mathcal{T}_S}, \{^{\text{Bo}}\boldsymbol{l}_i\}_{i\in\mathcal{L}_S}\right)}{\partial\left(\{\boldsymbol{T}_{\text{LB}_i}\}_{i\in\mathcal{T}_S}, \{^{\text{Bo}}\boldsymbol{l}_i\}_{i\in\mathcal{L}_S}\right)} \tag{7}$$

[1]A landmark is considered as well-constrained if it has enough observer keyframes, if is is not too close and too far from its closest observer keyframe and it the disparity angle of its observation bearing vectors is sufficient

which is a sparse block-diagonal matrix whose pattern is represented in the figure (4). The right (or local) relative pose Jacobian matrices are:

$$\boldsymbol{J}^{\boldsymbol{T}_{\mathrm{B}_i\mathrm{B}_j}}_{\{\boldsymbol{T}_{\mathrm{LB}_i},\boldsymbol{T}_{\mathrm{LB}_j}\}} = \begin{bmatrix} -\mathrm{Ad}^{-1}_{\boldsymbol{T}_{\mathrm{B}_i\mathrm{B}_j}} & \boldsymbol{I_{6\times 6}} \end{bmatrix} \tag{8}$$

where $\boldsymbol{Ad}_{(.)}$ is the adjoint matrix on SE(3). The consistency constraint ensures that the resulting sparsified distribution does not add any artificial information *ie* does not artificially reduces the joint entropy over any subset of variables. As proved in [6], since $\boldsymbol{J}_{\mathrm{S}}$ is invertible by construction, a solution to the unconstrained version of (6) is:

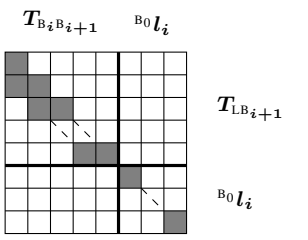


Fig. 4: Pattern of the Jacobian matrix $\boldsymbol{J_S}$

$$[\boldsymbol{\Omega}^*_{\mathrm{S}}]_i = (\{\boldsymbol{J}_{\mathrm{S}}\ \boldsymbol{\mathcal{I}}^{-1}_{\mathrm{D}}\ \boldsymbol{J}^{\top}_{\mathrm{S}}\}_i)^{-1} \tag{9}$$

where the subscript i denotes the i$^{\text{th}}$ diagonal block corresponding to the i$^{\text{th}}$ factor. We finally need to impose the consistency constraint from the solution of the relaxed problem. Classical methods [12] to solve problem (6), based on the solving of Semi-Definite Programming problems by interior point methods, which model the consistency constraint with log-det barriers, are intractable in real-time. We therefore use a non-optimal method to enforce this constraint with less complexity. Let $\boldsymbol{\lambda_{\min}}$ be the minimum eigenvalue of $\boldsymbol{\mathcal{I}}_{\mathrm{D}}$. Each eigenvalue $\boldsymbol{\lambda_i}$ of the computed blocks is compared to $\boldsymbol{\lambda_{\min}}$ and replaced by it if it exceeds it.

C. Packet exchange

Packet exchanges are handled in a separate thread. As described in figure (2), each robot periodically checks which robots are within communication range. For each spotted robot, it checks in its packet sending history the timestamp of its last successful packet sending to that robot. If new packets have been precomputed in the meanwhile, then they are sequentially sent to the other robot, and the communication history is updated as soon as a reception acknowledgement signal is received back from the recipient robot. Figure (3c) shows the content of each packet: it holds the means and the covariance matrices of the virtual measurements, plus the selected visual information (2D keypoints and descriptors) of each communicated vertex. Additionally, every measurement of the recipient robot associated to one of the selected keyframes (e.g. relative pose factors estimated from AprilTag [13] detection) is also transmitted.

D. Packet integration

Packet integration is handled by a third thread. Each received packet is first pre-aligned by connecting it to the previously received packet and to the robot trajectory thanks to the communicated inter-robot observations. Finally, loop closure algorithms are run in order to spot 2D-3D correspondences, using the communicated visual information on the selected keyframes. The found matches are then used to associate some received 2D keypoints and merge some communicated landmarks with the map landmarks.

E. Conclusion on the proposed method

As a summary, the FS-VIP method make robots regularly exchange data packets which summarize the successive portions of their map. It is inspired from the method developed by Paull *et al.* [9] for acoustic underwater SLAM which we adapted for the visual-inertial framework. The main differences we have introduced are the selection of the landmarks and keyframes for which ones local visual information is appended. In [9], landmarks are scarce and merged on a distance criteria while visual-inertial landmarks are much more numerous and should be merged based on visual matching. This also makes the sparsification process heavier. To bound its computation load, we pre-compute standard packets, while in [9], the sparsity of the underlying posegraph allows to compute them specifically for each single exchange. Finally, another difference is that in [9], each packet comes with an inter-robot relative range measurement computed from the time of flight.

IV. Min-k-Cover based-Selection for Visual-Inertial Packets (MKCS-VIP)

A. Overview of the proposed method

The method proposed in this section is based on the selection of landmarks and raw measurements to compute data packets that will be exchanged between robots. An overview of the MKCS-VIP method is given by the figure (5). All the process mentioned in this figure are detailed below.

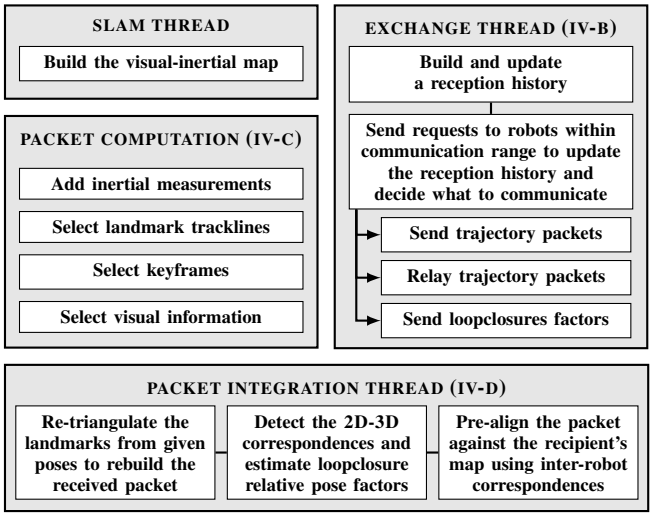


Fig. 5: Overview of MKCS-VIP

B. Exchange handler thread

We make each robot $\mathrm{R_X}$ hold a *reception* history (recall that it was a *sending* history in FS-VIP). For each robot $\mathrm{R_Y}$, it stores the timestamp $t_{\mathrm{Y}\to\mathrm{X}}$ of the most recent information it holds from robot $\mathrm{R_Y}$, may it have been received directly from robot $\mathrm{R_Y}$ or relayed by a third robot $\mathrm{R_Z}$. We assume that the whole data of $\mathrm{R_Y}$ anterior to $t_{\mathrm{Y}\to\mathrm{X}}$ which had to be

communicated has been successfully transmitted to $\mathrm{R_X}$ and that the clocks of both robots are synchronized.

We use the following method to keep the history up-to-date: each robot $\mathrm{R_X}$ periodically sends an update request to the other robots $\mathrm{R_Y}$ it can reach. $\mathrm{R_Y}$ returns the vector $(t_{\mathrm{Z}\to\mathrm{Y}})_{\mathrm{R_Z}\in\mathcal{R}}$ where $\mathcal{R}$ is the set of all robots. Based on this information, it checks if it can send some data about trajectory of any robot $\mathrm{R_Z}$ (including itself) such that $t_{\mathrm{Z}\to\mathrm{X}} \geq t_{\mathrm{Z}\to\mathrm{Y}} + t_{\min}$ where $t_{\mathrm{X}\to\mathrm{X}}$ is confounded with the current time and $t_{\min}$ is user-defined. If it does, an exchange is triggered about the data of $\mathrm{R_Y}$ posterior to $t_{\mathrm{Z}\to\mathrm{Y}}$. If the exchange boils down to relaying received data, then such data can be sent as it is. However, if $\mathrm{R_X}$ must send some information about its own trajectory, it is additionally required than a minimal distance of $d_{\min}$ has been run since the last exchange, and then the packet is computed as described below.

C. Packet computation

Contrary to FS-VIP, packets do not summarize the successive portions of the trajectory by computing some virtual measurement factors, but by selecting the best landmarks and visual-inertial factors which can best help to faithfully and consistently re-estimate it, as well as some localized visual information to spot inter-robot loop closures.

1) Inertial factors: First, we must provide some inertial information to make the scale factor observable. As a first option, we can communicate all the bunches of raw IMU measurements between the successive keyframes over the sub-trajectory (6 doubles per measurement). A possibly lighter option regarding the communication load is to send some pre-integrated IMU measurements [14]. Such measurements are relative orientation, velocity and position factors derived from the propagation of the raw IMU measurements between vertices $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. They are respectively denoted:

$$\boldsymbol{\Delta R}_{\mathrm{ij}}, \boldsymbol{\Delta v}_{\mathrm{ij}}, \boldsymbol{\Delta p}_{\mathrm{ij}} \sim \mathcal{N}\left((\boldsymbol{\Delta}\hat{\boldsymbol{R}}_{\mathrm{ij}}, \boldsymbol{\Delta}\hat{\boldsymbol{v}}_{\mathrm{ij}}, \boldsymbol{\Delta}\hat{\boldsymbol{p}}_{\mathrm{ij}}), \boldsymbol{\Sigma}_{\boldsymbol{\Delta}}\right) \quad (10)$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}}$ is the covariance matrix resulting from the IMU noise propagation. Those factors are derived assuming constant accelerometer and gyrometer biases ${}^{\mathrm{B}}\hat{\boldsymbol{b}}_{\boldsymbol{a}i}$ and ${}^{\mathrm{B}}\hat{\boldsymbol{b}}_{\boldsymbol{g}i}$ between t_i and t_j. The pre-integrated measurements can incorporate bias updates, projected as first order corrections via the Jacobian matrices. Therefore, each transmitted pre-integrated measurement should include the above estimates, covariance and Jacobian matrices One complete pre-integrated IMU measurement needs 87 doubles. Thus, sending pre-integrated IMU measurements is relevant as soon as there are on average more than about 15 IMU measurements between successive keyframes.

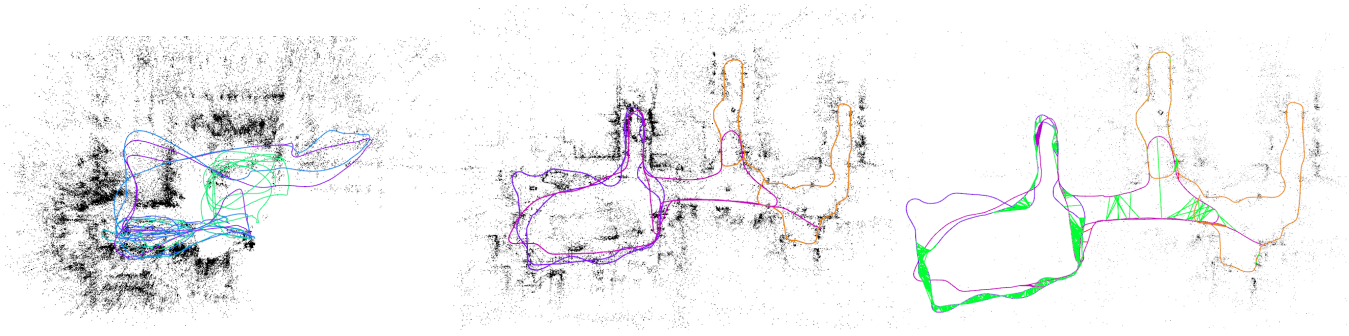
2) Visual factors: The inertial measurements need to be completed with some visual factors so that the trajectory is properly constrained. We thus add the visual factors associated to observations of a selected subset of landmarks. Those landmarks are selected using a heuristic approximation the solution to a MIN-K-COVER problem in order to mutually constraint the most keyframes with the minimum of landmark observations. The aim is to select the minimal subset of landmarks such that each keyframe observes at least n_{mkc} landmarks i.e. such that each keyframe is *covered*. The greedy heuristic involves to select the landmark which covers the most not fully covered keyframes until all keyframes are fully covered. At the end, some keyframes may be over-covered. Therefore, we iterate over the keyframes and remove the extra-observations associated to the most observed selected landmarks, paying attention not to make it badly-constrained. The associated 2D keypoints are transmitted without any visual descriptor.

3) Visual information: The selected visual-inertial factors allow to re-estimate the considered sub-trajectory. As in the FS-VIP, we complete them with some visual information localized on selected keyframes. To take loop closures into account and avoid to communicate redundant visual information, we build and update a covisibility graph to support their selection. An edge is added between two keyframes if their commonly observed landmark count and ratio exceed the respective user-defined thresholds $n_c^{\lim}$ and $r_c^{\lim}$. When a new keyframe is added to the map, the covisibility graph is updated, and the keyframe is selected if it is not adjacent to a previously selected keyframe. For each selected keyframe, we select on its visual frame the $n_{keypoints}$ keypoints associated to the most observed and constrained landmarks within the map, with their visual descriptor, as described in III-B.4.

4) Loop closures and inter-robot observations: The relative pose factors derived from loop closures and inter-robot observations are valuable information to share, and are included in the communicated packet if their detection timestamp is relevant w.r.t. the reception history of the recipient robot. In order to preserve the map consistency, it is crucial that their associated covariance matrices are properly estimated. A loop closure factor between two keyframes $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is estimated from the spotted 2D-3D correspondences. It involves solving two *Perspective-n-Point* (PNP) problems within a RANSAC scheme to estimate the global poses of both keyframes w.r.t. to their spotted commonly observed landmarks and thus derive their relative pose. Note that since the 3D structure is needed to spot loop closures, each robot is the only one to be able to detect loop closures against its trajectory from received 2D visual information.

D. Packet integration

As a summary, the final packet includes all the raw (or pre-integrated) inertial measurements along the communicated sub-trajectory, the relative poses between successive vertices (needed for initialization), the visual observations associated to the selected track-lines (communicated without visual descriptors), some visual information localized on some selected keyframes (with the visual descriptors) as well as the loop closure and the inter-robot observation factors spotted in the meantime. The received packet is chained to the previously received ones, connected to other trajectories through inter-robot observations and loop closures are searched between the received packet and the recipient robot's trajectory. Note that the IMU and camera models



(a) EUROC trajectories with trajectories MH1, MH2, MH3, MH4 and MH5

(b) SOTTEVILLE trajectories with sequences SV1 (dark purple, left), SV2 (orange, right) and SV3 (magenta, middle)

(c) Merged map of robot S1 in scenario 4 with aligned trajectories of robots S2 and S3 using MKCS-VIP

Fig. 6: Groundtruth trajectories of the datasets and example of map merging using MKCS-VIP

of the sender robot should be communicated at their first encounter. Each robot can consistently estimate the trajectory of the other robots, and such estimation can provide a base to register further data, such as local point-clouds asked on request.

E. Conclusion on the proposed method

Compared to the FS-VIP, MKCS-VIP make robots exchange more data but brings multiple advantages. First, the communicated packets are far less complex to build since they only involve the selection and the extraction of relevant data. A second advantage is that by sending raw measurements or factors directly derived from their propagation, the resulting packet is consistent and is not likely to be as conservative as the ones built in FS-VIP. Note also that all the communicated factors are relinearizable. Finally, a third advantage is that it is possible to relay data between robots and complete them with loop closure and inter-robot observation factors without any risk of *double counting*.

V. PERFORMANCE EVALUATION

A. Considered test scenarios

1) Used datasets: Both presented methods are evaluated on multi-robot scenarios built on the 5 aerial sequences from the EUROC dataset [15] and from 3 terrestrial sequences of a homemade dataset we acquired, which we call the SOTTEVILLE dataset. Table I gives some descriptive parameters of each robot trajectory, represented on figures (6a) and (6b).

Seq.	Length [m]	Duration [s]	ATE [m]	ARE [deg]	Scale	Scenarios 1	2	3	4
EUROC dataset									
MH1	76	149	0.225	0.023	0.127	✗	✗		
MH2	69	183	0.240	0.020	0.122		✗		
MH3	132	87	0.216	0.032	0.064	✗			
MH4	92	87	0.364	0.023	0.122	✗		✗	
MH5	100	98	0.253	0.021	0.509			✗	
SOTTEVILLE dataset									
S1	145	350	0.318	0.052	0.005				✗
S2	122	284	1.013	0.083	0.159				✗
S3	158	323	0.937	0.073	0.148				✗

TABLE I: Parameters of the sequences from the EUROC and SOTTEVILLE datasets and estimation accuracy of the trajectory estimation with VINS-MONO [16]

2) Multi-robot scenarios: The trajectories represented on figures (6a) and (6b) are estimated from the raw visual-inertial measurements with VINS-MONO [16] and carried into the MAPLAB framework [11] to be merged together (i.e. loop-closed and optimized). The resulting trajectory estimates are the best achievable when knowing the full set of visual-inertial measurements of all the involved robots. We consider them as groundtruth estimates and rely on them to simulate the inter-robot observations in the testing scenarios. We consider 4 multi-robot scenarios built using some subset of sequences taken from the EUROC and the SOTTEVILLE datasets, as summarized in table I. Scenario 1 confronts three disparate trajectories from the EUROC dataset, while the trajectories involved in scenarios 2 and 3 are similar and prone to inter-robot loop closures and observations. Scenario 4 involves the 3 trajectories from the SOTTEVILLE dataset. Note that while trajectory S3 is prone to loop closures and inter-robot observations with trajectories S1 and S2, this is not the case between those last two.

B. Method parameters

Table II gives the values of the user-defined parameters introduced throughout sections III and IV and used in the evaluation multi-robot scenarios.

FS-VIP parameters		MKCS-VIP parameters	
$d_{\min}$	3 m	$d_{\min}$	3 m
$n_c^{\lim}$	20	n_{mkc}	20 landmarks
$r_c^{\lim}$	20%	$n_c^{\lim}$	20 landmarks
$n_{keypoints}$	100	$r_c^{\lim}$	20%
$n_{landmarks}$	100	$n_{keypoints}$	100

TABLE II: Parameters used during replay

C. Performance metrics

On each scenario, each tested method is evaluated regarding i) the accuracy of the estimation of its own trajectory and of the robots from which ones it has received data; ii) the communication load and iii) its matching performances regarding inter-robot loop closure detection. We classically use Absolute Root Mean Squared Errors (RMSE) for translation (ATE), rotation (ARE) and scale computed using [17]) to assess the estimation accuracy. For each robot, we evaluate

Seq.	ATE [m]	ARE [deg]	Scale	Seq.	ATE [m]	ARE [deg]	Scale
Scenario 1				Scenario 4			
MH1	0.199	0.006	0.175	**S1**	0.054	0.003	0.001
MH3	0.111	0.008	0.092	**S2**	0.448	0.005	0.075
MH4	0.186	0.030	0.094	**S3**	0.220	0.005	0.100
Scenario 2				Scenario 3			
MH1	0.201	0.008	0.171	**MH4**	0.186	0.030	0.094
MH2	0.266	0.020	0.329	**MH5**	0.230	0.023	0.449

(a) Absolute RMSE metrics of the robots' own trajectories estimation when using FS-VIP

Seq.	ATE [m]	ARE [deg]	Scale	Seq.	ATE [m]	ARE [deg]	Scale
Scenario 1				Scenario 4			
MH1	0.343	0.005	0.451	**S1**	0.234	0.012	0.014
MH3	0.101	0.018	0.091	**S2**	0.504	0.028	0.083
MH4	0.147	0.010	0.121	**S3**	0.241	0.004	0.114
Scenario 2				Scenario 3			
MH1	0.203	0.004	0.193	**MH4**	0.121	0.012	0.080
MH2	0.078	0.017	0.063	**MH5**	0.230	0.024	0.448

(b) Absolute RMSE metrics of the robots' own trajectories estimation when using MKCS-VIP

Seq.	FS-VIP			MKCS-VIP		
	ATE [m]	ARE [deg]	Scale	ATE [m]	ARE [deg]	Scale
Scenario 1						
MH1←MH3	0.528	0.100	0.199	0.119	0.024	0.115
MH1←MH4	0.693	0.083	0.202	0.167	0.017	0.106
MH3←MH1	0.485	0.071	0.498	0.096	0.010	0.040
MH3←MH4	0.488	0.063	0.174	0.154	0.016	0.152
MH4←MH1	0.834	0.098	0.856	0.093	0.010	0.039
MH4←MH3	0.852	0.178	0.551	0.114	0.023	0.105
Scenario 2						
MH1←MH2	0.298	0.066	0.585	0.054	0.010	0.024
MH2←MH1	0.487	0.071	0.781	0.054	0.009	0.037
Scenario 3						
MH4←MH5	0.306	0.073	0.210	0.135	0.014	0.097
MH5←MH4	0.272	0.097	0.039	0.327	0.031	0.090
Scenario 4						
S1←S2	—	—	—	0.913	0.122	0.282
S1←S3	0.943	0.166	0.306	0.871	0.011	0.176
S2←S1	—	—	—	0.207	0.014	0.004
S2←S3	1.665	0.387	0.387	0.455	0.011	0.115
S3←S1	1.310	0.178	0.022	0.186	0.010	0.004
S3←S2	0.787	0.122	0.154	0.679	0.036	0.295

(c) Absolute RMSE metrics regarding the reconstruction of the trajectories of other robots

Seq.	FS-VIP			MKCS-VIP		
	ATE [m]	ARE [deg]	Scale	ATE [m]	ARE [deg]	Scale
Scenario 1						
MH1←MH3	0.327	0.010	0.159	0.500	0.017	0.151
MH1←MH4	0.779	1.190	0.187	0.416	0.368	0.131
MH3←MH1	0.997	0.108	0.754	0.221	0.019	0.128
MH3←MH4	0.553	0.040	0.875	0.247	0.026	0.069
MH4←MH1	0.900	0.033	0.431	0.226	0.029	0.063
MH4←MH3	0.821	0.046	0.264	0.235	0.017	0.080
Scenario 2						
MH1←MH2	0.272	0.063	0.583	0.240	0.010	0.074
MH2←MH1	0.774	0.060	0.875	0.140	0.028	0.087
Scenario 3						
MH4←MH5	0.329	0.075	0.040	0.225	0.027	0.099
MH5←MH4	0.365	0.015	0.125	0.412	0.062	0.138
Scenario 4						
S1←S2	—	—	—	1.101	0.090	0.051
S1←S3	2.043	0.015	0.061	1.015	0.043	0.097
S2←S1	—	—	—	1.148	0.086	0.023
S2←S3	2.019	0.269	0.304	1.221	0.049	0.275
S3←S1	1.383	0.264	0.132	0.352	0.012	0.037
S3←S2	0.403	0.008	0.032	1.782	0.028	0.134

(d) Absolute RMSE metrics of the inter-robot relative pose trajectories estimation

Seq.	# Vertices	# Landmarks	# Observations
MH1	38 (4.3%)	2413 (14.5%)	3671 (0.5%)
MH2	32 (4.7%)	2074 (16.1%)	3022 (0.6%)
MH3	82 (9.3%)	4209 (31.2%)	5801 (1.2%)
MH4	38 (6.2%)	2482 (22.9%)	3608 (1.1%)
MH5	41 (6.4%)	2671 (22.8%)	3838 (1.1%)
S1	45 (2.0%)	3089 (23.1%)	3828 (0.03%)
S2	45 (2.6%)	2509 (25.2%)	3193 (0.04%)
S3	57 (2.8%)	3588 (24.4%)	4551 (0.05%)

(e) Mean number and percentage of communicated items for FS-VIP

Seq.	# Landmarks	# Visual factors	# selected keyframes
MH1	290 (1.7%)	17,522 (2.8%)	51 (5.8%)
MH2	207 (1.6%)	13,395 (2.8%)	40 (5.9%)
MH3	752 (5.6%)	17,573 (3.8%)	105 (11%)
MH4	303 (2.8%)	12,263 (3.7%)	47 (7.6%)
MH5	293 (2.5%)	12,661 (3.6%)	50 (7.6%)
S1	301 (2.3%)	43,989 (4.3%)	34 (1.5%)
S2	378 (3.8%)	34,095 (4.7%)	40 (2.3%)
S3	486 (3.3%)	40,164 (4.3%)	65 (3.2%)

(f) Mean quantities and percentage of communicated items over the scenarios for MKCS-VIP

Scenario 1		Scenario 4	
MH1←MH3	80 (2.0%)	**S1←S2**	–
MH1←MH4	216 (8.7%)	**S1←S3**	62 (1.7%)
MH3←MH1	69 (2.8%)	**S2←S1**	–
MH3←MH4	242 (9.7%))	**S2←S3**	125 (3.4%)
MH4←MH1	66 (2.7%)	**S3←S1**	159 (5.1%)
MH4←MH3	30 (0.7%)	**S3←S2**	78 (3.1%)
Scenario 2		Scenario 3	
MH1←MH2	686 (33%)	**MH4←MH5**	708 (26%)
MH2←MH1	806 (33%)	**MH5←MH4**	453 (18%)

(g) Number and percentage of merged landmarks among the ones received for FS-VIP

Scenario 1		Scenario 4	
MH1←MH3	10	**S1←S2**	0
MH1←MH4	0	**S1←S3**	2
MH3←MH1	13	**S2←S1**	0
MH3←MH4	2	**S2←S3**	3
MH4←MH1	1	**S3←S1**	1
MH4←MH3	8	**S3←S2**	5
Scenario 2		Scenario 3	
MH1→MH2	34	**MH4→MH5**	39
MH2→MH1	64	**MH5→MH4**	51

(h) Number of detected inter-robot loop closures using the received visual information with MKCS-VIP

TABLE III: Tables of results from the simulation on the EuRoC and Sotteville datasets

the absolute RMSE of its own trajectory estimation, the absolute RMSE of the trajectories of the other robots it reconstructed, and also the absolute RMSE of the trajectories of inter-robot relative poses to assess the registration of each reconstructed map w.r.t. to the recipient robot's trajectory.

D. Evaluation of both methods

The tables IIIa and IIIb respectively give the *a posteriori* absolute RMSE for each robot estimating its own trajectory after having merged the information received from the other robots. The table IIIc shows for both methods, and for every robot in each scenario the *a posteriori* RMSE for each robot estimating the trajectories of the other robots from which it has received data. The table IIId displays the RMSE of the relative pose trajectories, estimated by the recipient robot, between sender and recipient robots. The tables IIIe and IIIf respectively detail the average amount of items each robot has communicated over the evaluated scenarios for both tested methods. Finally, the tables IIIg and IIIh quantify the number of inter-robot matchings (number of spotted loop closures or merged landmarks). In those tables, $i \leftarrow j$ means that the related metric is associated to the trajectory of robot j estimated in the map of robot i.

First, the tables IIIa and IIIb show that fusing the information provided by the other robots can help the recipient robot to improve accuracy of the estimation of its own trajectory. When looking at tables IIIc and IIId, it comes that MKCS-VIP performs better than FS-VIP in terms of estimation accuracy of the trajectories of the other robots. Actually, MKCS-VIP brings multiple advantages over FS-VIP. First, each robot can relay the information it got from other robots. As a consequence, in scenario 4, as shown on the figure 6c, while robots associated to sequences S1 and S2 never observe each other and never observe the same scene, the robot S3 is able to relay the information of S1 to S2 and vice versa with MKCS-VIP. Secondly, it allows to exchange the loop closure factors each robot has spotted against its own map 3D structure. As a consequence, the resulting maps are more constrained than with FS-VIP for which one the marginalization process prevents any subsequent update on the exchange data. Additionally, it turns out that FS-VIP yields very conservative sparsified factors, such that many landmarks merges are needed (which is hardly reached as shown in table IIIg) so that the spotted inter-robot loop closures sufficiently constraint the trajectories of the robots.

VI. Conclusion

In this article, we introduced two decentralized data sharing methods based on the exchange of visual inertial packets built: from factor sparsification techniques (FS-VIP) and from landmark and raw measurement selection (MKCS-VIP). The performances of both methods have been investigated on two datasets. MKCS-VIP presents better results and takes advantages of exchanged and relayed data to build a complete map of the environment. Even if FS-VIP presents some limitations mainly due to the used consistent sparsification technique, it still has interesting results and could be useful for asymmetric computer power situation as this method is less heavy on the receiver side. As a perspective, we plan to conduct a more extensive testing on a wider dataset, possibly with conjoint aerial and terrestrial robots. This should bring more confidence on method parameters tuning and the analysis of their influence on the estimation accuracy and the communication load. Finally, both methods should be validated on real embedded systems.

VII. Acknowledgements

We would like to thank the *Société Nationale des Chemins de Fer* (SNCF) for granting us the access to the warehouse where the above mentioned SOTTEVILLE dataset was acquired, as well as the *Direction Générale de l'Armement* (DGA) for financing this work.

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] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3962–3970.

[3] M. Karrer, P. Schmuck, and M. Chli, "Cvi-slam collaborative visual-inertial slam," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2762–2769, 2018.

[4] M. J. Schuster, C. Brand, H. Hirschmüller, M. Suppa, and M. Beetz, "Multi-robot 6d graph slam connecting decoupled local reference filters," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 5093–5100.

[5] L. Contreras, O. Kermorgant, and P. Martinet, "Efficient decentralized collaborative mapping for outdoor environments," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 56–63.

[6] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term slam," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 50–72, 2016.

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

[8] M. T. Lazaro, L. M. Paz, P. Pinies, J. A. Castellanos, and G. Grisetti, "Multi-robot slam using condensed measurements," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1069–1076.

[9] L. Paull, G. Huang, M. Seto, and J. J. Leonard, "Communication-constrained multi-auv cooperative slam," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 509–516.

[10] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[11] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1418–1425, 2018.

[12] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM journal on matrix analysis and applications*, vol. 19, no. 2, pp. 499–533, 1998.

[13] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[14] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation." Georgia Institute of Technology, 2015.

[15] 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, 2016.

[16] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[17] 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.