Decentralized Collaborative Localization and Map Update with Buildings

Maxime Escourrou, Joelle Al Hage, Philippe Bonnifait

▶ **To cite this version:**

Maxime Escourrou, Joelle Al Hage, Philippe Bonnifait. Decentralized Collaborative Localization and Map Update with Buildings. 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2024), Oct 2024, Abu Dhabi, United Arab Emirates. hal-04680126

HAL Id: hal-04680126
https://hal.science/hal-04680126v1

Submitted on 28 Aug 2024

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.

Decentralized Collaborative Localization and Map Update with Buildings

Maxime Escourrou, Joelle Al Hage and Philippe Bonnifait

Abstract— **In urban environments where GNSS performance is degraded, localization can be performed using stable and geo-referenced map features detected by on-board sensors. Prior maps are prone to errors which have a direct impact on localization accuracy. By exchanging observed features and sharing their maps, vehicles can simultaneously improve their localization and update the map. This paper deals with indirect collaboration, where vehicles do not observe each other directly. The features are obtained from building facades using 3D lidar sensors. The paper emphasizes real-time decentralized collaboration with direct communication between vehicles, without the need for a central server. The collaboration takes place when vehicles perceive the same geo-referenced facades. Vehicle poses and maps are collaboratively updated using a Schmidt Kalman filter that carefully manages the cross-covariance terms. To maintain consistent estimates, the Kullback-Leibler Average is used. We also present a lidar data processing pipeline to obtain reliable observations from building facades. Real tests carried out with experimental vehicles on the university campus are reported. The results show that indirect collaboration makes a significant contribution to localization and map update when compared to a standalone method.**

I. Introduction

Accurate and reliable localization is one of the fundamental tasks for safe navigation of autonomous vehicles. In urban areas, Global Navigation Satellites System (GNSS) is not sufficient to meet the localization requirements due to multipath signals and obstructions [1]. In such environments, vehicles can rely on stable and geo-referenced cartographic characteristics detected by on-board sensors.

For multiple vehicles, Collaborative Localization (CL) has demonstrated its ability to improve and robustify localization [2]–[4]. CL is usually based on inter-vehicle measurements that correlate the estimates of vehicle poses [5], [6]. In this case, vehicles must be in the same field of view, which limits the collaboration range. Collaborative Simultaneous Localization and Mapping (C-SLAM) allows constructing a common map while performing localization at the same time [7]–[9]. Few studies on SLAM approaches with a prior map can be found in the literature, such as [10]. Indeed, some methods focus solely on localization within an existing map without modifying it, while others construct their map without prior information.

The objective of this work is to improve the localization of multiple vehicles through indirect measurements by observing common landmarks in the form of building facades while updating collaboratively the map. Buildings are very stable landmarks and among the most easily observable in urban environments. They are nowadays very often referenced in maps (e.g., OpenStreetMap – OSM) and can be efficiently detected using 3D lidar measurements [11].

Collaboration can be performed through different architectures. The centralized architecture is easy to implement but suffers from high computational cost since each vehicle must send information to a central unit that performs the processing. A decentralized architecture involves a direct communication between vehicles. The system is then scalable and more robust to faults. However, it may suffer from overconfidence that results from the reuse of the same information resulting from communication loops (data incest). The channel filter allows the convergence to a consistent global estimate in a decentralized way, while allowing interrupted communications [12], [13]. However, this method suffers from high computation cost where the state of the whole system should be estimated. Another strategy based on Covariance Intersection filter is well adapted to decentralized data fusion but leads to pessimistic solutions [14], [15]. Schmidt Kalman Filter (SKF) is well adapted for decentralized architecture with communication constraints. Indeed, it allows estimating only a part of the state vector, while considering the impact of the non-estimated part [16]–[18].

Exchanging point clouds for collaboration requires massive sharing of information between vehicles. Feature extraction is instead performed to retrieve the key elements for localization and map update.

Feature extraction from lidar point clouds has been studied in many works. It allows operating on key elements of the environment, ensuring a faster association, and a reduced sensitivity to noise. For instance, the Lidar Odometry and Mapping (LOAM) extracts two types of features: corners and surfaces [19]. In this method, only geometric feature points are used for scan-to-map registration, which may not be sufficient for proper registration in some environments. In [20], E-LOAM was developed to give some context to the extracted features and optimize the registration step. For this purpose, the method combines LOAM with Normal Distributions Transform (NDT) [21]. Using OSM, the authors in [22] propose the use of descriptors in the form of a vector of distances each associated with an angle. Features in the form of descriptors are offered in SegMatch [23] that represent geometric characteristics and histograms. Descriptors can also be obtained using machine learning approaches. For example, SegMap creates descriptors from clustered objects to do the association with a map that contains the same type of descriptor [24].

The authors are with the Université de Technologie de Compiègne, CNRS, Heudiasyc, France. This work has been co-financed by the Hauts-de-France Region and Labex MS2T. It has been carried out within SIVALab, a joint laboratory between Renault and Heudiasyc (UTC/CNRS) and is part of the ANR ToICar project (ANR-21-CE33-0012).

In this paper, a fully decentralized CL with map update based on vehicle-to-building measurements is presented. The collaboration is done when at least two vehicles observe the same building facades at the same time using their on-board 3D lidars. At this moment, they exchange vehicle poses, covariances, cross-covariance, maps, and their lidar observations which are designed to be compact. The communicating vehicles can then update their states with the SKF. The poses of the non-communicating vehicles are not updated, but their effect on the covariance is determined. Likewise, each vehicle has its own map estimate (the poses of building facades) which is updated at the same time. To maintain consistency, a Kullback Leibler Average (KLA) [25] is used to fuse the state of collaborating vehicles before applying the SKF. Features in the form of facade poses are generated from lidar point clouds to be used directly in the SKF.

The contributions of this paper are: (i) the presentation of a novel method for decentralized CL with map update using indirect measurements based on SKF, (ii) a landmark extraction method in the form of poses from building lidar point clouds, (iii) the experimental evaluation with real-world data recorded with three vehicles.

The paper is organized as follows. Section II presents the SKF used for CL and map update. Section III details the observation generations from lidar point clouds. Section IV reports experimental results carried out with three experimental vehicles. An analysis of the performance achieved through indirect collaboration is provided.

II. Decentralized Collaborative Localization based on Schmidt Kalman Filter

A. Problem Statement

Let's consider the localization of vehicles in urban environments with buildings referenced in an uncertain prior map (e.g. OSM). Initially, this map is the same for all vehicles where a unique identifier (ID) is associated with each building facade and can be shared between vehicles. The localization is done with respect to a local ENU frame (East, North, Up) denoted M in which the coordinates of the initial map elements are transferred. The map is uncertain in the sense that the buildings are mapped with some errors. The goal is to improve the localization accuracy and consistency using collaboration between vehicles while updating the map. Indeed, each vehicle indirectly shares its localization information with the others via the map, which is refined simultaneously.

Vehicles can collaborate when they observe the same building facades (*indirect measurements*) at the same time. To accomplish this, each vehicle uses wireless communication to locally broadcast the IDs of the observed facades. These messages are then very short and consume a minimum of bandwidth. Other vehicles in the vicinity check whether the received IDs correspond to one of their own observations. If there is at least one in common, they engage a peer-to-peer communication with the sender, exchanging all the necessary information required for a decentralized estimation

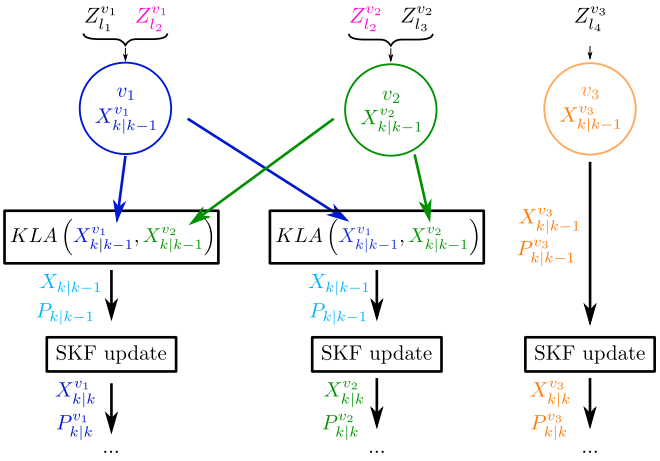

Fig. 1: General architecture of the method. When two vehicles observe a common landmark (l_2 in pink), they exchange their estimated states and fuse them with the KLA.

based on SKF. This includes sharing their maps, observations, poses and covariance matrices. The same procedure is carried out by the other vehicles. As communication delays are unavoidable, all observations are time-stamped with a common date and each vehicle maintains a buffer to handle out-of-sequence data to apply SKF updates at the right time.

The use of buildings for CL increases the range of collaboration. For example, even if the vehicles do not see each other directly, they can help each other in their localization tasks.

Consider a system of several vehicles (denoted V) and landmarks (denoted L). The joint state vector corresponds to the poses of the vehicles and the landmarks:

$$X = \begin{bmatrix} (X_v)_{v \in V} \\ (X_l)_{l \in L} \end{bmatrix} \tag{1}$$

Both vehicles and landmarks states are represented as 2D poses $X_v = [x, y, \theta]_v^T$ and $X_l = [x, y, \theta]_l^T$, with x and y the position, and θ the orientation.

B. Decentralized Collaborative Estimation

Consider a vehicle v_i and a subset of vehicles V_i observing at least one landmark in common with $v_i \in V_i$. The decentralized update of the system state using an SKF needs a rearrangement of the state vector in two parts: an estimated part s that includes the vehicles poses involved in the collaboration and the landmarks poses, and a parameter part p that contains the poses of the vehicles that do not collaborate at this moment with v_i. The p part is not updated, but its effect on the covariance is determined. A choice was made to consider all landmarks in the s part in this paper, for simplification.

The state vector is then:

$$X = \begin{bmatrix} X_s \\ X_p \end{bmatrix}, \tag{2}$$

where

$$X_s = \begin{bmatrix} (X_v)_{v \in V_i} \\ (X_l)_{l \in L} \end{bmatrix}, \tag{3}$$

and

$$X_p = (X_v)_{v \notin V_i} = (X_v)_{v \in \bar{V}_i}. \quad (4)$$

The covariance matrix is of the form:

$$P = \begin{bmatrix} P_{ss} & P_{sp} \\ P_{ps} & P_{pp} \end{bmatrix}. \quad (5)$$

At instant k, each vehicle has its own filter and can make the prediction step independently of the others using odometry data.

The predicted cross-covariance matrix between vehicles i and j is

$$P_{ij,k+1|k} = F_{i,k} P_{ij,k|k} F_{j,k}^T, \quad (6)$$

where F_i (respectively F_j) is the Jacobian matrix associated to the state transition of vehicle i (respectively j). In the absence of communication with vehicle j (i.e., j in p), its evolution model is supposed to be static and $F_j = I$.

Regarding the landmarks, the evolution model is static. The prediction of cross-covariances between a vehicle i and a landmark j is performed similarly to (6), with $F_{j,k} = I$.

The update step uses the observation $Z_{l_j}^{v_i}$ of landmark l_j done by vehicle v_i, in the vehicle frame:

$$Z_{l_j}^{v_i} = \begin{pmatrix} \cos\theta_{v_i} & \sin\theta_{v_i} & 0 \\ -\sin\theta_{v_i} & \cos\theta_{v_i} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{l_j} - x_{v_i} \\ y_{l_j} - y_{v_i} \\ \theta_{l_j} - \theta_{v_i} \end{pmatrix} + \beta \quad (7)$$

where β is the observation noise considered as white Gaussian with covariance matrix $R_{l_j}^{v_i}$. The construction of this observation from the lidar data is detailed in section III. Considering the restricted communications and the decentralized architecture, the observation vector of vehicle v_i consists of its own observations and those of the landmarks observed by other vehicles that are common with v_i.

The Jacobian matrix associated to the observation model can be written as:

$$H = \begin{bmatrix} H_s & H_p \end{bmatrix}. \quad (8)$$

Since the used observations are between a vehicle in s and a landmark in s, H_p is equal to zero. For more details, please see [26].

The optimal Kalman gain for the s part in then expressed in the form:

$$K_{s,k} = P_{ss,k|k-1} H_{s,k}^T \left[H_{s,k} P_{ss,k|k-1} H_{s,k}^T + R_k \right]^{-1}, \quad (9)$$

where R_k is the covariance matrix of the observation errors. With the SKF, the p part is not updated, which can be achieved by choosing the associated Kalman gain $K_p = 0$.

Therefore, the update step of the SKF is given simply by:

$$X_{p,k|k} = X_{p,k|k-1}, \quad (10)$$

$$P_{pp,k|k} = P_{pp,k|k-1}, \quad (11)$$

$$X_{s,k|k} = X_{s,k|k-1} + K_{s,k}(Z_k - h(X_{s,k|k-1})), \quad (12)$$

$$P_{ss,k|k} = (I - K_{s,k} H_{s,k}) P_{ss,k|k-1}, \quad (13)$$

with Z_k the observation vector and h the observation model given in (7).

The cross-covariance between collaborating and non collaborating vehicles is computed as follows:

$$P_{sp,k|k} = (I - K_{s,k} H_{s,k}) P_{sp,k|k-1}. \quad (14)$$

C. Fusion with Kullback-Leibler Average

Decentralization with communication constraints leads to multiple map estimates and several versions of vehicle-map cross-covariance. Indeed, each vehicle estimates its own map and collaboratively refines it using observations from other vehicles during update steps. This leads to differences in the map estimates as well as in the cross-covariance with the vehicles.

In the following, we denote $X^{v_i} = \begin{bmatrix} X_V^{v_i} \\ X_L^{v_i} \end{bmatrix}$ the estimation of the full state (including all vehicles and landmarks) by vehicle v_i, and P^{v_i} the associated covariance matrix.

Before the update step of the SKF, an unweighted Kullback-Leibler Average (KLA) [25] is used to fuse the full states of collaborating vehicles:

$$\bar{P}_{k|k-1}^{-1} = \frac{1}{m} \sum_{j=1}^{m} (P_{k|k-1}^{v_j})^{-1}, \quad (15)$$

$$\bar{X}_{k|k-1} = \bar{P}_{k|k-1} \frac{1}{m} \sum_{j=1}^{m} (P_{k|k-1}^{v_j})^{-1} X_{k|k-1}^{v_j}, \quad (16)$$

where m in the number of collaborating vehicles. The KLA is equivalent to the use of a covariance intersection filter, resulting in a conservative fusion. After this step, the SKF update step is applied (equations 9 to 14) using $\bar{X}_{k|k-1}$ and $\bar{P}_{k|k-1}$ as prediction.

The KLA guarantees consistent data fusion, provided that the estimates to be merged are consistent. The part of the state corresponding to vehicles in p is not predicted, and their estimates were only updated during the last collaboration, which may be very old. To keep a consistent estimate of the full state, the uncertainty of the prediction of part p is increased as follows:

$$P_{pp,k|k-1} = P_{pp,k-1|k-1} + Q_p \quad (17)$$

where Q_p is the covariance representing the uncertainty fixed according to the possible speed of the vehicles.

The general architecture of the proposed approach is shown in figure 1.

III. Facade observations

The measurements used in the SKF are generated from different lidar point clouds when observing facades. This section describes the process to get the lidar observations.

A. Reference Frames

A facade is defined as a plane delimited by two 2D points p_1 and p_2 located at ground level, and of sufficient height h to characterize the entire surface with a Normal Distribution (ND) grid (figure 2). By definition, a facade plane is plastered to the exterior side of the building wall. The facade landmark is defined by a normal vector located in the middle of the facade, at ground level (figure 2). Each facade landmark has a

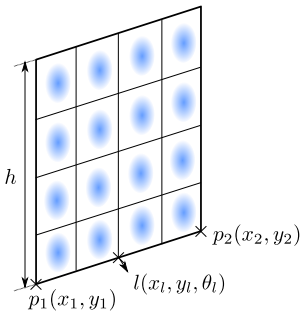

Fig. 2: Facade plane representation. The ND grid, represented as 3D blue ellipses, is fitted on the plane. A landmark is represented with 2D pose l located at the middle of $[p_1\ p_2]$ with the normal vector.

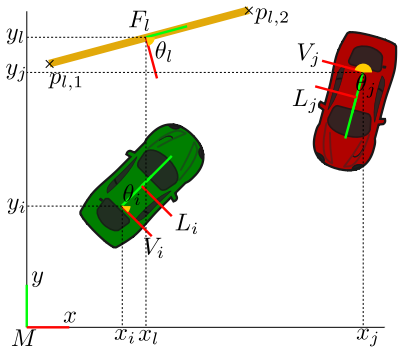

Fig. 3: Reference frames: L for lidar, V for vehicle, M for map and F_l for facade l.

unique identifier, which is shared among all vehicles through a common initial map. Since the CL is done in 2D, the facade landmark is represented by a 2D pose l located at the middle of the segment $[p_1\ p_2]$ with associated frame F_l (figure 3). These poses constitute the landmark map with state X_l (section II). For every vehicle, we also need to manage a lidar reference frame L_i and a vehicle frame V_i, which is the one to be localized.

B. Overview of the Processing Stages

In order to get the set of observations from a lidar point cloud according to our representation of the map, five steps are needed, illustrated in figure 4:

1) **Classification**: The lidar point cloud is classified in 3D to keep only building points, using Cylinder3D, a 3D convolution network for lidar segmentation [27]. This step reduces the number of points to be processed and keeps the relevant ones.
2) **Registration**: The building point cloud is aligned with the prior map using the Normal Distributions Transform (NDT) [28]. The transformation, obtained from this 3D alignment with a 2D constraint, is not used for localization, but only to perform the next step.
3) **Clustering**: Using the aligned point cloud, each point is associated to the nearest facade in 2D, within a maximum distance of 1 m.

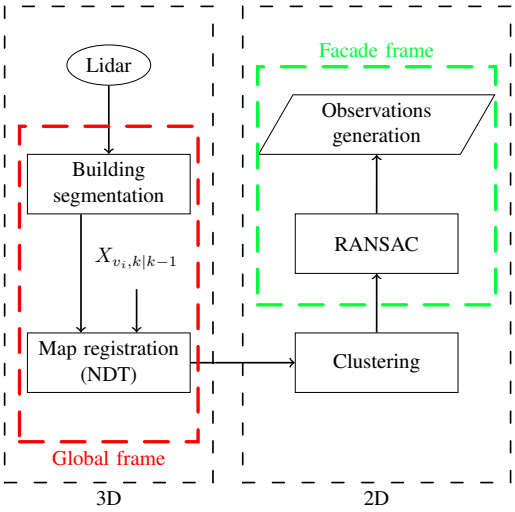

Fig. 4: Proposed architecture to generate the observations.

4) **RANSAC**: RANSAC is performed to eliminate a maximum of outliers [29].
5) **Observation generation**: An observation is extracted from each facade fully perceived to be used in the SKF.

In the following, we detail steps 2 to 5.

Registration and clustering

In order to associate each lidar point classified as a building with a facade, a global NDT registration is first performed [21]. Each point is then associated to the nearest facade, within a maximum distance. The NDT is used to describe more precisely the shapes of the facades with less storage compared to other methods like Iterative Closest Point (ICP) [30].

To do so, each facade plane is divided into a grid that contains 3D normal distributions, chosen uniformly [31] (figure 2). Then, these normal distributions are transformed from the facade frame F_l to the map frame M, to perform a global registration:

$$^{M}\mu_{il} = {}^{M}T_{F_l}.{}^{F_l}\mu_{il} \tag{18}$$

$$^{M}P_{il} = {}^{M}R_{F_l}.{}^{F_k}P_{il}.{}^{M}R_{F_l}^{T} \tag{19}$$

where $^{F_l}\mu_{il}$ and $^{F_l}P_{il}$ are the mean and covariance of the i^{th} normal distribution of the l^{th} facade, $^{M}T_{F_l}$ is the transformation matrix from the facade frame F_l to the map frame M and $^{M}R_{F_l}$ is the associated rotation matrix.

The registration consists in finding the optimal transformation between the constructed NDs from the map and the lidar point cloud by minimizing a negative log-likelihood function with a gradient descent optimization. The nearest ND to the lidar input points is obtained by searching the centroids of NDs using a kdtree. Even if a 3D point cloud is used, the final transformation is constrained to a 2D pose by setting the z translation, x and y rotations to 0 at each step, using the projected gradient descent [32].

After the registration step, each point classified as "building" in the lidar point cloud is associated with the nearest

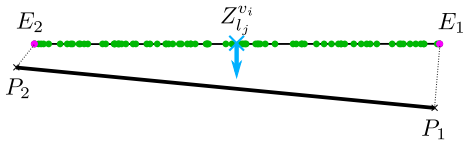

Fig. 5: Point cloud projected on the RANSAC model and observation generation. The extreme projected inliers E_1 and E_2 constitute the segment whose midpoint and normal are retrieved to form the observation $Z_{l_j}^{v_i}$.

facade by projecting the points on the segments that represent the facades. If the association distance is greater than a threshold (e.g., 1 m), the point is not considered.

Final refinement with RANSAC

A RANSAC is applied to the clustered points to keep only the points that best represent the shape of the facade, whether it is a straight line or a curve [29]. Two RANSAC models are then applied, a 2D line and a circle. The model that gathers the most inliers is selected. The points are then projected on the selected RANSAC model (figure 5). This assumes that the targeted facade is flat or circular.

Observation generation

The goal now is to get an observation in the form of a pose (at most one per facade). We consider that an observation can be built from a facade if it is fully observed by a vehicle. In practice, this is done by checking that the ratio of the measured segment length over the map facade length is between 0.95 and 1.05.

For this purpose, the extremes E_1 and E_2 are chosen to be the furthest inliers projected on the RANSAC model. Then for these points, the middle and the normal of the segment are computed (figure 5) in a comparable manner to what was done for the map, so as to obtain the same orientation.

The observation $Z_{l_j}^{v_i}$ is then obtained in the lidar frame, and is transformed into the vehicle frame to be used in the SKF.

IV. Experimental Results

A. Real data collection

In this section, the approach is evaluated using experimental data collected with three equipped vehicles. The data acquisition was performed on the campus of the Université de Technologie de Compiègne that presents a challenging urban environment, as shown in figure 6. A difficult area featuring narrow passages, metallic buildings, walkways, and stairs, disturbs the localization and the lidar observation. Each vehicle was equipped with a 32 layers Velodyne lidar VLP32-C and odometry sensor using wheel speed and gyro sensors. A SPAN CPT Inertial Measurement Unit (IMU) with GNSS with Post Processed Kinematic (PPK) was used for centimeter level accuracy ground-truth. Acquisition were made at a frequency of 10 Hz for lidar measurements and 100 Hz for dead-reckoning. GNSS was only used for initialization and is therefore not part of the localization system.

Fig. 6: The three experimental vehicles in an urban canyon. An example of a ND grid is represented on the right along with the facade plane and the normal vector in the middle l of the segment.

The vehicles were driven along planned trajectories, which are shown in figure 7. This figure illustrates the facade IDs and also highlights indirect collaborations with black connections. The initial map was retrieved from OSM with some simplifications to merge a few collinear facades. The recording and the processing of the experimental data was done using ROS (Robot Operating System) and PCL (Point Cloud Library) for point cloud manipulations. The NDT used is a modification of the multithreaded PCL version[1].

B. Scenario

The trajectories of the three vehicles are displayed in figure 7. The vehicle name is the same as the color of its trajectory. At the beginning of the experiment, vehicles Blue and Green collaborate directly by crossing each other until $t = 12$s. Orange is in standalone until $t = 8$s where it joins the two other vehicles, the three vehicles are then collaborating through the landmark 13. Blue is alone from $t = 12$s while Orange follows Green in the canyon, where they collaborate. Blue collaborates again with Green around $t = 30$s and collaborates then with both vehicles for around 20 seconds. Finally, Blue continues alone from $t = 47$s, and Orange follows Green until $t = 65$s, where their paths separate. They collaborate through the observations of the landmarks 2 and 36 (among others). Orange and Green have some occasions to collaborate with Blue through the observation of the landmark 9 around $t = 60$s, which shows that long-range collaboration without line of sight is possible.

C. Results

The presented results aim to evaluate the contribution of the collaboration, by comparing with the standalone case. The comparison with a centralized filter is not studied in this paper, but can be found in [26] in a simulation environment.

As there are unavoidable erroneous measurements, the SKF has been robustified through a Fault Detection and Exclusion (FDE) stage using residuals based on Mahalanobis

[1] https://github.com/koide3/ndt_omp

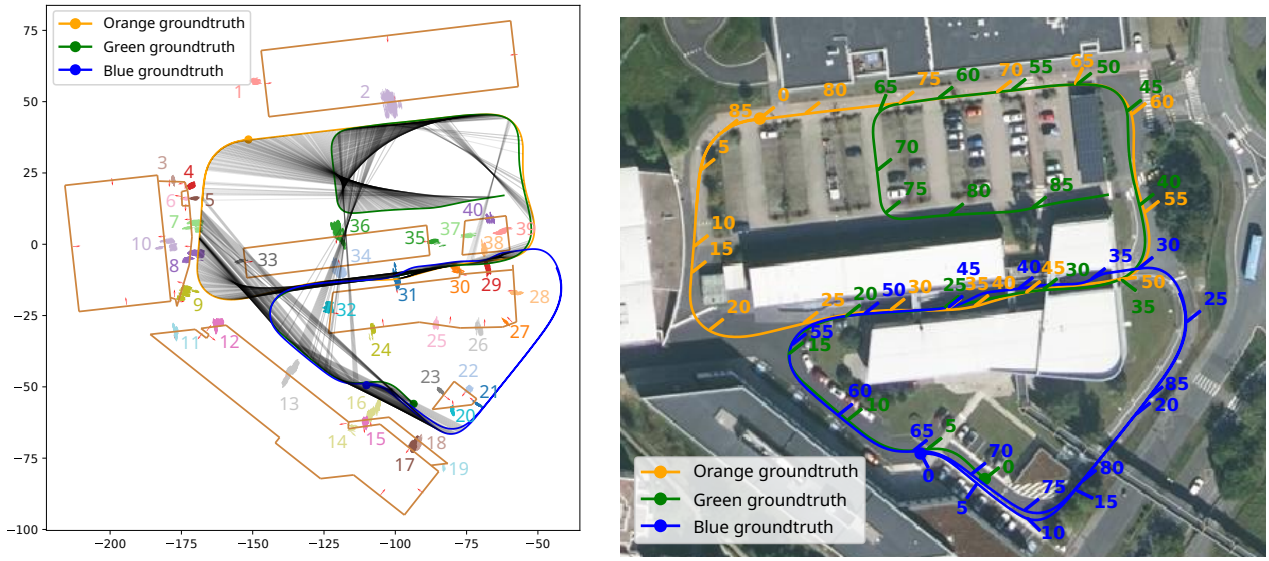

Fig. 7: Overview of the experiment. The beginning of each track is represented by a filled circle. When two vehicles collaborate, their locations are connected by a black line on the left plot. The arrows represent the observations of the vehicles, converted in the map frame. The IDs of the facades are also shown. On the right figure, the numbers represent the timestamps of several positions, in seconds.

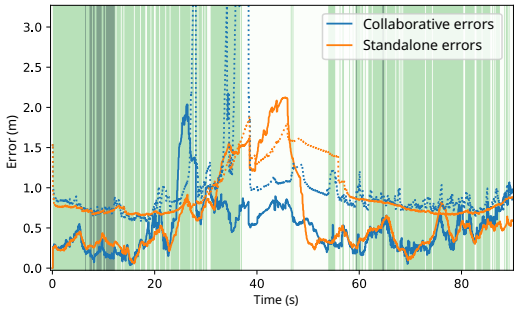

Fig. 8: Euclidean position errors of Green (in solid lines), with associated uncertainty bound at 99.97% (Rayleigh distribution) (in dots). The green areas highlight the number of vehicles involved in the collaboration (light green for two vehicles and dark green for three).

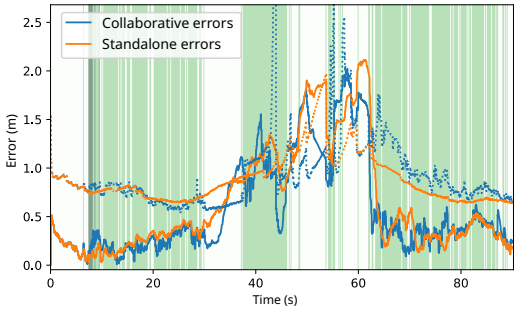

Fig. 9: Euclidean position errors of Orange (in solid lines), with associated uncertainty (in dots).

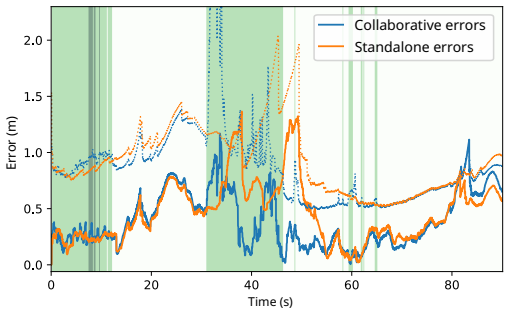

Fig. 10: Euclidean position errors of Blue (in solid lines), with associated uncertainty (in dots).

distances between the observations and their predictions. The faulty observations are then excluded from the data fusion.

Localization

Figures 8, 9 and 10 show the Euclidean position errors with the corresponding uncertainty region for the three vehicles. Results for both collaborative and standalone approaches are presented, incorporating FDE. For Green, an increase in error in collaborative mode can be noticed between $t = 25$s and $t = 30$s, which coincides with the start of the complex area where the facades are difficult to detect. The collaborative errors are higher than standalone ones, which can be explained by undetected erroneous measurements. Then the error decreases rapidly after collaborating with Blue around $t = 30$s and remains small until the end of the collaboration at $t = 38$s. This can be explained by the augmentation of the number of observations thanks to the collaboration. In standalone mode, the vehicle experiences a more significant increase in error within the difficult area

Vehicle	Mean absolute error (cm)		Consistency (%)	
	Collaborative	Standalone	Collaborative	Standalone
Blue	**39.4**	46.5	**99.1**	97.7
Green	**50.9**	62.6	**95.4**	85.3
Orange	**57.9**	64.6	**82.0**	77.7
All	**49.4**	57.9	**92.2**	86.9

TABLE I: Errors and consistency (99.7% uncertainty bound).

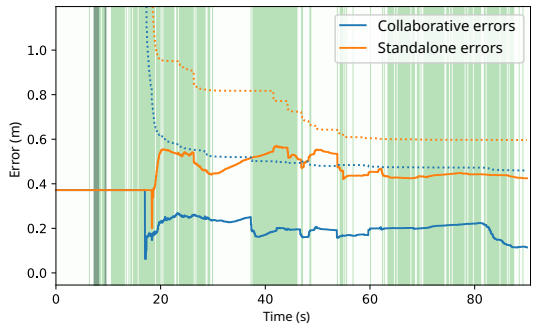

Fig. 11: Position error of landmark 31 estimated by Orange.

as fewer observations are available. Indeed, within this area, only one or two observations are made by the vehicle. The collaboration adds one or two observations, which represents a large increase in available information.

Regarding Blue, the standalone method encounters issues around $t = 40$s while traversing the difficult area. Green and Blue assist each other through collaboration. Indeed, these two vehicles approach from opposite directions. Collaboration allows each vehicle to obtain an improved version of the map. Starting from $t = 8$s, Orange collaborates with Green and follows it from $t = 20$s. The error of Orange increases between 40 and 60 seconds, which corresponds to the difficult area. As observed with other vehicles, the standalone method appears to have more difficulty correcting errors in this area.

Table I shows the mean Euclidean position error for all the vehicles. Collaboration enhances accuracy by approximately 15% compared to standalone mode.

Table I also presents the consistency obtained with the proportions of errors exceeding the 3σ region along the x and y axes (equivalent to a proportion of 99.7% for a normal distribution). It can be seen that collaboration improves the consistency of localization compared to the standalone method. Specifically, 92.2% of the errors fall within the uncertainty region for the collaborative case, whereas this value is only 86.9% for the standalone case. This is mainly due to the increased number of observations used through collaboration, enabling a better state estimation performance.

Map Update

A ground truth of the map carried out by surveyors is available. As the representations of the facade lengths are not always the same between the ground truth and OSM, errors are only calculated along the facade normal.

The overall map average error remains in the same order for collaborative and standalone methods, with 36.0 cm in collaborative versus 36.7 cm in standalone. This similarity can be attributed to an initial map with good relative positioning of landmarks. However, amelioration can be seen on distinct landmarks.

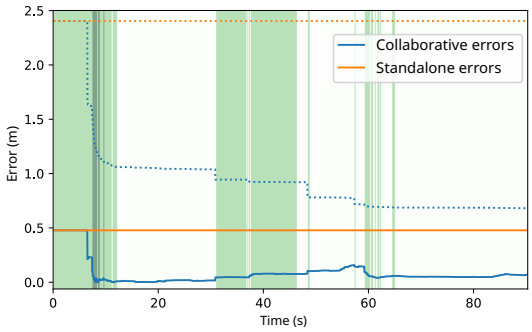

Fig. 12: Position error of landmark 2 estimated by Blue.

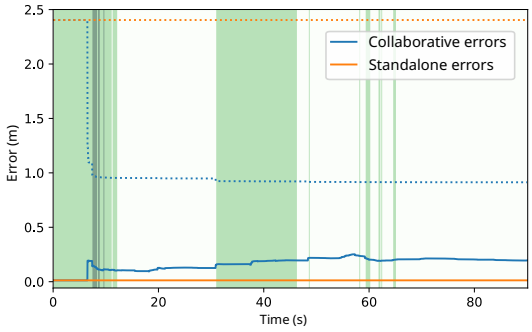

Fig. 13: Position error of landmark 6 estimated by Blue.

Figure 11 illustrates the position error of landmark 31 as estimated by Orange. The benefits of the collaboration are clear in this case. Indeed, Orange is not the first vehicle to observe this landmark; it collaborates with Green, which observed it previously. As a result, Orange achieves a more accurate estimation through collaboration.

Figure 12 shows the estimation of landmark 2 by Blue. In standalone mode, the initial error of this landmark remains unchanged, since landmark 2 is not observed by Blue. However, through collaboration with Orange, Blue improves the estimation of this landmark without directly observing it. Better accuracy with less pessimistic uncertainty region can be observed. This improvement is beneficial for Blue in the future when it does observe this landmark, as it already has a more accurate estimation with reduced uncertainty.

Figure 13 shows the example of an almost perfect landmark in the initial map (landmark 6), which is not directly observed by Blue. In this case, collaboration results in a slight increase in error compared to the initial value. In the standalone case where this landmark is not updated, the initial error in maintained. However, collaboration significantly reduces the uncertainty while maintaining consistency, making the landmark more valuable for localization.

Experimental results show that localization based on the perception of buildings by a lidar with OSM priors can achieve an average accuracy of 50 cm. This is an interesting

result, obtained without any absolute localization system like GNSS (except for initialization). The results also show that collaboration between vehicles improves both accuracy and consistency of localization and map estimation. Collaboration via common landmarks enables long-range collaborations (over 100 m) even in situations without a direct line of sight between vehicles.

V. Conclusion

The fully decentralized CL with map update presented in this paper uses vehicle-to-building observations. The Schmidt-Kalman filter was used to estimate the poses of communicating vehicles and to simultaneously update the map in a consistent manner with KLA. A pipeline for feature extraction from lidar building point clouds was proposed.

The proposed approach to estimate vehicle poses with map update, in a decentralized way, demonstrates improved localization performance in terms of accuracy and consistency. This improvement is mainly due to the enhancement of the map during navigation and the increased number of observations. Collaboration occurs when at least two vehicles observe the same building facade at the same time. Then, vehicles can collaborate even when they do not see each other, which increases the range of collaboration as shown by the real data experiments carried out with three vehicles.

In future work, we intend to improve the feature extraction method and to enable real-time implementation by avoiding the exchange of entire maps.

References

[1] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, "GNSS Position Integrity in Urban Environments: A Review of Literature," *IEEE Trans. on Intelligent Transportation Systems*, vol. 19, no. 9, 2018.

[2] S. Schön, C. Brenner, H. Alkhatib, M. Coenen, H. Dbouk, N. Garcia-Fernandez, C. Fischer, C. Heipke, K. Lohmann, I. Neumann, U. Nguyen, J.-A. Paffenholz, T. Peters, F. Rottensteiner, J. Schachtschneider, M. Sester, L. Sun, S. Vogel, R. Voges, and B. Wagner, "Integrity and Collaboration in Dynamic Sensor Networks," *Sensors*, vol. 18, no. 7, 2018.

[3] R. Ou, G. Liang, and T. L. Lam, "FPECMV: Learning-Based Fault-Tolerant Collaborative Localization Under Limited Connectivity," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 11 095–11 102.

[4] Z. Zhu, K. Zhu, Z. Zheng, S. Chen, and N. Zheng, "Multi-L: A Novel Multi-Robot Cooperative Localization Method in Indoor Environment," in *IEEE International Conference on Intelligent Transportation Systems*, 2022, pp. 2436–2443.

[5] J. Al Hage, M. E. El Najjar, and D. Pomorski, "Multi-sensor fusion approach with fault detection and exclusion based on the Kullback–Leibler Divergence: Application on collaborative multi-robot system," *Information Fusion*, vol. 37, 2017.

[6] S. H. Vemprala and S. Saripalli, "Collaborative Localization for Micro Aerial Vehicles," *IEEE Access*, vol. 9, 2021.

[7] P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, "Towards Collaborative Simultaneous Localization and Mapping: A Survey of the Current Research Landscape," *Field Robotics*, vol. 2, no. 1, pp. 971–1000, Mar. 2022.

[8] M. Ouyang, X. Shi, Y. Wang, Y. Tian, Y. Shen, D. Wang, P. Wang, and Z. Cao, "A Collaborative Visual SLAM Framework for Service Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8679–8685.

[9] R. Dubois, A. Eudes, and V. Frémont, "On Data Sharing Strategy for Decentralized Collaborative Visual-Inertial Simultaneous Localization And Mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 2123–2130.

[10] M. T. Lázaro, L. M. Paz, P. Piniés, and J. A. Castellanos, "Distributed Localization and Submapping for Robot Formations using a prior map," *IFAC Proceedings Volumes*, vol. 46, no. 10, pp. 138–145, Jun. 2013.

[11] M. Frosi, V. Gobbi, and M. Matteucci, "OSM-SLAM: Aiding SLAM with OpenStreetMaps priors," *Frontiers in Robotics and AI*, vol. 10, Mar. 2023.

[12] E. Nettleton, S. Thrun, H. Durrant-Whyte, and S. Sukkarieh, "Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles," in *Field and Service Robotics: Recent Advances in Reserch and Applications*, ser. Springer Tracts in Advanced Robotics, 2006, pp. 179–188.

[13] A. Makarenko, A. Brooks, T. Kaupp, H. Durrant-Whyte, and F. Dellaert, "Decentralised data fusion: A graphical model approach," in *International Conference on Information Fusion*, 2009, pp. 545–554.

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

[15] S. Fang, H. Li, and M. Yang, "Multi-Vehicle Cooperative SLAM Using Iterated Split Covariance Intersection Filter," in *IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 947–952.

[16] S. F. Schmidt, "Application of State-Space Methods to Navigation Problems," in *Advances in Control Systems*, 1966, vol. 3.

[17] R. Zanetti and C. D'Souza, "Recursive Implementations of the Schmidt-Kalman 'Consider' Filter," *The Journal of the Astronautical Sciences*, vol. 60, no. 3-4, 2013.

[18] L. Luft, T. Schubert, S. Roumeliotis, and W. Burgard, "Recursive Decentralized Collaborative Localization for Sparsely Communicating Robots," *Robotics: science and systems*, 2016.

[19] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems X*, 2014.

[20] H. Guo, J. Zhu, and Y. Chen, "E-LOAM: LiDAR Odometry and Mapping with Expanded Local Structural Information," *IEEE Transactions on Intelligent Vehicles*, 2022.

[21] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, "Beyond points: Evaluating recent 3D scan-matching algorithms," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 3631–3637.

[22] Y. Cho, G. Kim, S. Lee, and J.-H. Ryu, "OpenStreetMap-Based LiDAR Global Localization in Urban Environment Without a Prior LiDAR Map," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4999–5006, 2022.

[23] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *IEEE International Conference on Robotics and Automation*. Singapore, Singapore: IEEE, May 2017, pp. 5266–5272.

[24] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3D Segment Mapping using Data-Driven Descriptors," *Robotics: Science and Systems XIV*, 2018.

[25] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, 2014.

[26] M. Escourrou, J. Al Hage, and P. Bonnifait, "Decentralized Collaborative Localization with Map Update using Schmidt-Kalman Filter," in *International Conference on Information Fusion*, 2022, p. 8.

[27] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9934–9943.

[28] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro University, 2009.

[29] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum*, vol. 26, no. 2, 2007.

[30] A. Segal, D. Haehnel, and S. Thrun, "Generalized ICP," *Robotics: science and systems*, vol. 2, no. 4, 2009.

[31] E. Javanmardi, Y. Gu, M. Javanmardi, and S. Kamijo, "Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area," *IATSS Research*, vol. 43, no. 1, 2019.

[32] P. H. Calamai and J. J. Moré, "Projected gradient methods for linearly constrained problems," *Mathematical Programming*, vol. 39, no. 1, 1987.