HAL
open science

WeCo-SLAM: Wearable Cooperative SLAM System for Real-time Indoor Localization Under Challenging Conditions

Viachaslau Kachurka, Bastien Rault, Fernando Israel Ireta Muñoz, David Roussel, Fabien Bonardi, Jean-Yves Didier, Hicham Hadj-Abdelkader, Samia Bouchafa, Pierre Alliez, Maxime Robin

To cite this version:

Viachaslau Kachurka, Bastien Rault, Fernando Israel Ireta Muñoz, David Roussel, Fabien Bonardi, et al.. WeCo-SLAM: Wearable Cooperative SLAM System for Real-time Indoor Localization Under Challenging Conditions. IEEE Sensors Journal, 2022, 22 (6), pp.5122–5132. 10.1109/JSEN.2021.3101121 . hal-03609471

HAL Id: hal-03609471
https://hal.science/hal-03609471

Submitted on 30 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

WeCo-SLAM: Wearable Cooperative SLAM System for Real-Time Indoor Localization Under Challenging Conditions

Viachaslau Kachurka, Bastien Rault, Fernando I. Ireta Muñoz, David Roussel, Fabien Bonardi, Jean-Yves Didier, Hicham Hadj-Abdelkader, Samia Bouchafa, Pierre Alliez, and Maxime Robin

Abstract—Real-time globally consistent GPS tracking is critical for an accurate localization and is crucial for applica-tions such as autonomous navigation or multi-robot mapping. However, under challenging environment conditions such as indoor/outdoor transitions, GPS signals are partially available or not consistent over time. In this paper, a real-time tracking system for continuously locating emergency response agents in challenging conditions is presented. A cooperative local-ization method based on Laser-Visual-Inertial (LVI) and GPS sensors is achieved by communicating optimization events between a LiDAR-Inertial-SLAM (LI-SLAM) and Visual-Inertial-SLAM (VI-SLAM) that operate simultaneously. The estimation of the pose assisted by multiple SLAM approaches provides the GPS localization of the agent when a stand-alone GPS fails. The system has been tested under the terms of the MALIN Challenge, which aims to globally localize agents across outdoor and indoor environments under challenging conditions (such as smoked rooms, stairs, indoor/outdoor transitions, repetitive patterns, extreme lighting changes) where it is well known that a stand-alone SLAM will not be enough to maintaining the localization. The system achieved Absolute Trajectory Error of 0.48%, with a pose update rate between 15 and 20 Hz. Furthermore, the system is able to build a global consistent 3D LiDAR Map that is post-processed to create a 3D reconstruction at different level of details.

Index Terms—Computer vision, simultaneous localization and mapping, sensor fusion, indoor navigation, embedded software, terrain mapping.

I. Introduction

SLAM (Simultaneous Localization And Mapping) has been widely studied in the fields of robotics and computer vision. The robustness and accuracy of a SLAM method in unknown dynamic environments can hardly be achieved using only one sensor. The paradigm between computational power and availability of new sensors have lead to obtain large-scale dense maps by developing Multi-SLAM methods that cooperate to improve localization and to speed up registration of a detailed 3D representation of the environment. Applications such as autonomous multi-robot (MR-SLAM) navigation [1]–[3], parallel indoor/outdoor 3D registration [4], and multi-session mapping [5], [6] are very active fields in the literature.

MR-SLAM and multi-SLAM techniques focus mainly on two issues:

1) How the poses and the map are shared between the systems.
2) How the trajectories and the global map are corrected and updated.

By taking the main advantages of sensor fusion while performing SLAM, we propose a Wearable Cooperative SLAM system (WeCo-SLAM). Although Wearable, this system can easily be adapted to mobile robotics. The main contribution of the system is its ability to perform two complementary SLAM

This work was supported in part by the Localisation 3D (LOCA3D) Project in the Framework of the Challenge MAîtrise de la Localisation INdoor (MALIN) through the Direction Générale de l'Armement (DGA) and in part by French National Research Agency—https://challenge-malin.fr/. The associate editor coordinating the review of this article and approving it for publication was Dr. Valérie Renaudin. *(Corresponding author: David Roussel.)*

Viachaslau Kachurka, David Roussel, Fabien Bonardi, Jean-Yves Didier, Hicham Hadj-Abdelkader, and Samia Bouchafa are with IBISC, Université d'Évry, Université Paris-Saclay, 91020 Évry-Courcouronnes, France (e-mail: viachaslau.kachurka@ibisc.univ-evry.fr; david.roussel@ibisc.univ-evry.fr; fabien.bonardi@ibisc.univ-evry.fr; jean-yves.didier@ibisc.univ-evry.fr; hicham.hadj-abdelkader@ibisc.univ-evry.fr; samia.bouchafa@ibisc.univ-evry.fr).

Bastien Rault and Maxime Robin are with Innodura TB, 69603 Villeurbanne, France (e-mail: bastien.rault@innodura.fr; maxime.robin@innodura.fr).

Fernando I. Ireta Muñoz and Pierre Alliez are with the TITANE Project-Team, INRIA Sophia Antipolis—Méditerrannée Research Centre, 06902 Valbonne, France (e-mail: fernando.ireta-munoz@inria.fr; pierre.alliez@inria.fr).

approaches simultaneously, while communicating optimization events (such as loop closures and global pose adjustment) for a real-time 6DOF pose estimation process. Each SLAM method is improved by performing an independent tightly-coupled sensor fusion (LI-SLAM [7] and VI-SLAM [8], respectively). Finally, a global loosely-coupled sensor fusion between the obtained poses and the registered GPS positions allows to predict valid GNSS coordinates in indoor environments. This strategy allows the system to maintain a valid trajectory under arbitrary and unpredictable challenging conditions such as smoke-filled or dark rooms, indoor/outdoor transitions, extreme lighting conditions or even agent-crawling. Our system is an extension of previous works [9], [10] that have been proposed and tested under the environment constraints of the MALIN Challenge [11], which aims to localize emergency response agents in real-time under highly dynamic environments when GPS signals are missing or just partially available and also to provide a detailed map of the environment afterwards.

Sensor fusion approaches can be seen here as strategies that perform localization by jointly minimizing the error function between correspondences from different measurements while generating an aligned map. Fusion can be tightly-coupled or loosely-coupled based on the dependency between the sensors for pose estimation. In practice, Bayesian approaches use one of the sensors (e.g. an inertial measurement unit, IMU) to predict the pose whereas other sensors, camera(s) or LiDAR, are used to correct this estimation. The estimated pose is used for merging 3D points (e.g. LiDAR scans) to generate a single global consistent map.

Furthermore, pose estimation from multiple sensors can also be obtained by optimizing the 6 DOF pose over two (or more) individual SLAM approaches that are performed simultaneously to build a common 3D map. Depending on how the 3D map is built, these strategies can be classified here as collaborative or cooperative SLAM. The main difference between them relies on the contribution of each SLAM for performing both, odometry and 3D mapping.

Collaborative SLAM approaches [12]–[20] perform localization and mapping with an increased number of sensors (by using same SLAM approach and sensor-type) while optimizing over individual 3D maps to build a global map. These approaches are mostly employed for robot team mapping, where identical robots share and update the same global 3D map from different locations and simultaneously correct the poses of all robots over time. Moreover, in autonomous driving applications, the global mapping aims to match and merge camera-based and LiDAR-based maps either by performing coarse alignments between them or by correlating the extrinsics between the sensors. The corresponding feature points are put into bundle adjustment [21] to refine all camera poses.

Cooperative SLAM strategies [9], [10], [22]–[25] benefit from the complementarity of visual, LiDAR and/or IMU sensors. Indeed, a stand-alone SLAM approach might not be robust enough to maintain an accurate pose under arbitrary conditions. The 3D maps of each SLAM are not necessarily shared but their global alignment is assisted by the pose of each SLAM. Cited cooperative methods achieved better localization performance by combining visual and LiDAR approaches, where one approach can compensate the lack of robustness of the other while estimating the pose. For instance Shao *et al.* [22] compensate possible fails of LiDAR SLAM with the assistance of Visual SLAM, whereas Balazadegan *et al.* [23] initiate pose estimation with Visual SLAM then the pose is refined using the Generalized ICP over LiDAR scans. Zhang and Singh [24] follow the same principle as [23], with IMU predictions for visual-inertial odometry. This method allows to partially or totally bypass failure modes of one sensor and combine the rest to maintain robustness, by introducing the concept of reconfigurable pipeline between range, vision and IMU measurements. Zuo *et al.* approach [25], [26] performs an online multi-sensor calibration for maintaining accuracy and robustness. W.r.t to other cited methods, Zuo *et al.* propose a tightly coupled fusion of all IMU, visual and range data, but it lacks a loop closure detection stage. Other cited methods are then considered as loosely coupled since poses can be predicted by IMU and then corrected by either visual or LiDAR SLAMs.

To the best of our knowledge, real-time cooperative SLAM under challenging environmental conditions has received little attention. Hence, a method that propagates optimization events between two different SLAM architectures is proposed here. The 3D map is built using an improved version of LI-SLAM [24], [27] while the final pose is assisted by a VI-SLAM [8], [28]. Besides, a RANSAC optimization process on temporally paired positions ensures registration of SLAM and GNSS reference frames. A Kalman filter (KF) fuses coordinates retrieved in the global frame and predicts the position in GNSS-denied situations.

The key contributions of this paper are:

1) A new communication strategy of optimization events between two independent SLAM approaches.
2) A GPS coordinates prediction method using sensor fusion.
3) An optimized real-time LiDAR-based SLAM approach (20Hz).

This paper is organized as follows: Section II introduces the conditions and constraints, which bound the considered agent localization problem. Section III presents a general view of our proposal, both in terms of hardware and software. Section IV further develops the idea of SLAM cooperation, detailing our technical choices for the used approaches, as well as the fusion of the results. Then, Section V shows the obtained results while performing real-time localization using our cooperative SLAM approach and a post-processing offline 3D LOD reconstruction mapping. Finally, the paper ends with a conclusion.

II. Agent Localization Problem: Related Work

The agent localization problem was first formulated in [29] as a problem of accurately localizing military or emergency response agents in an unknown GNSS-denied environment. This problem formulation covers 16 essential user requirements, including: localization accuracy; physical robustness; real-time map building capability in unknown environments; weight, cost and energy consumption efficiency.

(a) Dark environment (b) High luminosity (c) Narrow passages (d) Smoke (e) Crawling (f) Discordance

Fig. 1. Different challenging situations experimented during real-life-like scenarios of MALIN Challenge.

While none of the up to date solutions have strictly met all the 16 requirements, a survey on localization and indoor positioning systems (IPS) for emergency responders [30] counts more than 30 IPSs developed by 2017; and a meta-review of surveys [31] conducted in 2019 describes more than 150 articles concerning "indoor positioning OR indoor localization". Our approach can be classified as "device based" and "infrastructure free" as it uses its own embedded sensors without the need of deploying dedicated beacons.

This is also the scope of the MALIN challenge of the French National Research Agency, which aims to provide a real-time indoor localization solution without using existing infrastructures such as Wi-Fi or cellular networks under challenging conditions. Such conditions are also known as "non-cooperative environment" and are described in [32] – as an environment where the conditions induce failures of both sensors and software. Some of the encountered difficulties are illustrated in Fig. 1. Among these we can find difficult lighting conditions (1a & 1b), narrow passages associated with erratic movements (1c & 1e), smoke (1d), and situations inducing contradictions between sensors measurements or algorithms (1f).

Therefore, among the aforementioned constraints and requirements, various issues can be identified, such as:

- *Real-time tracking and mapping* – the current localization should be available (and possibly transmitted to control center) in real-time, whereas complete localization trajectory and reconstructed map can be retrieved and processed later.
- *Transitions between indoor and outdoor environments* with potentially very different scales, therefore imposing difficulties on mapping and data management.
- *Tracking failures* due to erroneous or missing data, human-specific motion or environment factors.
- *Efficiency* in cost, weight, volume and energy consumption.

III. System Design and Architecture

The system presented hereafter is the result of an iterative process. So even if we only present the most recent prototype, several relevant conclusions, drawn from the testing of previous prototypes, are also mentioned. Our localization system can be described as a wearable multi-sensor, multi-SLAM, cooperation-based, real-time localization system. The study of this paper has been validated on, but is not limited to the hardware described in Fig. 2.

Fig. 2. Tactical waistcoat hardware configuration.

The hardware setup is installed on a tactical waistcoat, where the LiDAR, camera and inertial sensors are placed on the right shoulder, the controller on the back with the GNSS antenna, whereas the battery and phone-based UI are placed on the front. The autonomy of the system is up to 1 hour while running SLAM. The embedded PC used to run both SLAMs and fusion is a Neousys POC-545, 3.35 GHz CPU clock with 12 cores and 16 GB memory. The weight of the overall system (tactical waistcoat, embedded PC, frame, cables, sensors, battery) is just under 9 kg.

The following paragraphs describe the choices we made in terms of hardware and software.

A. Sensors

1) Camera: Situations of total darkness lead us to consider the use of an on-board illuminator associated with cameras. The conditions of smoke and unknown lighting eliminate the choice of a conventional RGB-camera. Most types of smoke (either artificial dry smoke [33] or several types of "natural" smoke [34]) are transparent to infrared imaging sensors. An obvious choice would be to use SWIR (0.9–1.7 μm) cameras which are relatively expensive, but these are also sensitive to the absence of lighting [35] and therefore require the use of an expensive illuminator both in terms of price and consumption.

Our final choice is an Intel RealSense sensor D435i, which supplies two NIR (0.7–1.0 μm) cameras (enabling stereo vision), and an embedded IMU (enabling visual-inertial approaches) associated with a NIR light emitter composed of two underpowered diodes (with a power demand $\leq$4 W). Even if NIR cameras do not perform as well as SWIR cameras in

smoke, our tests have shown the ability to extract features from their images for objects up to 4 m away.

Intel's D435i extrinsics parameters between camera(s) and internal IMU have been calibrated using Kalibr [36].

2) LiDAR: The rapid and non-planar movements of the wearer require the use of a 3D LiDAR to obtain a robust pose with a LiDAR-based SLAM. In indoor environments, LiDAR captures data ranging mostly below twenty meters. The vertical aperture of the sensor must be large enough for the algorithm to detect geometric features. This is why we opt for the widely used LiDAR, VLP-16 Puck from Velodyne, which has a 30° vertical aperture.

We also tried the RS-BPearl from Robosense with a 90° vertical opening. This LiDAR is more suitable on the stairs, but less in hallways, where it only sees walls and ceiling. Since camera-based SLAM mostly handle stairs well, but encounter difficulties in long corridors, we chose the VLP-16 LiDAR.

3) Inertial Measurement Units: The system is intended to be worn by a person, so we have chosen an inertial unit suitable for low frequencies. The RAD6-M from Texense, based on gas technology, is designed for low frequency measurements. Its low noise and its bandwidth are suitable for this application, considering the nature of human movement.

4) GNSS Unit: GNSS unit is a Navilock NL-8004U Multi GNSS receiver used with Galileo, GPS and Glonass constellations at an update rate of 1 Hz and a Circular Error Probability (CEP) of 2.5 m.

B. Software

On one hand, we chose a LiDAR-based SLAM in order to obtain 3D maps as dense as possible that we can use for the cartography post-processing. However, denser 3D maps represent a large volume of data which can lead to issues regarding the real time constraint for such SLAM. On the other hand, we chose to use a Visual SLAM as it can provide robust loop closures and relocalization based on Bag of Words signatures [37]. Visual SLAMs also use 3D maps (which are usually associated with keyframe poses), but these maps are relatively sparse compared to LiDAR 3D maps and do not provide additional 3D information. We therefore chose not to use them for the final cartography.

Due to their intrinsic nature, both SLAMs can have difficulties in specific situations, such as narrow places with a LiDAR-based SLAM featuring a 30° vertical aperture which prevents from picking up 3D points on the floor or ceiling, thus impacting the vertical motion estimation. Regarding the Visual SLAM, the main difficulty comes from the visual features tracking: either lack of features which can occur in long hallways with uniform walls, floors and ceilings, or erroneous features such as a moving shadow on a wall (see Fig. 1f) or stationary points in the field of view. Since both SLAMs can fail in certain situations, we established a cooperation framework (described in Fig. 3) in order to:

- Fuse both SLAM poses with GNSS data (when available) into a single UTM[1] pose.

[1]Universal Transverse Mercator

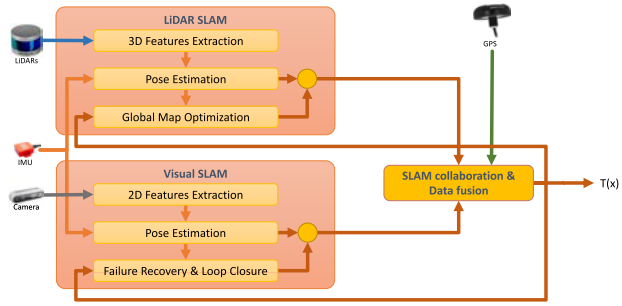


Fig. 3. The general scheme for our system architecture. Sensors data are processed by two semi-independent SLAMs. The relative positioning is then fused with existing GPS data by Kalman filter.

- Propagate Loop Closure events from Visual SLAM to LiDAR-based SLAM.
- Provide a recovery pose after tracking failure of either SLAM in order to maintain current navigation trajectory and map.

The following paragraphs describe the internals of each SLAM and the following section further develops the cooperation framework. As mentioned previously, the multi-sensor data are processed with corresponding software modules.

1) InnoSLAM: InnoSLAM is a LiDAR-Inertial SLAM based on the Kitware [38] implementation of LOAM [27]. It was first designed for non-real time applications and relies on the comparison of 3D features between consecutive pointclouds. Geometric information are first extracted by looking at the roughness of successive points along each channel of the LiDAR. Points are classified into three groups: planar, edge and unused. Each category of feature is matched with the corresponding category of the previous pointcloud by a (Levenberg-Marquardt) LM-ICP [39] algorithm, providing a first estimation of the motion between two measurements. Features are then matched in the same way to the internal feature map. This motion refinement step not only greatly reduces the drift, but also keeps a globally optimized map updated with new features in each pointcloud. Real time SLAM is obtained by reducing computation time. To do so, apart from using multithreaded code, the map is constantly filtered with a varying density depending on whether the system is in a wide or narrow environment. The map also has a fixed size and moves with the system. Points, that are too far away, are deleted. This is described in detail in one of our previous contributions [9].

The quality indicators described in section IV-D and the cooperation with VI-SLAM allow new strategies to improve the performance of InnoSLAM. One strategy we have implemented is to reset the local map when InnoSLAM has a bad quality indicator whilst VI-SLAM maintains a good quality indicator. This keeps the global map consistent and error-free. Another new strategy consists in relocating InnoSLAM with previously recorded pose whenever VI-SLAM sends a loop closure event. This can be used to detect whether InnoSLAM drifted or not, and if it did, we can rework the map to keep its consistency.

2) Visual SLAM: Challenging conditions impose several constraints onto our choice of a Visual SLAM. Unknown light

conditions, which can be partially corrected by automatic gain control, prevent accurate photometric calibration and thus the usage of a direct visual SLAM (such as DSO [40] or ROVIO [41]). The real-time processing condition primarily implies a sparse SLAM in the context of moderate resources – therefore, our choice is among indirect sparse SLAMs.

A monocular vision-only SLAM would be very fast, but has an undetermined scale factor (each SLAM run yields a new map and trajectory scale). This can be solved either by using stereo or 3D camera or by combining with IMU measurements – thus by applying a stereo, mono-inertial or stereo-inertial SLAM.

Among other desired properties, robustness is required to cope with eventual tracking failures. Therefore we need loop closure and failure recovery mechanisms. Feature points usage can provide a reliable recognition of previously visited locations using popular state-of-the-art bag-of-words (BoW) approaches [37], providing a possibility of relocalization or loop closure.

To fullfill the characteristics described above two families emerge as popular choices: ORB-SLAM [42], ORB-SLAM2 [42] and more particularly its visual-inertial successors VI-ORB-SLAM [43] and ORB-SLAM3 [44] as well as VINS-Mono [28] and its successor VINS-Fusion [8]. All of these implementations are "optimization-based": they use a form of bundle adjustment to solve a non-linear optimization problem in order to minimise relative or absolute errors between estimated position and observed data. The previous iteration of our visual SLAM (as described in [9]) was based on VI-ORB-SLAM [43].

Both families can employ visual-inertial data, provide some form of failure recovery and error correction strategies, and are reported to be real-time SLAMs. But the biggest difference between them lies in the keyframe management during optimization: ORB-SLAM family algorithms try to optimize a full keyframe connectivity graph, which can take considerable time in long running scenarios; while VINS family algorithms optimize only a sliding window of recent keyframes, thus employing "marginalization" [45]. While first type of optimization provides better precision, the second allows real time constraints even in long running scenarios. However, marginalizations can lead to pose drift.

Our tests on the aforementioned prototype with the same pool of resources show that using VI-ORB-SLAM along with LI-SLAM on the same embedded PC requires too much resources to respect real-time processing boundaries; while both monocular visual-inertial versions of VINS can respect the same boundaries. The marginalization precision tradeoff, found in VINS family algorithms, can be compensated by SLAM cooperation and data fusion. VINS family also has a computational benefit of tracking GFTT [46] points using an optical flow approach instead of tracking feature points like ORB-SLAM does with ORB [47] points. Such tracking approach, although more primitive, is faster and less sensitive to repetitive patterns. However, VINS-Fusion also uses BRIEF [48] feature points to create keyframes used for loop closures.

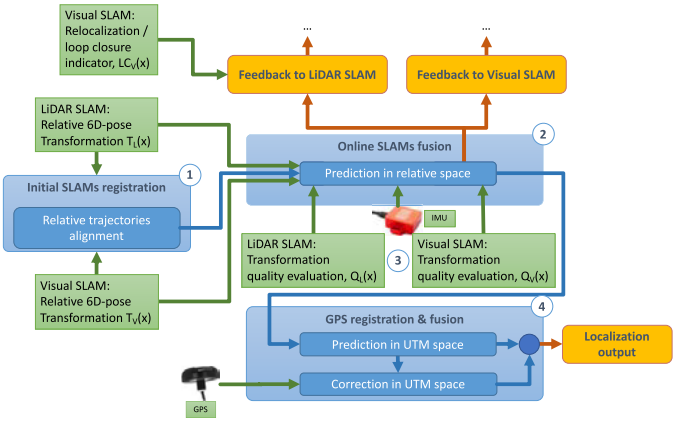


Fig. 4. Multi-SLAM cooperation scheme. Green blocks designate input data, yellow blocks - output data, blue blocks - processing modules.

IV. SLAM Cooperation and Data Fusion

Fig. 4 describes the main blocks in our localization system. Main stages can be identified: 1) Initial alignment of the trajectories of the two SLAMs; 2) Fusion of the poses between LI-SLAM and VI-SLAM while integrating IMU data, that can be used as an external pose when an individual (or both) SLAM method fails; 3) Tracking quality indicators of each SLAM used to merge their results; 4) Global fusion with GPS data.

A. Initial SLAMs Registration

The relative transformations between LiDAR and Visual SLAM, $T_L(x)$ and $T_V(x)$, respectively, are provided at first in their own spatial and temporal reference frames, which are determined during the system's initialization stage.

Extrinsics between LiDAR and the camera are estimated by aligning trajectories using Horn's method [49]. Particularly, the alignment is performed at the beginning of each mission once each independent SLAM method has reached an established threshold distance (estimated up to 10 m during our tests). The alignment between absolute poses is achieved by minimizing the distance between correspondences found by the closest timestamps. The resulting transformation is used for estimating relative pose between LI-SLAM and VI-SLAM.

B. Online SLAMs Fusion

With this first alignment, estimated positions and orientations from both SLAMs are properly conditioned in the same reference frame and can be fused with the IMU data by a Kalman filter method. For this purpose, we use an Error State Kalman Filter (ESKF) for a robust pose estimation as described in [50] and inspired from [51]. Compared to a traditional Kalman filter or Extended Kalman filter option, the state vector is not populated with the systems parameters to estimate, but the errors associated to each parameter and spawned by following prediction and correction steps. Such change involves an additional step in the algorithm after correction in order to apply the estimated error to the system state vector and the error state vector is then reset.

This adjustment is said to be more efficient because an error state vector is generally small and values to estimate are closer to the linearization point involved by the model [50]. Our error-state vector is defined as $\tilde{\mathrm{x}} = [\ \tilde{\mathrm{p}}^\intercal\ \delta\theta^\intercal\ \tilde{\mathrm{v}}^\intercal\ \tilde{\mathrm{b}}_a^\intercal\ \tilde{\mathrm{b}}_g^\intercal\ ]$, where $\tilde{\mathrm{p}}^\intercal$ and $\tilde{\mathrm{v}}^\intercal$ are the errors in position and velocity in the global frame, $\delta\theta^\intercal$ is the angular error expressed as a quaternion in the global frame, and $\tilde{\mathrm{b}}_a^\intercal(t)$ and $\tilde{\mathrm{b}}_g^\intercal(t)$ are the accelerometric and gyroscopic biases errors. This vector is estimated at each IMU acquisition using 4th order Runge-Kutta numerical integration, and then corrected at each SLAM iteration.

C. GPS Registration and Fusion

Positions and orientations estimations, made in the relative space, are fused with GPS data. This processing block implies two steps.

First, positions from SLAMs fusion are projected onto latitude-longitude plane according to the extrinsic rotation parameters from the initial SLAMs registration (section IV-A). Therefore, this projection is highly dependent on that initial calibration although a slight rotation offset can be compensated later by the Kalman process. Positions from SLAMs fusion blocks are then temporally paired with coordinates given by the GPS, and a RANSAC process is performed on these pairs to estimate a transformation between the relative map (map used by the SLAMs fusion process) and the UTM reference (global map associated to the GPS coordinates).

Then, a Kalman filter is applied on the positions estimated in the global map. The considered state vector here includes position and heading in the 2-dimensions UTM map $\tilde{\mathbf{x}}_t = [\tilde{x}, \tilde{y}, \tilde{h}]^T$. Prediction is achieved with estimated positions from the SLAMs and corrected according to GPS data. Quality of the GPS data is evaluated thanks to the geometric dilution of precision measure given by the GNSS. This indicator is used as a weighting factor in the Kalman correction step and GPS data are even rejected if dilution is higher than a given threshold. Covariance matrices, implied in these processes, are weighted by quality indicators detailed below in order to benefit from the most reliable source of information and weaken data tainted with error and uncertainty. In GNSS-denied situations, classic GPS-provided correction is not possible, and quality indicators are used to decide the correction step.

Thus, we employ the aforementioned quality indicators which reflect quantified SLAM's tracking self-assessment, $Q_L(x)$ and $Q_V(x)$.

D. SLAM Tracking Quality Self-Assessment

A SLAM result quality indicator should be a measure of SLAM's inner "confidence" about its most recent result. If we can find a common ground to calculate such an indicator for both visual and LiDAR SLAMs, they can become comparable, and thus we can quantify the quality level of the SLAMs' respective results. We define three states of SLAMs' tracking quality which represent their self-estimate risk of error: *low*, *medium* and *high*. These states are then used in the Online SLAMs Fusion: data provided by one SLAM in *low* error-risk state is thought to be more reliable than data provided by another SLAM in *medium* or *high* error-risk state.

Both SLAMs employed in our system, InnoSLAM and VINS-Fusion, are using non-linear minimisation problem solving [52] as the main method of finding the optimal solution between estimated pose and observed data. We can therefore use these already existing minimization processes to extract the elements necessary for our quality indicators without additional computations.

1) InnoSLAM: InnoSLAM feature types, Edge and Planar points, are matched with its corresponding map by Point-to-Line ICP and Point-to-Plane ICP respectively. Minimizing the error functions of these ICP provides the transformation between two pointclouds, and can be represented by the equation:

$$\mathcal{T}^*(x) = \arg\min_{\mathcal{T}(x)} \sum_{i=1}^{N} \|(\mathrm{Q_L}_i^{\omega} - \mathrm{Q_L}^*)^\intercal\ \mathrm{A}\ (\mathrm{Q_L}_i^{\omega} - \mathrm{Q_L}^*)\|^2 \in SE(3) \quad (1)$$

where $\mathrm{Q_L}_i^{\omega} = \mathrm{R(x)Q_L}_i + \mathrm{T(x)}$ are the transformed feature points, $\mathrm{A} = (\mathbb{I} - \mathrm{EE}^\intercal)$ for Edge points and $\mathrm{A} = \mathrm{EE}^\intercal$ for Planar points, where E is the eigenvector of the covariance matrix $\mathrm{Q_L}^*$. The evolution of this matrix reflects the quality of LiDAR SLAM's result according to each degree of freedom. We then used empiric thresholds in translation T(x) and rotation R(x) to estimate the current state of InnoSLAM. We analysed InnoSLAM behaviour in different situations to select a set of consistent thresholds, we defined $Q_{Lt} < 0.025\,\mathrm{m}^2$ and $Q_{Lr} < 0.05\,\mathrm{rad}^2$ as the *low* risk state, and $Q_{Lt} > 0.1\,\mathrm{m}^2$ and $Q_{Lr} > 0.2\,\mathrm{rad}^2$ as the *high* risk state. The in-between is the *medium* risk state.

2) VINS-Fusion: In the case of VINS-Fusion, the non-linear problem concerns finding such parameters that the sum of residual errors is minimised. The problem formulation for the visual-inertial bundle adjustment (equation and notation adopted from [53, eq. 26]) is:

$$\mathcal{X}_k^* = \arg\min_{\mathcal{X}_k}(||\mathbf{r}_0||^2_{\mathbf{\Sigma}_0} + \sum_{(i,j)\in\mathcal{K}_k} ||\mathbf{r}_{\mathcal{I}_{ij}}||^2_{\mathbf{\Sigma}_{ij}} + \sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i} ||\mathbf{r}_{\mathcal{C}_{il}}||^2_{\mathbf{\Sigma}_C}) \quad (2)$$

where $\mathcal{I}_{ij}$ represent the set of IMU measurements acquired between keyframes at times i and j; $\mathcal{C}_{il}$ represent image measurement for landmark l in keyframe at time i. $\mathbf{r}_0$, $\mathbf{r}_{\mathcal{I}_{ij}}$ and $\mathbf{r}_{\mathcal{C}_{il}}$ are the residual errors associated to the measurements, and $\mathbf{\Sigma}_0$, $\mathbf{\Sigma}_{ij}$ and $\mathbf{\Sigma}_C$ are the corresponding covariance matrices. The residual errors can be seen as functions of system state $\mathcal{X}_k$, which quantify the mismatch between measured quantity and the predicted value of this quantity given $\mathcal{X}_k$.

However, these covariance matrices are decomposed and hidden inside the optimization framework, and their extraction requires additional computations. Therefore, the covariance-based definition of states (like for InnoSLAM) cannot be used for VINS-Fusion. Instead we can use the resulting residual errors, considering that the better the minimisation process went (the lower are the residual errors), the more reliable are the estimated values in the optimized state $\mathcal{X}_k^*$.

Therefore, one can employ a synthetic function to quantify the optimization result quality, using the residual errors and

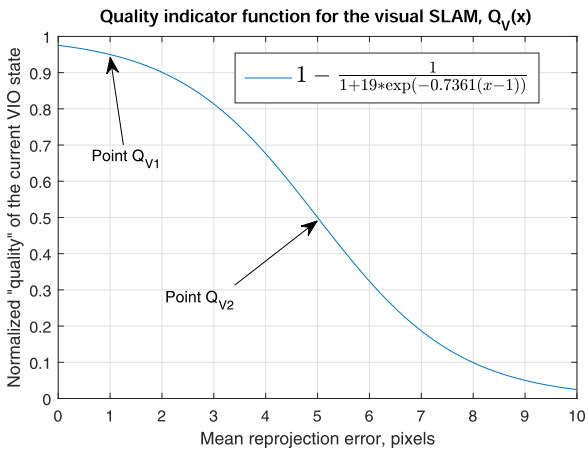


Fig. 5. Synthetic function, producing quantitative self-evaluation of the quality of visual SLAM's state. Points Q_{V1} (1; 0.95) and Q_{V2} (5; 0.5) serve as references to enable the computation of function's parameters.

their evolution as input, several thresholds for the output value define the corresponding error risk state of the SLAM: low residual error should yield high quality value (and *low* error risk state), and vice-versa.

The choice of such a synthetic function is a matter of heuristic and is guided by several criteria:

- It should be monotonically decreasing: larger residual errors should lead to lower quality values;
- It should have at least one inflection point in order to have three distinguishable regions: gradual descent for "*low*" error-risk state, steep slope for "*medium*" error-risk state, and a decreasing slope for "*high*" error-risk state.

Richard's curve, known as generalized logistics function, is a suitable candidate due to its three distinguishable regions (notation partly adopted from [54]):

$$Q_V(x) = 1 - \frac{1}{1 + D \times \exp(-B \times (x - M))} \tag{3}$$

where x represents the mean camera-landmark residual error $\overline{\mathbf{r}_{\mathcal{C}_{il}}}$, — or, in more simple terms, reprojection errors for the features observed in keyframes; and coefficients D, B and M are used only to parametrize the curve.

The parameter values are dependent on the choice of two reference points, Q_{V1} and Q_{V2} (see Fig. 5), for which we can set our arbitrary values. E.g., for our most recent prototype, we consider that any mean reprojection error less than 1 pixel would imply that the system is very confident in the quality of its results, providing first reference point $Q_V(1) = 0.95$. However, if the mean reprojection error is higher than 5 pixels, the quality assessment should reflect this uncertainty, providing the second reference point $Q_V(5) = 0.5$. The computation of parameters with two arbitrary reference points, $Q_{V1} = (\lambda_1, \alpha_1)$; $Q_{V2} = (\lambda_2, \alpha_2)$ in this case, becomes trivial:

$$M = \lambda_1, \;\; D = \frac{\alpha_1}{1 - \alpha_1}, \quad B = \frac{\ln D - \ln \frac{\alpha_2}{1-\alpha_2}}{\lambda_2 - \lambda_1}$$

We empirically define quality states of VINS-Fusion as: $Q_V > 0.75$ for *low* risk of error, $Q_V < 0.40$ for *high* risk of error, and the in-between for the *medium* risk of error.

The states of each SLAM are updated on each new data input to ESKF, and different approaches can be used to keep a robust pose estimation. First, we ignore the data from a SLAM in *high* error-risk state if the other is in a more reliable state, so it does not disturb ESKF process. Second, these indicators are also used to reinitialize InnoSLAM's maps if it encounters errors while visual SLAM is in a stable state, in order to keep InnoSLAM's maps consistent. And third, whenever both SLAMs are in the *medium* risk state, the covariance matrix of the higher risk value SLAM is boosted to favor the SLAM with the lowest risk of error in the fusion process.

E. Loop Closure and External Pose Indication

Visual SLAMs are known for their relocalization/loop closure technique: confident recognition of previously visited places, which can be used to rectify trajectories and even recover from tracking failures. However, the robust real-time loop closure detection task remains an open problem for LiDAR-based SLAM (see survey [55] on the topic). Therefore, each time a loop detection in visual SLAM happens, we can generate an event with corresponding data, $LC_V(x)$ (set, consisting of differential 6D-pose $\Delta p_{i/j}^{WL}$ between matched keyframes $\mathcal{C}_i$ and $\mathcal{C}_j$, converted to InnoSLAM's relative world reference frame, as well as the indications of matched keyframes — i and j keyframe indices and timestamps), to be sent to InnoSLAM.

From another point of view, a weak point in visual SLAM is the failure recovery strategy: relocalization is not always possible, and auto-resetting the tracking usually destroys the map and previous trajectory. One can imagine a "re-initialization" procedure – after visual tracking is lost, a new tracking should be initialized automatically based on previously measured motion model while preserving the already acquired trajectories and maps and hence improving the system's tolerance to failures. However, such an approach [10] is mostly suitable for ORB-SLAM-like approaches, where tracking failure happens due to lack of matched features. In VINS-like approaches, tracking failure happens when the bundle adjustment diverges and compensates the errors by abnormally large biases or velocities. In this case, the last known motion model is already erroneous, and cannot be used as a motion hypothesis during re-initialization. Thus, we can use externally provided data (the last known fusion's output 6D-pose, converted to visual SLAM's relative reference frame, ${p_i}^{WV}$) as a re-initialization starting pose, while leaving previous trajectory and map unchanged, allowing future loop closures.

V. Results

To evaluate the accuracy of our system, we have planned a trajectory passing through waypoints identified and geolocated on a cadastral map. Such waypoints have an uncertainty of about 30 cm in longitude and latitude and about 10° in orientation with respect to the true North. The trajectory includes difficulties not managed by state-of-the-art SLAM algorithms, like crawling in a hallway on several meters,

walking sideways in front of a uniform wall, climbing stairs in total darkness forward and backward, carrying an object in the field of view of sensors (making it a static object in a dynamic environment), running and jumping off stairs, and indoor/outdoors transitions.

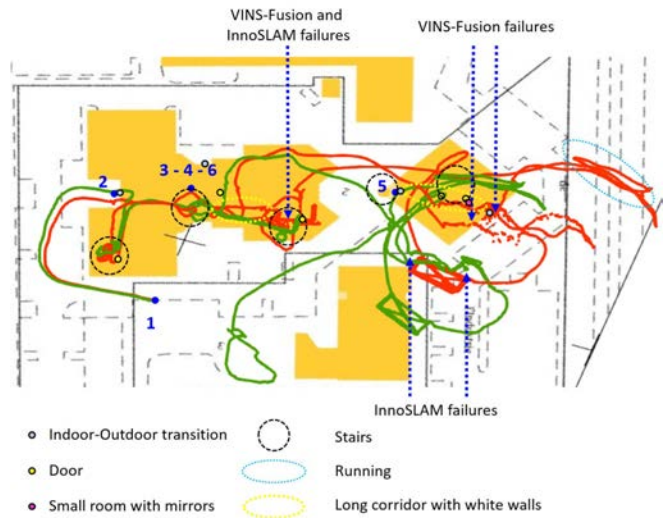
Fig. 6a reflects InnoSLAM and VINS-Fusion trajectories on a real-time 950 m run bounded by 4 geolocated waypoints, numbered from 1 to 6 according to the order these points were crossed during the run. Waypoints 3, 4 & 6 represent the same location crossed three times with distinct orientations. The followed path features indoor / outdoor transitions, doors crossings, long hallways with uniform walls and floors, stairs with various lighting conditions and a small room with mirrors. Both SLAMs failed several times during the run. VI-SLAM failed three times in long hallways, most notably when pointing a fake weapon in front of the camera creating more stationary points than the few feature points on the walls, floor or ceiling. It's first fail was recovered after climbing stairs in which InnoSLAM first failed because of the lack of horizontal features in the stairs compared to the vertical features on the walls. VI-SLAM second hallway fail happened before entering a small room with mirrors on walls, where InnoSLAM drifted by several degrees. In this case InnoSLAM most likely failed due to the exiguity of the room rather than the presence of mirrors since minimum range of the LiDAR was set to 50 cm. InnoSLAM encountered no issue during the last hallway fail of VI-SLAM, but failed while being outdoor. This can happen when few geometric features are available, like in an open field. InnoSLAM got an error of several degrees in orientation due to this fail.

Due to multiple orientations, only one loop closure at waypoint (3-4-6) was detected during the run shown in Fig. 6a and Fig. 6b. Multiple crossings of waypoints (2) and (5) happened at different levels of the building. Specific geometric patterns paths have been followed: an "L"-shaped pattern can be seen between waypoints (3-4-6) and (5), and another one composed of 2 right triangles can be seen below waypoint (5). These geometric patterns can be used to assess position and orientation drifts.

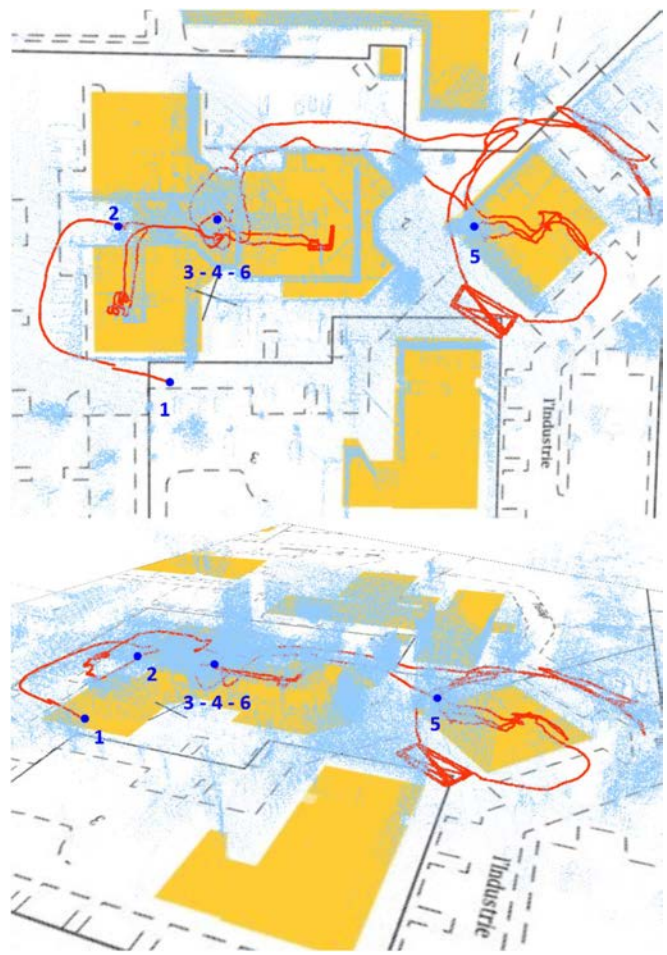
Fig. 6b illustrates results of the real-time fusion between SLAMs on the run. One can notice that none of the individual SLAMs trajectories are accurately estimated whereas the fusion process produces a consistent trajectory.

A. Online Localization

To assess the estimation quality, we adopted from [49] the absolute trajectory error (ATE) without pre-alignment (due to usage of global reference frame) for pose estimations at each waypoint. Fig. 7 shows individual SLAMs, Fusion and GPS trajectory errors compared to geolocated waypoints (1) to (6). At the end of the run, we get an $ATE = 4.59$ m (see Fig. 7), which is 0.48% of the total traveled distance of 950 m, and never exceeds 1.1% during the run. The root mean square error (RMSE) is 3.14 m. Regarding orientation, we obtained an $ATE = 3.3°$ at the end and $RMSE = 8.46°$. Our previous work [9] used a much more resource heavy approach and mentioned similar results, with a positional ATE of 0.41% computed over a 185 m sub-section of the run featuring only in-building navigation and indoor / outdoor transitions. While errors accumulate over time and traveled distance, they may be corrected with loop closures.



(a) InnoSLAM (green) and VINS-Fusion (red) separate trajectories: view from atop



(b) Estimated trajectory (red) from Fusion process and detected map features (light blue): views from atop and in trimetric projection

Fig. 6. 950m run with several difficulties as presented over a cadastral plan (Buildings in yellow, other topographic lines in gray. Waypoints are indicated with blue dots).

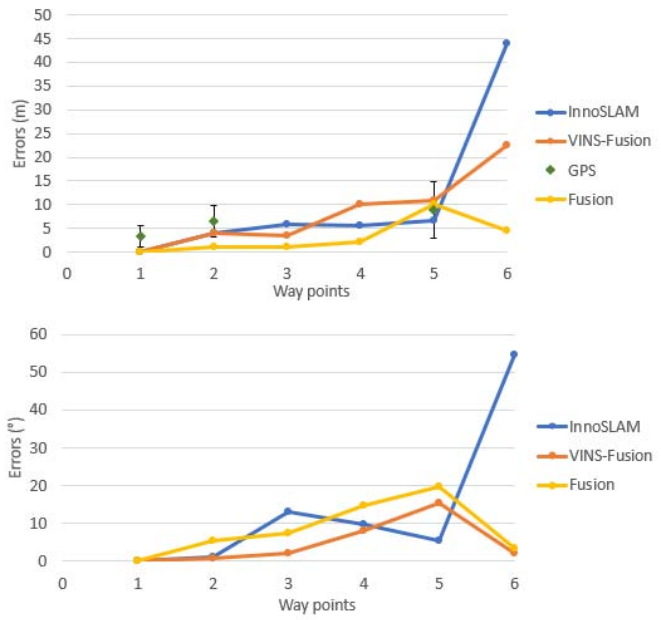


Fig. 7. Absolute Trajectory Errors (ATE) obtained before and after fusion, on estimated position and orientation, along a 950 meters trajectory based on known cadastral waypoint localization. GPS errors are also represented with their standard deviations when GPS was available.

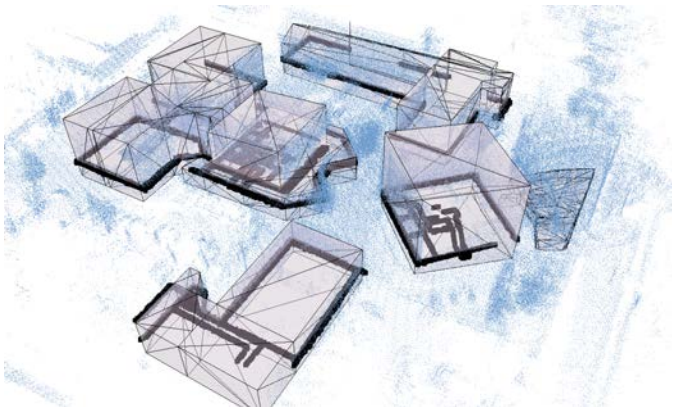


Fig. 8. Offline mapping of evaluation scenario with levels of details (up to LOD1), as seen in trimetric projection. Initial 3D pointclouds are represented in blue. Pink volumes show buildings' models reconstruction, and black lines shows partial indoor/outdoor floorplans.

Fusion update is triggered by the IMU at a 200 Hz rate, InnoSLAM mean correction rate occurs at 15 to 20 Hz, Visual-SLAM mean correction rate occurs at 8 to 10 Hz processing only 1 frame out of 3 to preserve computational resources. Due to the terms of the MALIN challenge, fusion pose is recorded to file at 5 Hz and radio-transmitted at 0.5 Hz.

B. Offline Mapping

Same as in [9], generated 3D pointclouds of visited buildings are post-processed to obtain a reconstruction at three different Level Of Details (LOD). For levels 0 (LOD0) and 1 (LOD1), a random forest classification method [56] is employed for segmenting planes in the pointcloud. Detected planes belongs to walls, ceilings and floors. For LOD0, 3D points belonging to walls are isolated and projected on their principal plane (estimated by PCA). For LOD1, we optimize over the intersection of all detected planes using the kinetic algorithm [57]. Finally, for level 2 (LOD2), we perform a triangulation over the 3D points using Poisson reconstruction. Most of the process has been automatized in order to respect the terms of the MALIN challenge to a 3D map reconstruction under 10 minutes. Therefore, we have performed LOD0 and LOD1 reconstruction levels for the results of this paper. In [9] we have presented an offline LOD2 reconstruction of a similar scenario where we could post-process the input pointclouds without a time limit constraint.

VI. Conclusion

In this work, we proposed a wearable cooperative SLAM system to localize emergency response agents in real-time across challenging environments and also produce a map of the visited environment. The first part of this article is dedicated to hardware and software studies in order to meet the requirements of the MALIN challenge in terms of indoor/outdoor navigation and mapping. A map, with high level of detail can be obtained through LiDAR acquisitions. We have chosen a cooperative approach between the SLAMs in order to obtain a robust navigation that can take advantage of the strengths of each approach as illustrated in the second part. Several points were investigated to ensure the best possible cooperation between LI-SLAM and VI-SLAM such as the propagation of loop closures, poses fusion assisted by quality indicators and fusion pose re-initialization assistance whenever one of the SLAMs fails. The system proposed in this article represents the current state of our incremental research. However, improvements are possible, in particular, concerning homogenization of the quality indicators, and also concerning the optimizations performed by each SLAM which could benefit from a common optimization to save computing resources. Mutual support between SLAMs can further be enhanced by providing them with the same capabilities such as loop closure which is still an open problem with LiDAR data.

References

[1] M. A. Abdelgalil, M. M. Nasr, M. H. Elalfy, A. Khamis, and F. Karray, "Multi-robot SLAM: An overview and quantitative evaluation of MRGS ROS framework for MR-SLAM," in *Robot Intelligence Technology and Applications 5*, J.-H. Kim *et al.*, Eds. Springer, 2019, pp. 165–183, doi: 10.1007/978-3-319-78452-6_15.

[2] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1243–1256, Dec. 2006, doi: 10.1177/0278364906072250.

[3] Z. Wang, S. Huang, and G. Dissanayake, "Multi-robot simultaneous localization and mapping using D-SLAM framework," in *Proc. 3rd Int. Conf. Intell. Sensors, Sensor Netw. Inf.*, 2007, pp. 317–322.

[4] G. Hardouin, J. Moras, F. Morbidi, J. Marzat, and E. M. Mouaddib, "Next-best-view planning for surface reconstruction of large-scale 3D environments with multiple UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 1567–1574. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03086499

[5] Z. Ma, K. Qian, W. Zhao, X. Ma, and H. Yu, "Multi-session mapping for indoor substation environment using a head-mounted RGB-D sensor," in *Proc. IEEE Int. Conf. Energy Internet (ICEI)*, May 2019, pp. 1–6.

[6] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. J. Leonard, "Real-time 6-DOF multi-session visual SLAM over large-scale environments," *Robot. Auton. Syst.*, vol. 61, no. 10, pp. 1144–1158, Oct. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889012001406

[7] (Mar. 2021). *InnoSLAM*. [Online]. Available: https://www.innodura.fr/challenge-malin-slam/

[8] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019, *arXiv:1901.03638*. [Online]. Available: http://arxiv.org/abs/1901.03638

[9] P. Alliez *et al.*, "Real-time multi-SLAM system for agent localization and 3D mapping in dynamic scenarios," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 4894–4900.

[10] P. Alliez *et al.*, "Indoor localization and mapping: Towards tracking resilience through a multi-SLAM approach," in *Proc. 28th Medit. Conf. Control Automat. (MED)*, Sep. 2020, pp. 465–470.

[11] (Mar. 22, 2021). *MALIN Challenge*. [Online]. Available: https://www.challenge-malin.fr

[12] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[13] S. Urban and S. Hinz, "MultiCol-SLAM—A modular real-time multi-camera SLAM system," 2016, *arXiv:1610.07336*. [Online]. Available: http://arxiv.org/abs/1610.07336

[14] S. Urban, S. Wursthorn, J. Leitloff, and S. Hinz, "MultiCol bundle adjustment: A generic method for pose estimation, simultaneous self-calibration and reconstruction for arbitrary multi-camera systems," *Int. J. Comput. Vis.*, vol. 121, no. 2, pp. 234–252, Jan. 2017.

[15] T. Schneider *et al.*, "MAPLAB: An open framework for research in visual-inertial mapping and localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1418–1425, Jul. 2018.

[16] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3D segment mapping using data-driven descriptors," in *Proc. 14th Robot., Sci. Syst.*, Jun. 2018, pp. 1–11.

[17] R. Dubé *et al.*, "SegMap: Segment-based mapping and localization using data-driven descriptors," *Int. J. Robot. Res.*, vol. 39, nos. 2–3, pp. 339–355, Mar. 2020.

[18] L. Riazuelo, J. Civera, and J. M. M. Montiel, "C2TAM: A cloud framework for cooperative tracking and mapping," *Robot. Auton. Syst.*, vol. 62, no. 4, pp. 401–413, Apr. 2014.

[19] J. Kuo, M. Muglikar, Z. Zhang, and D. Scaramuzza, "Redesigning SLAM for arbitrary multi-camera systems," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2020, pp. 2116–2122, doi: 10.1109/ICRA40945.2020.9197553.

[20] S. Golodetz, T. Cavallari, N. A. Lord, V. A. Prisacariu, D. W. Murray, and P. H. S. Torr, "Collaborative large-scale dense 3D reconstruction with online inter-agent pose optimisation," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 11, pp. 2895–2905, Nov. 2018, doi: 10.1109/TVCG.2018.2868533.

[21] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2004.

[22] W. Shao, S. Vijayarangan, C. Li, and G. Kantor, "Stereo visual inertial LiDAR simultaneous localization and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 370–377.

[23] Y. Balazadegan, S. Hosseinyalamdary, and Y. Gao, "Visual-LiDAR odometry aided by reduced IMU," *ISPRS Int. J. Geo-Inf.*, vol. 5, p. 24, Jan. 2016.

[24] J. Zhang and S. Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *J. Field Robot.*, vol. 35, no. 8, pp. 1242–1264, Aug. 2018.

[25] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-fusion: LiDAR-inertial-camera odometry," 2019, *arXiv:1909.04102*. [Online]. Available: http://arxiv.org/abs/1909.04102

[26] X. Zuo *et al.*, "LIC-fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5112–5119.

[27] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. Conf.*, vol. 2, Jul. 2014, pp. 1–9.

[28] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[29] J. Rantakokko, P. Händel, M. Fredholm, and F. Marsten-Eklöf, "User requirements for localization and tracking technology: A survey of mission-specific needs and constraints," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, Sep. 2010, pp. 1–9.

[30] A. F. G. Ferreira, D. M. A. Fernandes, A. P. Catarino, and J. L. Monteiro, "Localization and positioning systems for emergency responders: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2836–2870, 4th Quart., 2017.

[31] G. M. Mendoza-Silva, J. Torres-Sospedra, and J. Huerta, "A meta-review of indoor positioning systems," *Sensors*, vol. 19, no. 20, p. 4507, 2019.

[32] J. Rantakokko *et al.*, "Accurate and reliable soldier and first responder indoor positioning: Multisensor systems and cooperative localization," *IEEE Wireless Commun.*, vol. 18, no. 2, pp. 10–18, Apr. 2011.

[33] J. Schneider, E.-C. Koch, and A. Dochnahl, "Method of producing a screening smoke with one-way transparency in the infrared spectrum," U.S. Patent 6 484 640, Nov. 26, 2002.

[34] R. W. Bergstrom *et al.*, "Spectral absorption properties of atmospheric aerosols," *Atmos. Chem. Phys.*, vol. 7, no. 23, pp. 5937–5943, Dec. 2007.

[35] V. Kachurka, D. Roussel, H. Hadj-Abdelkader, F. Bonardi, J.-Y. Didier, and S. Bouchafa, "SWIR camera-based localization and mapping in challenging environments," in *Proc. Int. Conf. Image Anal. Process.* Cham, Switzerland: Springer, 2019, pp. 446–456.

[36] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 4304–4311.

[37] D. Galvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[38] (Mar. 2021). *Kitware*. [Online]. Available: https://www.kitware.fr/

[39] P. J. Besl and D. N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[40] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2016.

[41] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.

[42] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[43] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[44] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," 2020, *arXiv:2007.11898*. [Online]. Available: http://arxiv.org/abs/2007.11898

[45] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *J. Field Robot.*, vol. 27, no. 5, pp. 587–608, 2010.

[46] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 1994, pp. 593–600.

[47] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Washington, DC, USA, 2011, pp. 2564–2571.

[48] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer-Verlag, 2010, pp. 778–792.

[49] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[50] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation," in *Proc. AIAA Guid., Navigat., Control Conf.*, Aug. 2011, p. 6615.

[51] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.

[52] S. Agarwal *et al.* (Mar. 22, 2021). *Ceres Solver*. [Online]. Available: http://ceres-solver.org

[53] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[54] F. J. Richards, "A flexible growth function for empirical use," *J. Exp. Botany*, vol. 10, no. 2, pp. 290–301, 1959.

[55] S. Arshad and G.-W. Kim, "Role of deep learning in loop closure detection for visual and LiDAR SLAM: A survey," *Sensors*, vol. 21, no. 4, p. 1243, Feb. 2021.

[56] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 69–85, Aug. 2012.

[57] J.-P. Bauchet, "Kinetic data structures for the geometric modeling of urban environments," Ph.D. dissertation, École doctorale Sci. et Technol. de l'Inf. et de la Commun., Univ. Côte d'Azur, Inria, France, Dec. 2019. [Online]. Available: https://hal.inria.fr/tel-02432386/file/thesis_bauchet.pdf