Consistent Decentralized Cooperative Localization for Autonomous Vehicles using LiDAR, GNSS and HD maps

Elwan Héry, Philippe Xu, Philippe Bonnifait

Consistent Decentralized Cooperative Localization for Autonomous Vehicles using LiDAR, GNSS and HD maps

Elwan Héry, Philippe Xu and Philippe Bonnifait
Université de Technologie de Compiègne
CNRS, Heudiasyc UMR 7253
CS 60 319, 60 203 Compiègne cedex, France
`elwan.hery@hds.utc.fr`

Abstract

To navigate autonomously, a vehicle must be able to localize itself with respect to its driving environment and the vehicles with which it interacts. This work presents a decentralized cooperative localization method. It is based on the exchange of Local Dynamic Maps (LDM), which are cyberphysical representations of the physical driving environment containing poses and kinematic information about nearby vehicles. An LDM acts as an abstraction layer that makes the cooperation framework sensor-agnostic, and it can even improve the localization of a sensorless communicating vehicle. With this goal in mind, this work focuses on the property of consistency in LDM estimates. Uncertainty in the estimates needs to be properly modeled, so that the estimation error can be statistically bounded for a given confidence level. To obtain a consistent system, we first introduce a decentralized fusion framework that can cope with LDMs whose errors have an unknown degree of correlation. Second, we present a consistent method for estimating the relative pose between vehicles, using a 2D LiDAR with a point-to-line metric within an iterative-closest-point approach, combined with communicated polygonal shape models. Finally, we add a bias estimator in order to reduce position errors when non-differential GNSS receivers are used, based on visual observations of features geo-referenced in a High-Definition (HD) map. Real experiments were conducted, and the consistency of our approach was demonstrated on a platooning scenario using two experimental vehicles. The full experimental dataset used in this work is publicly available.

1 Introduction

An autonomous vehicle needs to know its own pose (i.e., position and orientation in a working frame) as well as the poses of the road users in its immediate driving environment. Static information about the environment can be created from simultaneous localization and mapping (SLAM), or retrieved from a High Definition (HD) map, which is a geographic database containing relevant information such as lane borders, road signs, and speed limits. At the same time, dynamic information can be represented in a Local Dynamic Map (LDM) (ETSI, 2011), which is defined as an overlay of the static map. The LDM contains dynamic information, e.g., the pose, longitudinal speed, or yaw rate of vehicles with which the ego vehicle may interact. An LDM is typically built using perception information from onboard sensors such cameras or LiDARs. The information provided by the LDM can be used directly for control (Cognetti et al., 2014; Rosa et al., 2015)

or path planning, independently of the sensors in the vehicle. While tremendous improvements have been achieved through the use of deep learning techniques(Capellier et al., 2019), perception information remains limited to the physical field of view of the sensors, i.e., range and occlusions. Exchanges of information between multiple vehicles via wireless communication provide a means of virtually increasing the field of view of those vehicles (Bounini et al., 2020). In this work we show that LDMs, which are usually used as a cyberphysical representation of the driving environment for decision and control, can also be used to improve localization through data fusion.

To take full advantage of wireless communication, cooperative vehicles need to be especially cautious about how the information that they share is represented, in particular in relation to the inherent estimation uncertainty of this information. In terms of localization information, the uncertainty is typically represented as a covariance matrix. For a data fusion scheme to be reliable, it is crucial that the uncertainty of the individual elements of information is not underestimated. In the case of localization, this means that the estimated covariance matrix of the estimation error should be at least as large as the true unknown covariance matrix. In practice, this property can be measured experimentally by first setting a confidence level $1 - \alpha$, or equivalently a risk α. For each estimate, we check whether the ground truth is inside the confidence domain computed at the confidence level $1 - \alpha$ from the estimated covariance matrix. If so, we say that this estimate is consistent. If the ratio of consistent estimates, known as the *coverage rate* and denoted as C, is greater than the confidence level, we say that the estimation process is consistent. For safety-critical applications such as autonomous driving, guaranteeing consistency is a key requirement, even though it may lead to over-pessimistic estimations (i.e., $C \gg 1 - \alpha$). There are many sources of inconsistency: Gaussian approximation of noise, non-linear error propagation, and biased data or correlated information are among the most common sources.

Throughout this paper we focus on the consistency property of a decentralized cooperative localization architecture. The main contribution of this work is to obtain a consistent localization system by providing consistency to all the sub-components of the architecture. Different sources of inconsistency are examined and applied to different modules. We first formalize the cooperative process as a fusion of exchanged LDMs. In this context, the correlation between exchanged data, i.e., data incest, is handled through the use of covariance intersections. We then present a cooperative relative localization using a LiDAR sensor. Having consistent perception-based estimates is challenging for machine-learning-based approaches. Performance metrics (such as average accuracy) that are commonly used for object detection and tracking often disregard the consistency aspect. In our work we adopt a geometric approach that makes use of a cooperative vehicle's ability to share information. We describe in detail all the processes that are necessary to clean up potential sources of inconsistency. Finally, we illustrate the need for a bias estimator when using low-cost GNSS receivers. This can be via measurements from a camera able to measure landmarks, e.g., lane markings, that are geo-referenced in an HD map.

The article is organized as follows. We begin by giving a state of the art for cooperative localization problems in Section 2, then give the problem statement regarding LDM estimation for cooperative localization in Section 3. In Section 4 we introduce the asynchronous data fusion with the extrapolation of the LDM, the extended Kalman filter (EKF) and the covariance intersection filter (CIF) updates. In Section 5 we present a consistent relative localization based on LiDAR observation and point-to-model minimization. Finally, we present our experimental results in Section 6, and our conclusion in Section 7.

2   Related work

Improving localization via multiple cooperating agents can be done in many ways. In the context of SLAM-based methodologies, distributed approaches have been proposed (Cunningham et al., 2010; Cunningham et al., 2013) to distribute the mapping among multiple robots with smoothing and optimization approaches, and these have shown interesting characteristics in relation to filtering. SLAM-based approaches require a homogeneity in terms of perception capabilities from the cooperating agent, since the static features in

the map need to be represented in the same way. Another cooperative localization approach, introduced by (Kurazume et al., 1994), consists in moving two groups of robots one after the other, with the static group acting as the landmarks for the moving group. This use of the other robots as landmarks when they communicate their positions or their poses has become a generalized method, with moving landmark robots and the need for uncertainty to be taken into account for these robots (Martinelli et al., 2005). Some of this research deals with rigid structures between robots (Zelazo et al., 2014; Zelazo et al., 2015) allowing them to move in swarms. The links between the nodes formed by the positions shared by the robots are determined by measuring the ranges between these robots or the bearings (Martinelli et al., 2005). In order to measure these ranges or bearings (Montesano et al., 2005; Kim et al., 2020), different exteroceptive sensors can be used: cameras (Challita et al., 2009; Khammari et al., 2005), radars, or LiDARs. Using dynamic objects instead of static features, in association with a higher level of abstraction such as an LDM, enables the use of a more diverse set of perception sensors. The quantity of information shared is also less than when sharing local feature maps.

These methods are useful in different areas of mobile robotics. AUVs (Autonomous Underwater Vehicles) use acoustic waves to determine distances between them and then identify their positions by trilateration (Bahr et al., 2009). Acoustic waves are also used by land-based robots (Lin et al., 2005). In the case of communicating vehicles such as in VANet (Vehicular Ad-Hoc Network) (Hoang et al., 2018), inter-distances (Rohani et al., 2015) can be determined using other signals. Modems used for direct 802.11p communication can return the received signal strength indicator (RSSI) (Cruz et al., 2017; Hoang et al., 2015; Severi et al., 2018). Another low-cost solution is to use Ultra WideBand (UWB) systems (Hoang et al., 2017; Hoang et al., 2018; Severi et al., 2018). The arrival time of communicated messages can also be used to estimate an interdistance (Peker and Acarman, 2017).

A centralized algorithm makes it possible to localize all the vehicles in an optimal way. It takes into account the various observations made by the different vehicles and can manage the potential correlations of the errors of these observations.

(Biswas et al., 2006) presents an optimization method for sensor networks. This centralized solution nevertheless requires a server or a leader vehicle to do all the computations (Worrall and Nebot, 2007). All the observations also need to be communicated, which is difficult when there is a large quantity of information from LiDARs or cameras. Some computations can be assigned to vehicles individually in order to reduce the information communicated. For example, relative poses estimated from a perception sensor can be communicated instead of the raw data from the sensor. Problems may still occur if communication is interrupted between a vehicle and the system responsible for global localization. In order to avoid these problems, decentralized methods allow the information communicated in each vehicle to be taken into account, which can ensure a degree of independence for individual vehicles in relation to the rest of the fleet (Mu et al., 2011; Roumeliotis and Bekey, 2000). Even in cases where communication is limited, the localization of each agent can be estimated (Zhu and Kia, 2019; Safaei and Mahyuddin, 2019). Different decentralized cooperative localization methods have been presented in the literature (Mu et al., 2011; Gao et al., 2019; Roumeliotis and Bekey, 2002; Grime and Durrant-Whyte, 1994). The sharing of information can give rise to collaboration. Relative poses of observed robots (Howard et al., 2003) or robot states can be exchanged (Aufrère et al., 2010; Karam et al., 2006a; Karam et al., 2006b; Karam, 2009; Tischler and Hummel, 2005). Another solution is to directly share sensor measurements (Madhavan et al., 2002) such as lane borders or GNSS measurements (Rife and Xiao, 2010).

Cooperative localization means that it is important to be able to correctly propagate uncertainties (Roumeliotis and Rekleitis, 2004) in order to estimate the confidence associated with the desired states (Roumeliotis and Bekey, 2002) and to maintain a good integrity (Liu et al., 2018). It is also important to be able to detect defects and reject outliers (Al Hage et al., 2019). Interval analysis methods (Kieffer et al., 2000; Jaulin et al., 2001; Lassoued et al., 2017) such as constraint propagation maintain a good integrity (Jo and Lee, 2008) when they are combined with robust methods termed *q-relaxed* (Jaulin et al., 2001). Probabilistic methods also allow uncertainties to be handled effectively (Fox et al., 2000), and outliers can also be filtered out.

One of the main sources of error in estimating this uncertainty in a cooperative localization system comes from the communication of vehicle states. If one vehicle estimates its state from the state of a second vehicle, and vice versa, then some information is being used several times. Some data fusion methods such as the Kalman filter assume that the errors on the information to be merged are uncorrelated. If the same information is used several times, data consanguinity appears, which leads to overconfidence in the result obtained. The communication of sensor measurements avoids this problem, but significantly increases the amount of information communicated. Each vehicle can also update a dependency tree to avoid this problem (Howard et al., 2003). In the case where the errors are correlated, estimating the correlation between these errors becomes complex. The intersection of covariance (Julier and Uhlmann, 1997; Li and Nashashibi, 2012; Reinhardt et al., 2012; Carrillo-Arce et al., 2013) makes it possible to remain consistent in this type of situation. It involves finding the smallest uncertainty ellipse containing the uncertainty ellipses to be merged. However, this method of merging data often gives pessimistic results that are less accurate than those given by a Kalman filter. In order to avoid being too pessimistic and to remain accurate, the split covariance intersection filter can be used (Julier and Uhlmann, 2001; Li and Nashashibi, 2012; Li, 2012; Li and Nashashibi, 2013; Wanasinghe et al., 2014; Delobel et al., 2019). This represents a good compromise between the Kalman filter and the covariance intersection filter.

3 Problem statement

Different cooperative vehicles are likely to have heterogeneous sets of sensors and perception modalities. A common representation space, such as an LDM, is therefore necessary when merging diverse types of perception information. An LDM contains a list of dynamic objects that are described by their pose, their longitudinal speed, and yaw rate, together with a time stamp. The LDM resulting from the fusion of all communicated LDMs can be used as an abstraction layer, which gives the best absolute localization and the best relative localization of the other dynamic agents independently of the sensors.
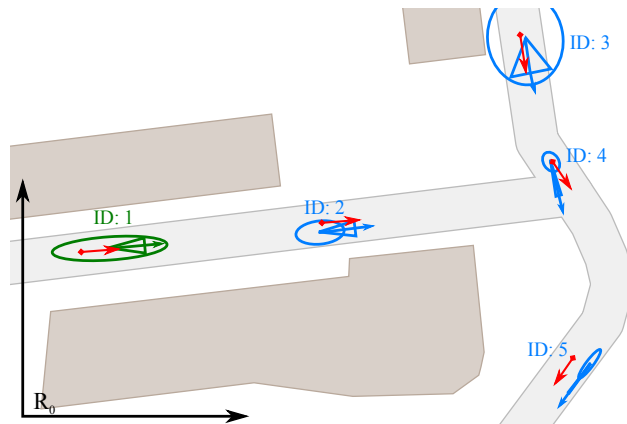
An LDM is a cyberphysical representation of the real world, as illustrated in Figure 1. In this work we assume that the road is flat (although not necessarily horizontal) and that all the vehicles are in the same plane. The LDM is therefore shown in a 2D space. Figure 1a shows a real-world scenario with multiple vehicles, where each vehicle is only able to perceive and communicate with a limited number of other agents. Figure 1b illustrates the content of the LDM. Each vehicle is mapped to a unique ID and represented with its estimated pose along with its associated uncertainty. As introduced in Section 1, we wish to obtain consistent estimates from the LDM. For a given confidence level, the uncertainty can be represented as a confidence region. In Figure 1b, the uncertainty of the position is represented by an ellipse, while the confidence interval over the heading is represented by a cone. The vehicles with ID 1, 2 and 3 are modeled by consistent estimates, since their true pose (in red) is contained within their respective confidence regions. Vehicle ID 1 has an uncertain estimate in the along-track direction, which may be acceptable for lane-keeping. As for vehicle ID 3, its whole pose is very uncertain. If it is an autonomous vehicle, it should disengage the autonomous driving mode, even though here its estimate happens by chance to be close to the true pose. The pose estimate of vehicle ID 4, on the other hand, is not consistent because of the heading, while for vehicle ID 5 it is the position estimate that is not consistent. Our goal is to avoid inconsistent estimates such as for vehicles 4 and 5.

An LDM is often geometrically defined in a frame attached to the agent itself. For each vehicle, this mobile frame is positioned at the center of the rear wheels, with the x-axis oriented toward the front of the vehicle and the y-axis pointing to the left side. In practice, each observation is given in the frame of its sensor. If the transformation between the frame of the sensor and the mobile frame is constant and well calibrated, the observations can be easily expressed directly in the mobile frame of the vehicle. In this work we make an assumption that transformations are always constant and well calibrated.

To combine multiple LDMs defined in various frames, we define a common ENU (East, North, Up) reference frame denoted R_0, as illustrated in Figure 1b. The origin of R_0 can be chosen arbitrarily, e.g., in the center
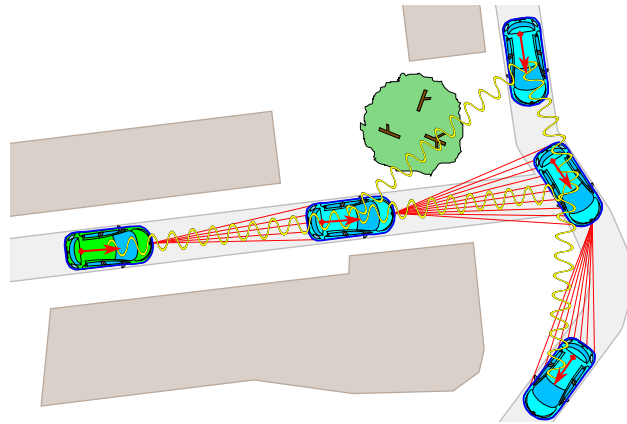
(a) *Real world: Here, vehicles are using LiDAR for perception (in red), but not all vehicles possess a LiDAR (the vehicle in the upper right-hand corner does not). Direct wireless communication is shown in yellow.*

(b) *LDM of the leftmost vehicle (in green). This is a cyberphysical representation of the real world comprising poses with uncertainty ellipses (for position) and cones (for heading). A working frame R_0 is needed. Each vehicle has an ID and its own LDM.*

Figure 1: *Real-world and cyberphysical representation via local dynamic maps.*

of the city in which the vehicles navigate. In order to avoid changing the frame back and forth we adopt a strategy where all the LDMs are always expressed in R_0 and not in the mobile frames. To simplify frame transformations, we use the compounding and inverse operators $\oplus$ and $\ominus$ introduced by Smith, Self and Cheeseman in (Smith et al., 1990) and presented in Appendix A. In the remainder of the paper the pose (position and orientation) of a vehicle i in a frame R_j will be written as ${}^j q_i = \left({}^j x_i, {}^j y_i, {}^j \theta_i \right)^T$. For simplicity, a pose ${}^0 q_i$ expressed in the ENU frame R_0 will be denoted as q_i so as to simplify notation.

4   Asynchronous Decentralized Data Fusion

4.1   LDM State Vector

In this work, all LDM agents are vehicles that can communicate and exchange information. In practice, perceived non-communicating agents can also be added to the LDM, since cooperation can still be useful for sharing the states of these agents with other communicating vehicles unable to perceive them. But since this work focuses on cooperative localization, and since non-communicating agents cannot directly improve their own localization, they are not considered. We also suppose that a unique identification (ID) is communicated by each vehicle. Linking objects in two different LDMs with different contents is therefore straightforward. There are several ways to choose these IDs, but this is not the purpose of this article. Finally, we assume that the clocks of all the agents are synchronized (e.g., through GPS time with a GNSS receiver) with negligible errors.

The estimated state of the objects

$$\widehat{X}[t] = \left(\widehat{X}_1[t], \ldots, \widehat{X}_N[t] \right)^T, \tag{1}$$

in the LDM is composed of the states of the N agents in cooperation. The number N changes over time. Below, the time component $[t]$ will be omitted when there is no ambiguity. For each agent, the state

$$\widehat{X}_i = \left(\widehat{q}_i, \widehat{u}_i \right)^T, \tag{2}$$

includes the pose $\widehat{q}_i = \left(\widehat{x}_i, \widehat{y}_i, \widehat{\theta}_i \right)^T$ and the kinetic vector $\widehat{u}_i = (\widehat{v}_i, \widehat{\omega}_i)^T$. The longitudinal speed $\widehat{v}_i$ and the yaw rate $\widehat{\omega}_i$ are used to extrapolate the pose at a given time. Therefore, each sub-state $\widehat{X}_i$ is a vector of dimension 5.

The estimation processes of the LDMs are decentralized and asynchronous: each agent runs the same algorithm and uses wireless communication to send and receive LDMs. Each vehicle has its own LDM, which represents its vision of the dynamic world around it. The sensor observations, as well as the LDMs received from nearby vehicles, improve the vehicle's LDM. In this paper we present the data fusion of the LDM of one vehicle, since each vehicle computes its LDM independently of the others. For the experimental results, the algorithms are the same on every vehicle, even though the data fusion does not require this to be the case, provided that the consistency of the computation of each agent is guaranteed.

4.2   LDM Extrapolation for Update

When a new element of information is available from an on-board sensor, the state is extrapolated at the acquisition time of this observation before this new information is merged. However, if the observation is an LDM that has been communicated by another vehicle, and which typically is received with some delay, this LDM is extrapolated at the time of the state. For each agent i, the evolution model uses its kinetic vector:

$$X_i[t|t_{-1}] = X_i[t_{-1}|t_{-1}] + V_i[t|t_{-1}], \tag{3}$$

with

$$V_i\left[t|t_{-1}\right] = \begin{pmatrix} v_i\left[t_{-1}|t_{-1}\right]\Delta t \cos\left(\theta_i\left[t_{-1}|t_{-1}\right] + \omega_i\left[t_{-1}|t_{-1}\right]\frac{\Delta t}{2}\right) \\ v_i\left[t_{-1}|t_{-1}\right]\Delta t \sin\left(\theta_i\left[t_{-1}|t_{-1}\right] + \omega_i\left[t_{-1}|t_{-1}\right]\frac{\Delta t}{2}\right) \\ \omega_i\left[t_{-1}|t_{-1}\right]\Delta t \\ 0 \\ 0 \end{pmatrix}, \tag{4}$$

where $\Delta t = t - t_{-1}$ is the time difference between the observation time t and the time t_{-1} of the last estimation of the LDM. In this work we use only a simple kinetic prediction model. A more accurate dynamic model could, however, be used, given that the acceleration and the intentions of the communicating vehicles are known.

After this stage, we obtain the evolution model of the LDM:

$$X\left[t|t_{-1}\right] = X\left[t_{-1}|t_{-1}\right] + V\left[t|t_{-1}\right], \tag{5}$$

with

$$V\left[t|t_{-1}\right] = \begin{pmatrix} V_1\left[t|t_{-1}\right] \\ \vdots \\ V_N\left[t|t_{-1}\right] \end{pmatrix}. \tag{6}$$

It can be seen from Equation 4 that the kinetic values are assumed to be constant until a new observation is available, which is not a problem for the extrapolation of the ego vehicle installation, since this is done at a high frequency. That kinetic values are constant for other vehicles is not known with such certainty, but this is taken into account as part of wider model noise.

We need to address the fact that the elapsed time Δt is not constant between two different extrapolations. Suppose that the evolution model of the kinetic values is constant with an additive error

$$u_i\left[t|t_{-1}\right] = u_i\left[t_{-1}|t_{-1}\right] + \int_{t_{-1}}^{t} \epsilon_{u_i}(\tau)d\tau \tag{7}$$

where $\epsilon_{u_i}(\tau)$ is a continuous white noise. $\forall \tau \in [t_{-1}, t]$, $\epsilon_{u_i}(\tau)$ is a white noise with a constant power $\begin{pmatrix} \nu_{v,i} & 0 \\ 0 & \nu_{\omega,i} \end{pmatrix}$, and so the covariance matrix of the global model discrete error is

$$\Sigma_{model} = \begin{pmatrix} \Sigma_{model,1} & 0_{5\times5} & 0_{5\times5} \\ 0_{5\times5} & \ddots & 0_{5\times5} \\ 0_{5\times5} & 0_{5\times5} & \Sigma_{model,N} \end{pmatrix} \quad \text{with} \quad \Sigma_{model,i} = \begin{pmatrix} 0_{3\times3} & 0 & 0 \\ 0 & \nu_{v,i} & 0 \\ 0 & 0 & \nu_{\omega,i} \end{pmatrix} \tag{8}$$

This model is multiplied by $|\Delta t|$ (the absolute value is needed in case $t < t_{-1}$, which can occur when the processing of the exteroceptive sensors is greater than one sampling period and requires a slight backtracking). The covariance matrix of the state is extrapolated with a first-order approximation:

$$\Sigma_X\left[t|t_{-1}\right] = J\Sigma_X\left[t_{-1}|t_{-1}\right]J^T + \Sigma_{model}\left|\Delta t\right| \tag{9}$$

where J is the Jacobian matrix of the evolution model (5):

$$J = \begin{pmatrix} J_1 & 0_{5\times5} & 0_{5\times5} \\ 0_{5\times5} & \ddots & 0_{5\times5} \\ 0_{5\times5} & 0_{5\times5} & J_N \end{pmatrix} \quad \text{with} \quad J_i = \begin{pmatrix} 1 & 0 & -v_i\left[t_{-1}|t_{-1}\right]\Delta t s_i & \Delta t c_i & -v_i\left[t_{-1}|t_{-1}\right]\frac{\Delta t^2}{2}s_i \\ 0 & 1 & v_i\left[t_{-1}|t_{-1}\right]\Delta t c_i & \Delta t s_i & -v_i\left[t_{-1}|t_{-1}\right]\frac{\Delta t^2}{2}c_i \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$
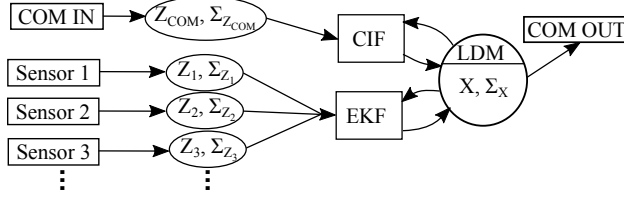
Figure 2: *LDM (X, Σ_X) data fusion with an EKF for sensor observations and their covariance matrices $(Z_1, \Sigma_{Z_1}, Z_2, \Sigma_{Z_2}...)$, and a CIF for the received LDMs (through the communication module "COM IN") and their covariance matrices $(Z_{COM}, \Sigma_{Z_{COM}})$.*

where $c_i = \cos\left(\theta_i\,[t_{-1}|t_{-1}] + \omega_i\,[t_{-1}|t_{-1}]\,\frac{\Delta t}{2}\right)$ and $s_i = \sin\left(\theta_i\,[t_{-1}|t_{-1}] + \omega_i\,[t_{-1}|t_{-1}]\,\frac{\Delta t}{2}\right)$.

It should be noted that since the kinetic information for the ego vehicle is updated at 100 Hz, the variance of the kinetic error is small for this particular object. For the other objects in the LDM, this information is updated through communication (around 10 Hz). The variance for propagating the uncertainty on the speeds of other vehicles has to be larger than for the ego vehicle. This uncertainty must be large enough for the prediction to remain consistent, in order that bad predictions from other vehicles do not impair the LDM estimation. Since this model assumes a constant motion during the period Δt, this time difference needs to remain sufficiently small.

4.3 LDM Update

We use either of two data fusion mechanisms, depending on whether the observations come from the on-board sensors or from another LDM received by communication.

Figure 2 illustrates the data fusion algorithm. The sensor observations are merged using an Extended Kalman filter (EKF), and the communicated LDMs using a covariance intersection filter (CIF). These two algorithms improve the ego LDM state and its covariance matrix.

4.3.1 Kalman update for on-board sensors

When a new sensor observation is available, it is used in an extended Kalman filter with its observation model obs (). We thus make the usual assumption that the measurement noise is conditionally independent.

From the observation $Z\,[t]$ and the extrapolated state $X\,[t|t_{-1}]$ and their covariance matrices Σ_Z and Σ_X, we first compute the innovation:

$$Y\,[t] = Z\,[t] - \mathbf{obs}\,(X\,[t|t_{-1}]).\tag{11}$$

The covariance matrix of this innovation can be estimated with the observation matrix, or with the Jacobian matrix H of the observation model if it is not linear:

$$\Sigma_Y\,[t] = H\Sigma_X\,[t|t_{-1}]\,H^T + \Sigma_Z\,[t].\tag{12}$$

The Kalman gain is computed as:

$$K = \Sigma_X\,[t|t_{-1}]\,H^T\Sigma_Y^{-1}\,[t].\tag{13}$$

We use the Joseph form (for a better numerical conditioning) of the Kalman filter to estimate the new state and its covariance matrix:

$$X\,[t|t] = X\,[t|t_{-1}] + KY\,[t],\tag{14}$$

$$\Sigma_X[t|t] = (I - KH)\Sigma_X[t|t_{-1}](I - KH)^T + K\Sigma_Z[t]K^T. \tag{15}$$

4.3.2 Covariance intersection update for communicated LDM

If an LDM is sent from one vehicle to another and then communicated back to the first vehicle after a data fusion update, the errors of these two LDMs are correlated. As the correlation is unknown, the Covariance Intersection Filter (CIF) can be used (Julier and Uhlmann, 1997) to avoid data incest issues:

$$\begin{cases} \Sigma_{X,\omega}^{-1}[t|t] &= \omega\Sigma_X^{-1}[t|t_{-1}] + (1-\omega)H^T\Sigma_Z^{-1}[t]H \\ K_\omega &= (1-\omega)\Sigma_X[t|t_{-1}]H^T\Sigma_Z^{-1}[t] \\ X[t|t] &= X[t|t_{-1}] + K_\omega(Z[t] - HX[t|t_{-1}]) \end{cases} \tag{16}$$

The weight $\omega \in [0,1]$ is chosen such that the determinant (or the trace) of the covariance matrix $\Sigma_{X,\omega}[t|t]$ is minimal:

$$\widehat{\omega} = \arg\min_{\omega} \det(\Sigma_{X,\omega}[t|t]), \tag{17}$$

where $\det()$ is the determinant. In our experiments the trace gave similar results. The Python Scipy library was used for this minimization with Brent's method (Brent, 1971).

The Joseph form of the CIF used in this work has better numerical stability than other classical forms. Updating the covariance matrix involves the addition of two positive definite matrices and avoids the need for several matrix inversions. This form is demonstrated in Appendix B.

The update that we apply is similar to the Kalman filter update. The innovation and its covariance matrix are first computed:

$$Y[t] = Z[t] - \mathbf{obs}(X[t|t_{-1}]), \tag{18}$$

$$\Sigma_{Y,\omega}[t] = H\left(\frac{1}{\omega}\Sigma_X[t|t_{-1}]\right)H^T + \frac{1}{1-\omega}\Sigma_Z[t]. \tag{19}$$

We simply add the weights $1/\omega$ and $1/1-\omega$ to the state covariance matrix $\Sigma_X[t|t_{-1}]$ and the observation $\Sigma_Z[t]$, respectively. To avoid division by zero, $\omega \neq 0$ and $\omega \neq 1$ for these equations, with the cases $\omega = 0$ and $\omega = 1$ being handled separately. When using the covariance intersection, the case where $\omega = 0$ or $\omega = 1$ occurs when one covariance matrix is somehow dominated by another: for example, the covariance intersection between a covariance matrix Σ and $\lambda\Sigma$, with $\lambda > 0$, will result in $\omega = 0$ for any $\lambda > 1$ and $\omega = 1$ for any $\lambda < 1$. Note that this example is not the only case where $\omega = 0$ or $\omega = 1$ occurs. In our case, these two singularities correspond to the cases where the previous state or the observation is more accurate than the other.

$$K_\omega = \frac{1}{\omega}\Sigma_X[t|t_{-1}]H^T\Sigma_{Y,\omega}^{-1}[t], \tag{20}$$

$$\begin{aligned} \Sigma_{X,\omega}[t|t] &= (I - K_\omega H)\left(\frac{1}{\omega}\Sigma_X[t|t_{-1}]\right)(I - K_\omega H)^T + K_\omega\frac{1}{1-\omega}\Sigma_Z[t]K_\omega^T. \\ &+ K_\omega\frac{1}{1-\omega}\Sigma_Z[t]K_\omega^T. \end{aligned} \tag{21}$$

When the optimal weight $\widehat{\omega}$ is found, the new state and its covariance matrix can be estimated:

$$\begin{aligned} X[t|t] &= X[t|t_{-1}] + K_{\widehat{\omega}}Y[t], \tag{22} \\ \Sigma_X[t|t] &= \Sigma_{X,\widehat{\omega}}[t|t]. \tag{23} \end{aligned}$$

Singularities are handled as follows:

If $\widehat{\omega} = 0$:

$$
\begin{aligned}
X[t|t] &= X[t|t_{-1}] + \Sigma_X[t|t]\, H^T \Sigma_Z^{-1}[t]\,(Z[t] - HX[t|t_{-1}]), & (24) \\
\Sigma_X[t|t] &= \left(H^T \Sigma_Z^{-1}[t]H\right)^{-1}. & (25)
\end{aligned}
$$

Else if $\widehat{\omega} = 1$:

$$
\begin{aligned}
X[t|t] &= X[t|t_{-1}], & (26) \\
\Sigma_X[t|t] &= \Sigma_X[t|t_{-1}]. & (27)
\end{aligned}
$$

With regard to computation time, this algorithm is fast in this case, since the 2D assumption limits the size of the LDM state at $5N$, with N the number of agents involved in the LDM. However, the proposed formulation of the CIF involves matrix inversions inside the minimization process. If many vehicles need to be included in the LDM, this strategy could take too much time. A solution could be to use the Fast CIF approximation of $\widehat{\omega}$ (Niehsen, 2002) (which is computed analytically without iterative optimization) and to use this weight in our formulation of $X[t|t]$ and $\Sigma_X[t|t]$ afterward.

5 Consistent Relative Localization

With V2V communications, a cooperative LDM can be built by aggregating the localization information communicated by the different vehicles. Each vehicle is thus able to localize the others up to the precision level of the individual system. With low-cost mass market GNSS receivers, the expected accuracy is at best at the meter level. By using a perception sensor such as a LiDAR, a vehicle is able to provide a more accurate relative localization of other vehicles. Current technology enables centimeter-level distance measurements. By using these relative localization, the accuracy of the LDMs can be enhanced and the localization of each individual agent is consequently improved. In this section, we present a 2D LiDAR geometric approach for consistent relative localization between cooperative vehicles.

In the context of cooperative vehicles, the perception process differs from the classical vehicle detection problem inasmuch as the presence of surrounding vehicles is known prior to the perception itself. The objective of our perception module is to compute an accurate and consistent estimate of the relative poses of surrounding cooperative vehicles. To make full use of the accuracy of LiDAR measurements, a fine geometric representation of the vehicles is needed. Where a simplistic rectangular bounding box is used to represent a vehicle, the potential benefit of having centimeter-precise measurements is lost. We therefore suppose that a 2D polygonal model representing the shape of the vehicle is accessible. This model represents the shape of the perceived vehicle at the level of the bumpers, since height does not change from one vehicle to another for safety reason. This model could either be contained in the communicated data itself or retrieved from a database if the model of the vehicle is communicated. The proposed method could be generalized with a 3D LiDAR and a polyhedric model (Coenen et al., 2017). However, a 2D perception already gives an accurate relative pose with a small computation time, as the complexity is decreased from six to three. This method has some downsides when the road is not planar (e.g. with a speed bump) or when the bumpers of the vehicles are not at the same level as the LiDAR (in the case of trucks, for example).

Let q_p be the pose in R_0 of the vehicle that perceives and q_t the pose in R_0 of the target vehicle, i.e., the vehicle that is perceived. These poses are those that need to be estimated in the LDM. Let R_p be the mobile frame of the perceiving vehicle and $^p q_t$ the relative pose of the target vehicle in this frame. This relative pose corresponds to:

$$
^p q_t = \ominus q_p \oplus q_t. \tag{28}
$$
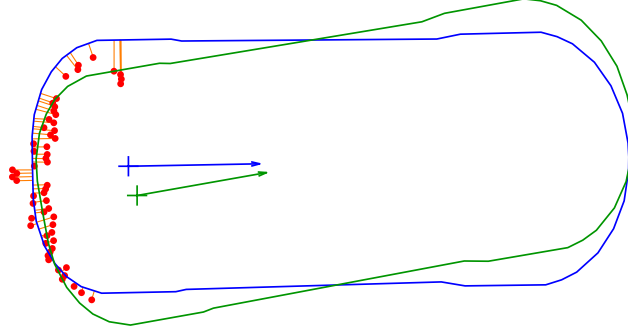
Figure 3: *Distances to minimize (in orange) between the LiDAR points (in red) and the initial polygonal model in blue. The optimal solution is found with the polygonal model in green.*
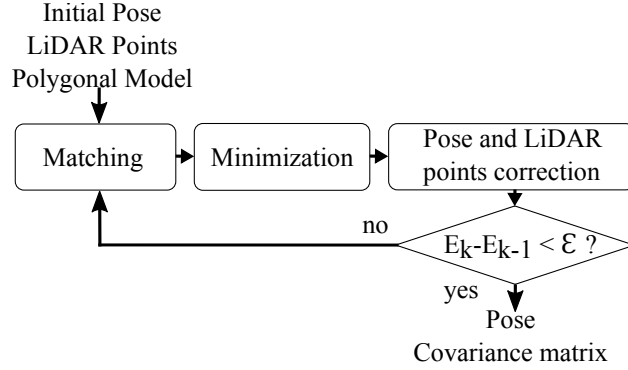
Figure 4: *Iterative matching and minimization algorithm.*

$\oplus$ and $\ominus$ are the compounding and inverse operator presented in Appendix A.

Let M be a 2D geometric model of the shape of the target vehicle and let us suppose that it is perceived in R_p as a cluster $\{p_1, p_2, \ldots, p_\gamma\}$ of 2D LiDAR points. Estimating the relative pose can therefore be formalized as finding the rigid transformation to be applied to the LiDAR points that minimizes the distances between these points and the model, as illustrated in Figure 3. This minimization problem can be formulated as follows:

$$\Delta\widehat{q} = \arg\min_{\Delta q} E\left(\Delta q\right), \tag{29}$$

where E is an error function defined as

$$E\left(\Delta q\right) = \frac{1}{\gamma}\sum_{i=1}^{\gamma} d\left(\Delta q; p_i, M\right)^2, \tag{30}$$

with $d(\Delta q; p_i, M)$ being a signed distance between the LiDAR point p_i to the model M, given a rigid transformation Δq.

This type of minimization is often encountered when matching successive LiDAR scans (Censi, 2008) and is typically solved with ICP-like methods (Iterative Closest Point). We use a similar algorithm to match the LiDAR points with the polygonal model.

5.1   Iterative Minimization of Point-to-Model Distances

Given an initial guess of the relative pose, the algorithm starts by matching each LiDAR point in the cluster with an edge in the model (Fig. 4). After this first step, a minimization algorithm is used to find

the transformation that minimizes the distances between the LiDAR points and their matched edges. This transformation is used to correct the relative position between the LiDAR points and the model. The process is repeated until convergence, i.e., the variation in the minimization error is below a convergence threshold ϵ. Implementing this iterative matching and minimization process as illustrated in Figure 4 requires defining the distance metric, minimizing the associated error function, and, importantly, modeling the uncertainty associated with the computed solution.

Distance metric To use the shortest distances during the minimization, a point-to-line matching is used. We have previously (Héry et al., 2018) compared this type of matching to other matchings. Once the closest segment j is found, we link the LiDAR point p_i to the line defined by the point m_j and the normal n_j to this segment at this point. The signed orthogonal distance between this LiDAR point and this line is defined with the compounding operator $\oplus$ presented in Appendix A:

$$d(\Delta q; p_i, M) = (\Delta q \oplus p_i - m_j) \cdot n_i \tag{31}$$

With this matching, the shortest distance to the line is always used. However, the projection is not always on the segment. When a polygonal model is sufficiently discretized, i.e., when the relative angle between successive segments is small, the error is also small.

Error minimization In Equation 31, the presence of the cosine and sine functions means that the error function to be minimized is non-convex, resulting in a complex minimization problem. To solve this issue, Low (Low, 2004) proposed a linear least-squares optimization for point-to-plan ICP. We have simplified this algorithm to obtain a 2D point-to-line version. We have previously (Héry et al., 2018) compared this minimization to an alternative minimization. The solution chosen to avoid the non-linearity of the transformation matrix ΔT is to use a first-order small angle approximation:

$$\Delta T = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & \Delta x \\ \sin(\Delta\theta) & \cos(\Delta\theta) & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -\Delta\theta & \Delta x \\ \Delta\theta & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix}. \tag{32}$$

This approximation assumes that the rotation angle is small between two iterations, and this is shown to be the case in practice. Equation 31 then becomes

$$\begin{aligned}(\Delta T p_i - m_j) \cdot n_j &= \left(\Delta T \begin{pmatrix} p_{ix} \\ p_{iy} \\ 1 \end{pmatrix} - \begin{pmatrix} m_{jx} \\ m_{jy} \\ 1 \end{pmatrix} \right) \cdot \begin{pmatrix} n_{jx} \\ n_{jy} \\ 1 \end{pmatrix}, \\ &\approx a_i \Delta q - b_i, \end{aligned} \tag{33}$$

with

$$\begin{aligned} a_i &= \begin{bmatrix} n_{jx} & n_{jy} & n_{jy}p_{ix} - n_{jx}p_{iy} \end{bmatrix}, \\ b_i &= n_{jx}m_{jx} + n_{jy}m_{jy} - n_{jx}p_{ix} - n_{jy}p_{iy}. \end{aligned}$$

The N linear equations can be written as a global linear equation:

$$A\Delta q - b, \tag{34}$$

with

$$A = \begin{bmatrix} a_1 \\ \vdots \\ a_\gamma \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} b_1 \\ \vdots \\ b_\gamma \end{bmatrix}.$$

The minimization can then be written as a linear least-squares problem

$$\Delta\widehat{q} = \min_{\Delta q} \|A\Delta q - B\|^2, \tag{35}$$

which can be solved using a pseudo-inverse matrix to solve Equation 34 equal to zero, for which the solution is given by the Moore-Penrose (Ben-israel, 2002) pseudo-inverse:

$$\Delta\widehat{q} = \mathrm{pinv}\,(A)\,b. \tag{36}$$

Uncertainty model To quantify the uncertainty of the solution computed previously, we use the residual $E(\Delta\widehat{q})$, the Hessian matrix and the degree of freedom $\kappa = 3$ to compute a covariance matrix (Bengtsson and Baerveldt, 2003; Prakhya et al., 2015; Censi, 2007):

$$\Sigma_{\widehat{q}} = 2\frac{E(\Delta\widehat{q})}{\gamma - \kappa}\left(\frac{\partial^2 E}{\partial q^2}(\Delta\widehat{q})\right)^{-1}, \tag{37}$$

with the Hessian matrix:

$$\frac{\partial^2 E}{\partial q^2}(\Delta\widehat{q}) = 2A^T A. \tag{38}$$

Intuitively, the uncertainty of the solution is large if the residual is high or if the error function is *flat* around the computed solution. Conversely, the uncertainty is small if the residual is low and if the error function is *sharp* near the computed solution.

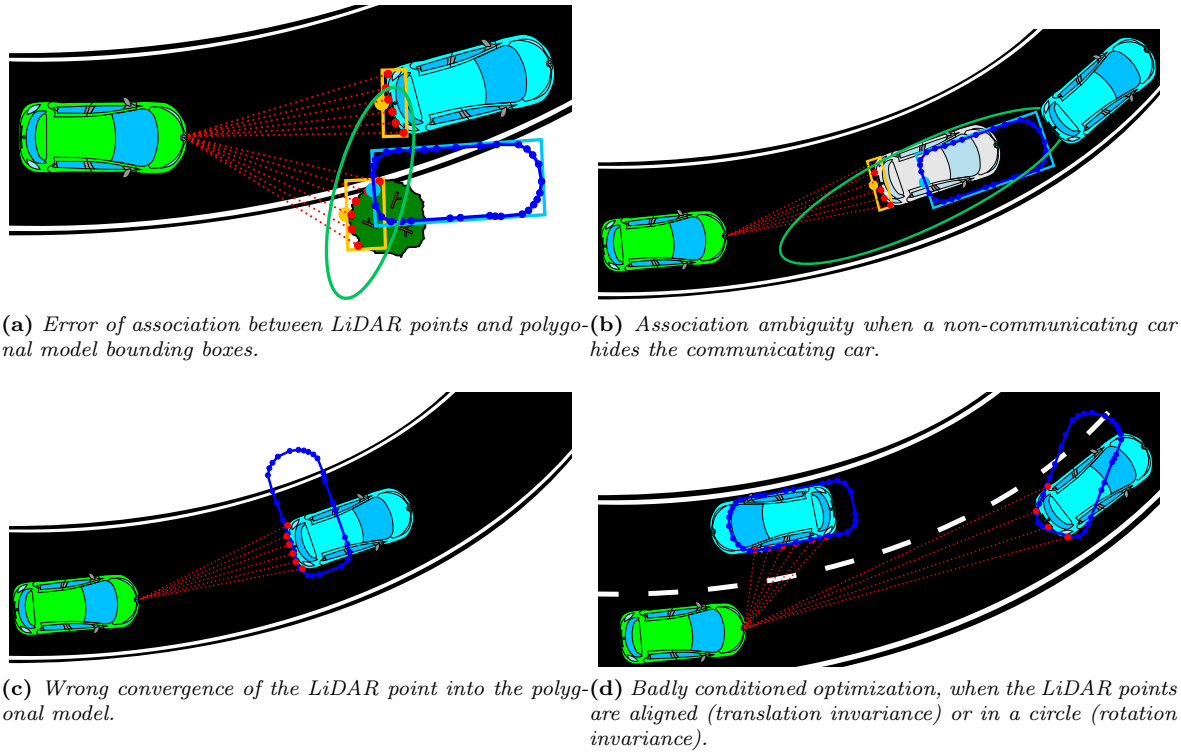## 5.2 Consistency Robustification

To have a good consistency level, four additional steps are included to make the optimization more robust. First, an initial guess of the relative pose, as close as possible to the true pose, is made, so as to avoid converging to a local minimum of the error function. Second, it is established that the cluster of LiDAR points used to match the model corresponds to the target vehicle and not to another object (Fig. 5a and 5b). Third, the convergence of the solution to a global minimum (Fig. 5c) is verified. Finally, a check is done to ensure that the minimization problem is well conditioned (Fig. 5d). These four steps are presented in this section.

5.2.1 Improved initial relative pose

An initial guess of the relative pose ${}^p q_t$ can be computed from estimates of q_p and q_t using Equation 28. However, this initial pose can be inaccurate if the estimates of q_p and q_t are also inaccurate. To improve this initialization we proceed as follows.

The LiDAR points are moved from the perceiving vehicle frame to the target vehicle frame. It is then possible to compute a bounding box with the minimum and maximum $p_{x\,i}$ and $p_{y\,i}$ coordinates, as shown in Figure 6. Since the LiDAR points are used in the frame of the target vehicle, the bounding box is oriented in the same direction as the pose of the target vehicle ${}^p q_t$. A bounding box of the model using the same method can also be computed from the minimum and maximum of the $m_{x\,j}$ and $m_{y\,j}$ coordinates of the model points.

As the vehicle is partially visible, the bounding box for the LiDAR points is smaller than the bounding box for the model. These two bounding boxes cannot therefore correspond completely, but they have the same orientation, meaning that only one point of the vehicle needs to correspond in order to correct the position error. Figure 6 illustrates this correction for a point of the vehicle (at the center of the rear) used to correct the position error of the pose ${}^p q_{t,init}$. Given that the relative orientation is approximately known,

(a) *Error of association between LiDAR points and polygonal model bounding boxes.*

(b) *Association ambiguity when a non-communicating car hides the communicating car.*

(c) *Wrong convergence of the LiDAR point into the polygonal model.*

(d) *Badly conditioned optimization, when the LiDAR points are aligned (translation invariance) or in a circle (rotation invariance).*

Figure 5: *Examples of problems to be addressed in avoiding consistency issues. The ego vehicle's ground truth is in green, and the communicating vehicles' ground truth and their real communicated polygonal models and bounding boxes are in blue. The LiDAR points are in red, and their bounding boxes in orange. Finally, the uncertainty ellipses of the communicated poses are in green.*
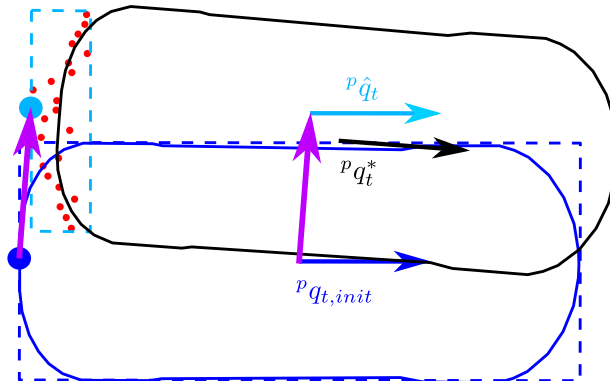
Figure 6: *Fast localization using the deviation (in purple) between the bounding box (in light blue) of the LiDAR cluster (in red) and the bounding box (in dark blue) of the vehicle polygonal model (in dark blue) with their reference points. The initial pose $^{p}q_{t,init}$ can then be corrected to obtain the new estimate $^{p}\hat{q}_{t}$ near the ground truth $^{p}q_{t}^{*}$.*
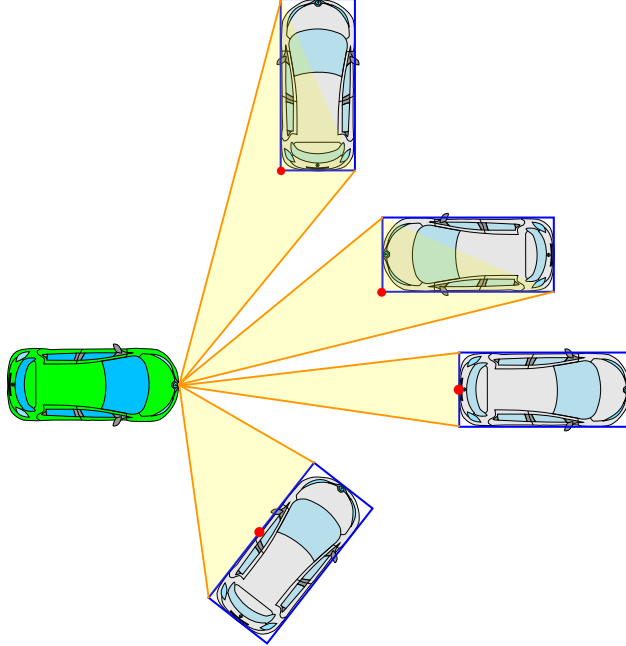
Figure 7: *The ego vehicle's (in green) perception cones (in orange) are estimated from the bounding boxes of the communicated models (in blue). The reference point (in red), can then be deduced in order to compare the bounding boxes of the LiDAR cluster and the vehicle model.*

it is possible to deduce whether the perceiving vehicle is detecting mainly the rear, the front, one side, or one angle of the target vehicle. The center of the rear, the front, one side, or one corner of the bounding boxes can then be taken as a common reference point (see Figure 7). The relative position between these two points gives a correction for the relative pose.

This method does not correct the orientation error, and a small position error remains. However, the remaining error is small and the initial condition of the minimization problem is near the optimal solution.
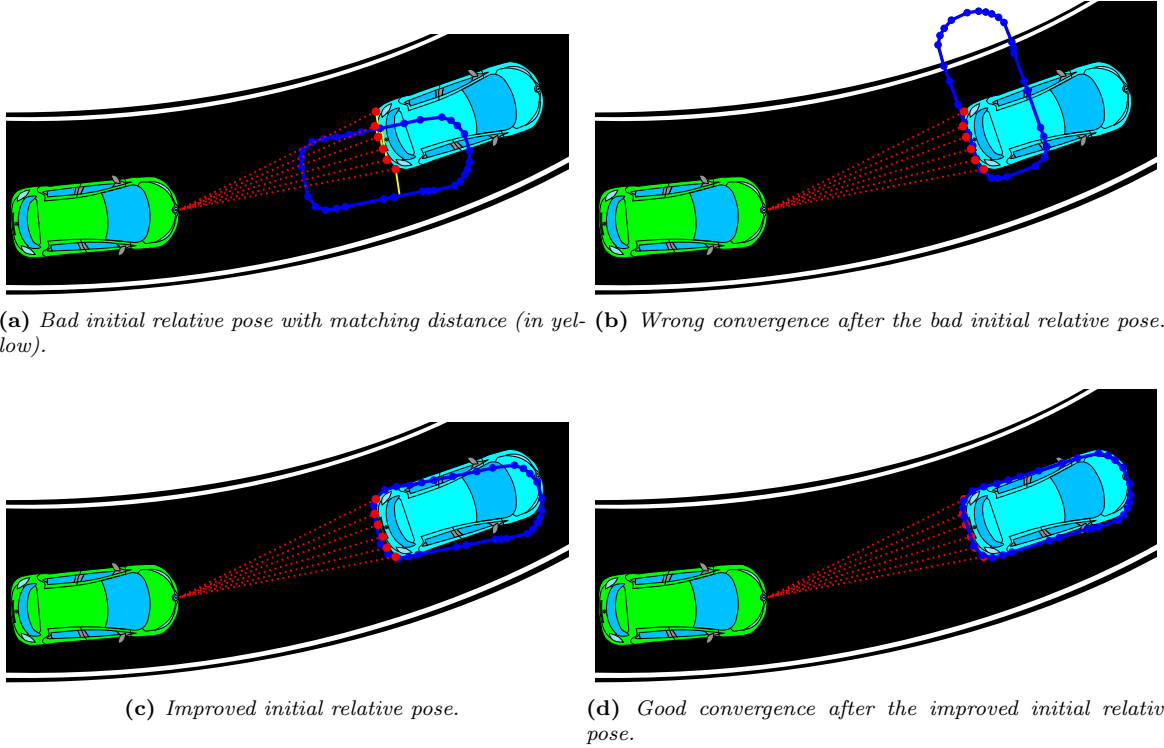
5.2.2 Wrong association rejection

The detection part of our LiDAR perception module is a simple geometric clustering of 2D points. It is then necessary to make an association and to identify the cluster to be linked to a communicating vehicle. If we choose the cluster that is closest to the initial relative pose, there is no guarantee that it actually corresponds to the target vehicle. A cluster on the border of the road can easily be mistaken for the target vehicle. Consequently it is wiser to keep all the clusters in the uncertainty ellipse of the initial relative pose, as illustrated in Figure 5a. Once the optimization has converged, the resulting residual is representative of the matching of the LiDAR points with the model. If the cluster from a bush at the side of the road is matched with the model of the vehicle, the residual is large. This residual can then be compared to a threshold computed from the accuracy of the LiDAR measurements. If the closest cluster is rejected by this test, the minimization can be applied to the second-closest cluster, and so on, until a good cluster is found.

This method works when the cluster and the model are sufficiently different. However, when the cluster corresponds to the LiDAR points of another vehicle with a similar shape, this test may fail. For example, if the communicating vehicle (in blue in Figure 5b) is occluded by a non-communicating vehicle (in gray in Figure 5b), and if the position error and uncertainty ellipse are large, these two vehicles may be confused. In practice, it does not occur if the ellipse is sufficiently small..

In practice, since the LDM is improved by LiDAR perceptions, it often becomes accurate enough and this

(a) *Bad initial relative pose with matching distance (in yellow).*

(b) *Wrong convergence after the bad initial relative pose.*

(c) *Improved initial relative pose.*

(d) *Good convergence after the improved initial relative pose.*

Figure 8: *Wrong and good convergence after the improved initial relative pose estimation. The ego vehicle's ground truth is in green, and the communicating vehicle's ground truth and their real communicated polygonal models and bounding boxes are in blue. The LiDAR points are in red.*

test is not necessary. It is only used to reject wrong associations at the beginning of the cooperative process when the LDM is inaccurate because other vehicles have not yet been perceived.

### 5.2.3 Wrong convergence rejection

If the initial relative pose used for the ICP is too far removed from the real relative pose, the optimization can converge to a local minimum and the optimal solution will be missed.

The LiDAR points at the rear of the vehicle can, for example, converge on a side of the polygonal model (Fig. 5c). As the ICP gives an estimate of the relative pose and its covariance matrix, a Mahalanobis distance can be computed between this relative pose $^p\hat{q}_t$ and the initial relative pose $^pq_{t,init}$:

$$D = \sqrt{\left(^pq_{t,init} - {}^p\hat{q}_t\right)^T \left(\Sigma_{^pq_{t,init}} + \Sigma_{^p\hat{q}_t}\right)^{-1} \left(^pq_{t,init} - {}^p\hat{q}_t\right)}. \tag{39}$$

This distance can then be compared to a threshold. In practice, a simple threshold on the heading error $\left|^p\theta_{t,init} - {}^p\hat{\theta}_t\right|$ is enough to reject wrong convergence.

When the improved initial relative pose estimation is used, the initial condition is close to the optimal solution (Figure 8c). Therefore, the minimization problem can converge to this solution and avoid a local minimum (Figure 8d). The LiDAR points at the rear of the vehicle could, for example, converge to the side of the model (Figure 8b) if these points are initially in the middle of the model (Figure 8a). In practice, the test on the heading error is therefore not often used where the improved initial relative pose estimate is computed.
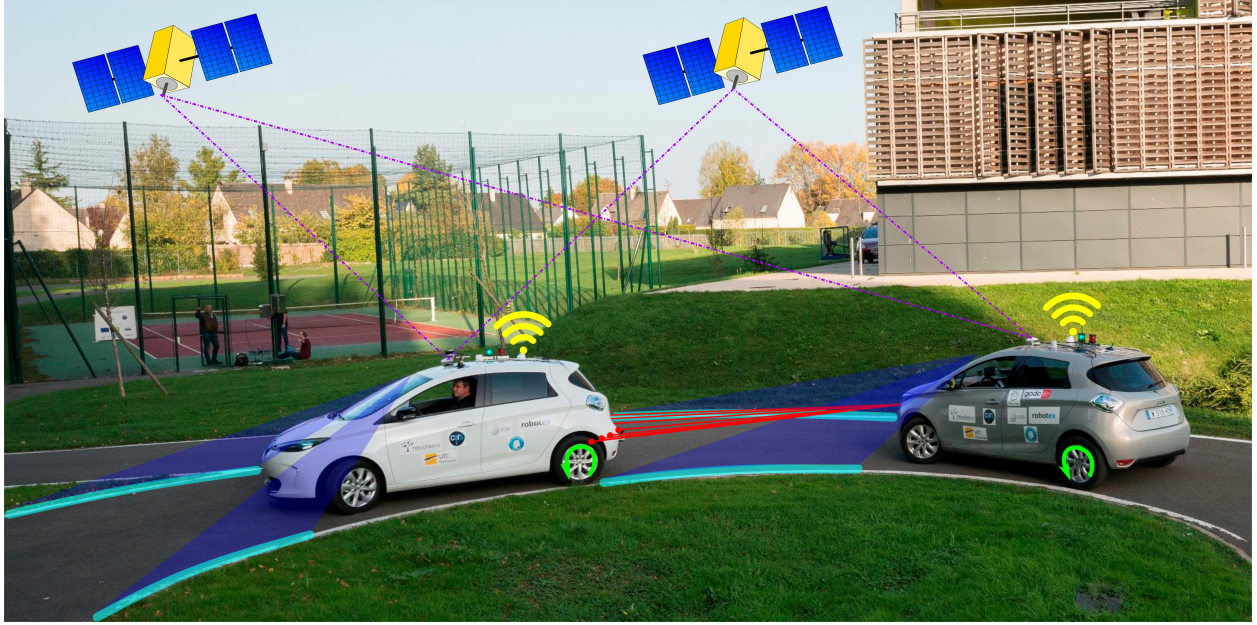
Figure 9: *Platooning of two autonomous Renault ZOEs on the UTC Seville test track. Both vehicles use GNSS localization from a ublox 8 receiver (in purple). Direct Wifi 802.11p communication (in yellow) is simulated in this work, as is the detection of lane markings (in blue). The longitudinal speed and yaw rate (in green) are obtained from the CAN bus. Finally, the tracking vehicle is equipped with a SICK LD-MRS LiDAR (in red).*

5.2.4 Bad condition number detection

When only few LiDAR points are detected, or if they are aligned on one single line, the optimization problem can be badly conditioned.

Translation or rotation invariance can create a bad convergence of the ICP, as illustrated in Figure 5d. The condition number of the matrix A of the optimization can be used to avoid this issue:

$$\text{cond}(A) < 10^5, \tag{40}$$

Another solution is to compute the determinant of the Hessian matrix that we want to inverse in order to estimate the covariance matrix:

$$\det(A^T A) > 10^{-5}. \tag{41}$$

This rough thresholds are sufficient to detect the issue. These tests are even sensitive enough to identify the difference in shape detected by the LiDAR.

6 Experimental results

For our experiments, we used two Renault ZOE experimental vehicles driving on the Seville test track at the University of Technology of Compiègne in France (see Figure (9)). For the perception part, we used the horizontal layer of a 4-layer SICK LD-MRS LiDAR sensor. For global localization, we used a ublox EVK M8T, a low-cost single frequency GNSS receiver using GPS and GLONASS constellations. Additionally, we used longitudinal speed and yaw rate provided by the ESP (Electronic Stability Program) through the vehicle's CAN bus. For ground truth, both vehicles were equipped with a NovAtel SPAN-CPT GNSS/INS

receiver with RTK (Real Time Kinematic) corrections. The full experimental dataset used in this work is publicly available[1].

To measure the performance of the pose estimation process in terms of consistency, we measured the coverage ratio C with a χ^2 test at a given confidence level $1 - \alpha \in [0, 1]$. Let $\widehat{q}$ be an estimate of a pose q with ground truth value q^*. Let $\Sigma_{\widehat{q}}$ and Σ_{q^*} be the covariance matrices associated with $\widehat{q}$ and q^*, respectively. For simulated data, the ground truth was considered perfect and Σ_{q^*} ignored. However, for real data, given the high precision of the LiDAR measurements, it was important to include the covariance Σ_{q^*} of the ground truth value, given that its uncertainty, even with RTK corrections, was not negligible. The estimate $\widehat{q}$ was deemed to be consistent with respect to q^* at the confidence level $1 - \alpha$ if the following condition was satisfied:

$$(q^* - \widehat{q})^T (\Sigma_{q^*} + \Sigma_{\widehat{q}})^{-1} (q^* - \widehat{q}) < \chi^2_{3;\alpha}, \tag{42}$$

with $\chi^2_{3;\alpha}$ the quantile of the χ^2 distribution for 3 degrees of freedom (x, y, θ) and with a $1 - \alpha$ confidence. For a set of estimates, the coverage ratio C was defined as the ratio of samples satisfying Equation 42. The estimation process was deemed consistent if $C > 1 - \alpha$. For the remainder of the paper we set $1 - \alpha = 95\%$.

We first did some experiments in simulation to give us an idea of the performance of our LiDAR-based relative localization. Some real data were then processed offline from a record and replay scheme using the ROS (Robot Operating System) middleware. The replay feature enabled us to simulate the communication between the two vehicles.

6.1 Relative localization

To test the computation of the relative pose between the two vehicles, we simulated the perceptions from a polygonal model with 280 edges generated with a cubic interpolation of the polygonal model used for the minimization. A Gaussian noise with a standard deviation of 0.1 m was added to the ranges of the simulated LiDAR. The initial relative poses were simulated with an inter-distance of 10 m and a Gaussian noise of 0.5 m on the along- and cross-track axes and of 5° for the heading.

First, we compared our solution, which uses a point-to-line projection along with a pseudo-inversion based minimization, with the polynomial minimization proposed in (Censi, 2008). We also compared our results to another matching metric referred to as point-to-projection. Each LiDAR point is projected onto the polygonal model in order to obtain the closest point of the model matched to this LiDAR point. Table 1 details the performance of the different configurations. It can be seen from these results that the proposed pseudo-inverse minimization and point-to-line matching gives the best consistency, even though the metric distance may be less accurate. Further experimental results with other matching metrics can be found in (Héry et al., 2018).

In order to have a deeper insight into the behavior of the relative perception, we decomposed the perception configuration further into two separate scenarios. When following another vehicle, there are typically two types of configuration. The first type of configuration (see Figure 10a) occurs where one vehicle is following another in a straight lane. Only the rear of the leader vehicle is in the field of view of the follower vehicle's LiDAR. In the second type of configuration (see Figure 10b), the two vehicles are traveling in a curved lane, with the rear and one side of the leader vehicle in the field of view of the LiDAR.

Table 2 reports, for the two considered configurations, the coverage ratio C as well as the average absolute error for each of the pose components: x (along-track), y (cross-track) and θ (relative angle). Table 2 shows that the heading error decreases significantly between the straight and the curved configuration. When only the rear of the vehicle is visible, a rotation invariance appears in the optimization problem. This invariance also has an effect on the position error, particularly in the cross-track direction. In the along-track direction, the error remains constrained by the perception of the rear of the vehicle. The additional consistency

---

[1] The "2018_07_27_Cooperative_Localization" dataset is accessible at https://datasets.hds.utc.fr.

Table 1: *Comparison between the proposed point-to-line and a point-to-projection matching, and between the proposed pseudo-inverse minimization and a polynomial minimization. The mean of the norms of the position errors $\|\bar{\epsilon_p}\|$ and the mean of the absolute value of the orientation error $|\bar{\epsilon_\theta}|$ give the accuracy of the method, and the consistency is computed from the coverage rate C.*

		Point to projection	Point to line
Polynomial minimization	$\|\bar{\epsilon_p}\|$ (cm)	7.8	13.7
	$\|\bar{\epsilon_\theta}\|$ (°)	2.84	5.97
	C (%)	58.8	69.8
Pseudo-inverse minimization	$\|\bar{\epsilon_p}\|$ (cm)	7.8	11.5
	$\|\bar{\epsilon_\theta}\|$ (°)	2.83	5.64
	C (%)	83.9	91.6

(a) *Straight lane.* **(b)** *Curved lane.*
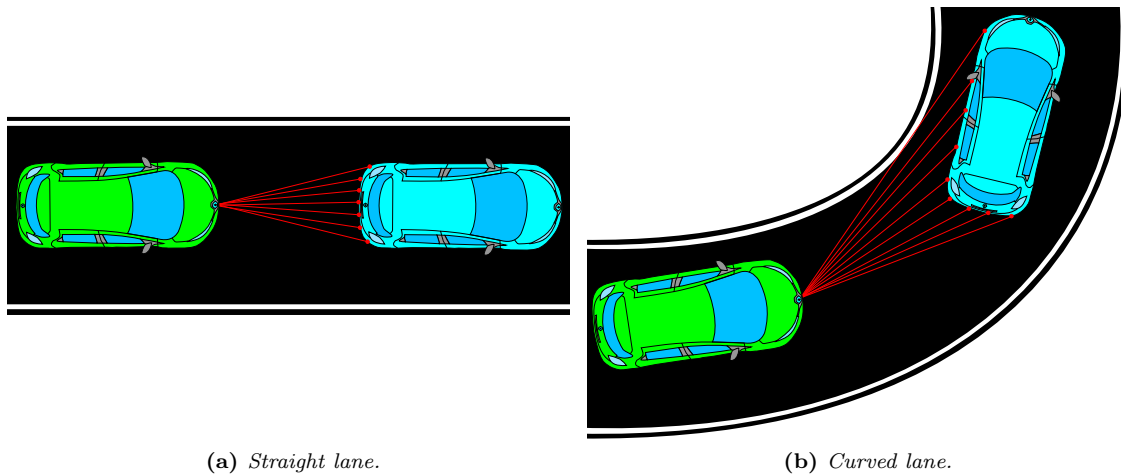
Figure 10: *Straight and curved lane scenarios with a leader vehicle (in blue) and a follower vehicle (in green) equipped with a LiDAR (in red).*

Table 2: *Average absolute errors $|\bar{\epsilon_x}|$, $|\bar{\epsilon_y}|$ and $|\bar{\epsilon_\theta}|$ and coverage rate C of the relative pose estimates on straight and curved lane scenarios*

	$|\bar{\epsilon_x}|$ (cm)	$|\bar{\epsilon_y}|$ (cm)	$|\bar{\epsilon_\theta}|$ (°)	C (%)
Straight lane	2.9	10.5	5.6	91.6
Curved lane	2.4	3.0	0.9	91.2

robustification steps improve the consistency of the relative pose estimation. The coverage rate remains a little below the desired coverage rate of 95%. This is mainly due to the approximate covariance matrix computed from the Hessian in Equation 37. Nevertheless, we believe the consistency is more than acceptable for our problem, and it should be remarked that it does not differ much between the two configurations. This demonstrates that even though the relative pose is estimated less accurately in the straight lane configuration, its uncertainty is reasonably well taken into account in its covariance matrix, and this is very important in practice.

6.2 Absolute localization

A GNSS receiver is an inexpensive means of providing an absolute localization. The heading angle of a vehicle can be computed by the receiver from the Doppler measurements (assuming non-slip rolling). As for the position, without differential corrections such as RTK, a bias is very likely to appear in the estimated position computed by the receiver. For different vehicles traveling in the same vicinity, their GNSS biases are highly correlated. Using relative pose measurements between them is consequently not enough to correct the estimates. To be able to make use of the positions computed by the GNSS receivers, we add a 2-dimensional bias estimation for each vehicle: $b_e[t] = \begin{bmatrix} b_x[t] & b_y[t] \end{bmatrix}^T$. Since the bias changes slowly over time, we chose a constant evolution model $b_e[t|t_{-1}] = b_e[t_{-1}|t_{-1}] + \epsilon_b \Delta t$ with the error ϵ_b independent of the time Δt.

The observation model is the pose corrected with the bias:

$$\text{obs}_{\text{GNSS}}\left(X[t|t_{-1}]\right) = q_e[t|t_{-1}] + \begin{pmatrix} b_e[t|t_{-1}] \\ 0 \end{pmatrix}. \tag{43}$$

To estimate the position bias of a GNSS receiver, additional sources of information are needed. Since we are using an HD map, the vehicle is able to detect some features that are geo-referenced in the map. In our case study, the vehicle measures its lateral distance to the center of the lane, e.g., by using a camera. This measure provides a bias estimate only in a one-dimensional space, i.e., orthogonal to the lane marking, but this is enough to estimate the 2D bias if the path of the vehicle contains a turn. In these experiments, this observation was simulated from the lateral distance computed to the center of the lane on the HD map with the ground truth position, because we were not using a vision system accurate enough to measure lateral deviation in the high-curvature roundabouts of our test track. A Gaussian noise of 20 cm standard deviation was added to this lateral distance, this being the typical accuracy of this kind of measurement.

We use the map matching presented in (Héry et al., 2017), based on the Lanelet projection (Bender et al., 2014), in order to obtain the Frenet frame with the closest origin (for this matching) of $p_e[t|t_{-1}]$, $\left[p_{frenet}, \overrightarrow{t}_{frenet}, \overrightarrow{n}_{frenet} \right]$. Where p_{frenet} is the origin of the frame, $\overrightarrow{t}_{frenet}$ the vector tangent to the center of the lane and $\overrightarrow{n}_{frenet}$ the normal vector.

The observation model that gives the lateral signed deviation is then a projection onto this normal vector:

$$\text{obs}_{CAM}\left(X[t|t_{-1}]\right) = \left(p_e[t|t_{-1}] - p_{frenet}\right)^T \cdot \overrightarrow{n}_{frenet}. \tag{44}$$

To illustrate the effect of having a biased position estimation in terms of consistency, we use a graphical representation inspired from the Stanford diagram, which is widely used for integrity performance measurement. In Figure 11, the x-axis represents the norm of the 2D position estimation error computed with respect to the ground truth. The y-axis represents the estimate bounds of the errors for a 95% confidence level using a Rayleigh distribution to approximate the norm in the position error. If a point is situated below the diagonal line $y = x$, this means that the actual error is larger than the estimated bound, i.e., the estimate is not consistent. For an estimation process to be consistent at a given confidence level $1 - \alpha$, the ratio of points below the diagonal should not exceed α. For a deeper performance analysis, we also provide the empirical distribution for both the errors and the bounds as well as a box plot over the errors.
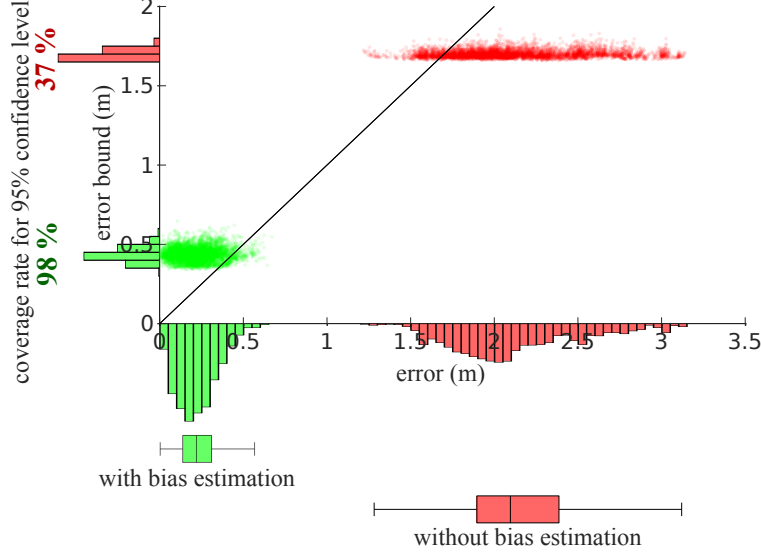
Figure 11: *Stanford-like diagram of the position error without (in red) and with (in green) the GNSS bias estimation.*

In Figure 11, the red points were provided directly by the ublox GNSS receiver. It can be seen that without a bias estimation process, almost all the points are situated below the diagonal (63%), resulting in inaccurate and inconsistent position estimates. Using the bias estimator, we obtain the green points. Performance can be seen to improve significantly, as regards both accuracy and consistency. However, the resulting pose is pessimistic, since the coverage rate is 98%. The covariance matrix of the GNSS pose observation was computed using the "horizontal accuracy" and "heading accuracy" values provided by the GNSS receiver. We remark that the definitions of these fields are not clearly specified in the ublox documentation, and it is our conjecture that these values do not correspond precisely to standard deviations.

6.3 Cooperative localization with LDM

Once the consistency of the absolute and relative pose estimates has been achieved, the cooperative localization with LDM can be put in place.

Kinetic observations To be able to use the evolution model (Eq. 4), the longitudinal speed and the yaw rate need to be estimated. The ESP estimates this information from a gyrometer and wheel encoders. Since the kinetic component u_e of the ego vehicle is observed directly, the observation model of the EKF update is simply

$$\text{obs}_{\text{CAN}}\left(X\left[t|t_{-1}\right]\right) = u_e\left[t|t_{-1}\right] = \left[v_e\left[t|t_{-1}\right] \ \omega_e\left[t|t_{-1}\right]\right]^T. \tag{45}$$

LDM observations If an LDM received by the ego vehicle contains the same list of vehicles as its own LDM in the same order, then the observation model is simply the identity function:

$$\text{obs}_{\text{CAN}}\left(X\left[t|t-1\right]\right) = X\left[t|t-1\right]. \tag{46}$$

To avoid any data incest, this observation model is used with the CIF update. If new vehicles are provided by the communicated LDMs, then the ego vehicle needs to increase the size of its own LDM state vector and initialize the newly added vehicles using the values provided by the communicated LDMs.

Relative pose observations The observation from the ego vehicle of the relative pose ${}^{i}q_e$ of a vehicle i provides a geometric constraint between the two pose components q_e and q_i in the LDM state vector. The correlation between the poses of these two vehicles introduced by this observation is also taken into account

Figure 12: *Test track with the nine laps traveled by the leader (in blue) and the follower (in green) vehicles. (Leaflet | Tiles © Esri - Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community).*

in the LDM. Without this measurement the problem would be equivalent to a localization and tracking problem (Chieh-Chih Wang et al., 2003; Vaughan et al., 2000; Stegagno et al., 2013). The observation model is obtained from the $\oplus$ and $\ominus$ operators:

$$\text{obs}_{\text{LiDAR}} \left(X \left[t | t_{-1} \right] \right) = \ominus q_e \left[t | t_{-1} \right] \oplus q_i \left[t | t_{-1} \right]. \tag{47}$$
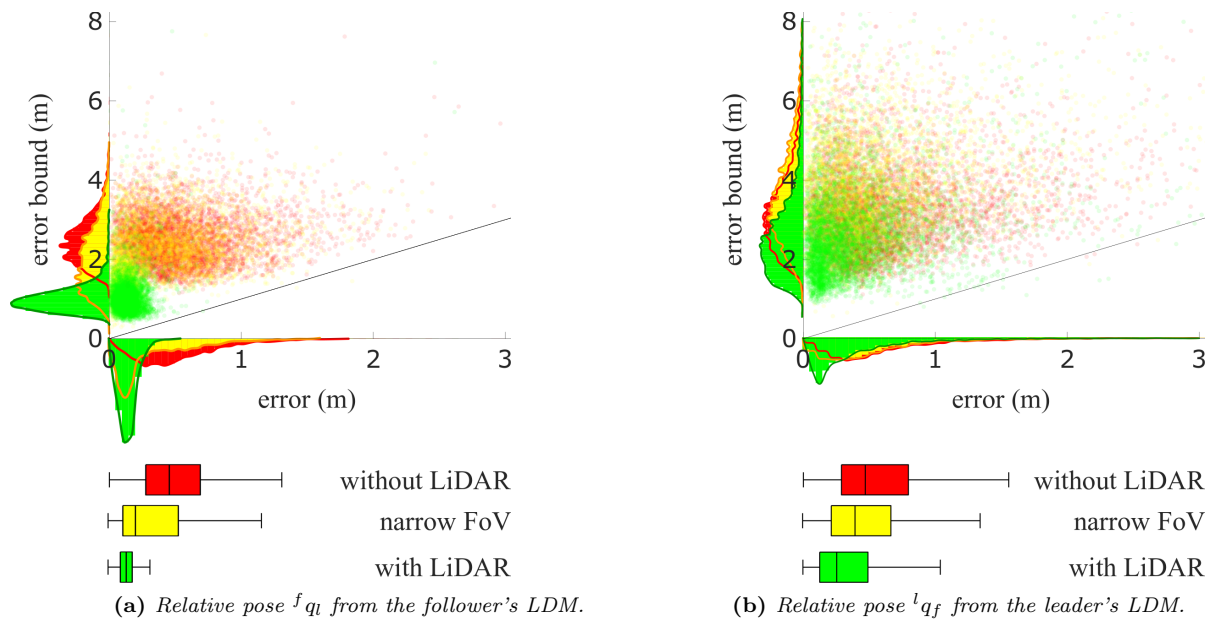
To test the cooperative localization, we drove the two ZOE vehicles in a platooning configuration, that is to say with one vehicle following the other, on a test track comprising two roundabouts linked by a straight lane (Fig. 12). We did nine laps during a ten-minute drive (this corresponds to the data set that we are going to make public with ROS bags and .csv and readable in Python). In this platooning configuration, the preceding vehicle is referred to as the leader and denoted as l, while the other is called the follower and denoted as f. Only the follower is able to measure the relative pose w.r.t. the leader. Because the LDMs are computed in a decentralized manner, each vehicle possesses its own estimate of the LDM containing the states of both vehicles.

Although the relative poses are improved greatly, the absolute poses remain similar (see table 3).

To provide a clearer idea of the contribution of the LiDAR measurements, Figure 13 shows the performances of the relative pose computed from the two LDMs. In order to simulate classical highway platooning scenarios where one vehicle follows another in a straight lane, we limited the LiDAR to a narrow field of view (FoV) so that no measurements were available in the roundabout sections. Figure 13a shows that from the follower's perspective, the relative pose estimation is significantly improved when using the LiDAR. With the narrow FoV, the improvement is less significant. In fact, as described in Section 6.1, the LiDAR performance is worse in the straight lane configuration. From the leader's perspective (Fig. 13b), it can be seen that the relative pose is also improved, although the leader does not perceive the following vehicle directly. This illustrates and quantifies the value of cooperative localization. The improvement results from the communication of other vehicle's poses via LDMs. With the communication of the state of the ego vehicle only, it would not be possible.

Table 3: *Position $\|\bar{\epsilon_p}\|$ and heading $|\bar{\epsilon_\theta}|$ error and coverage rate C of the cooperative localization with and without LiDAR perception.*

	Without LiDAR		With LiDAR	
	q_f	q_l	q_f	q_l
$\|\bar{\epsilon_p}\|$ (m)	0.22	0.23	0.20	0.25
$|\epsilon_\theta|$ (°)	2.20	3.66	1.23	2.47
C (%)	98.4	99.8	96.8	94.9

(a) *Relative pose $^f q_l$ from the follower's LDM.*

(b) *Relative pose $^l q_f$ from the leader's LDM.*

Figure 13: *Stanford-like diagrams for the relative position errors of $^f q_l$ and $^l q_f$ without (in red) and with (in green) LiDAR measurements and with a limited field of view (in yellow).*

In the absence of relative pose measurements, the poses of the different vehicles could be known thanks to the communication, but they would remain independent w.r.t. each other. Therefore the results would be no different from the case where there is no communication. To maintain consistent LDMs, each vehicle needs to provide a consistent absolute pose estimate. With relative pose measurements, some geometric constraints are added between the components of the LDMs. If the relative pose estimate is also consistent, then the cooperative localization is improved overall, in terms of both accuracy and consistency.

It can be seen from table 4 that even where one vehicle is not able to detect lane borders, the communication of the LDM with a vehicle that it perceives, or that it is perceived by, can be used to estimate the bias. The position error is between 0.2 m and 0.4 m, rather than between 1.5 m and 2 m, which is the case when the bias is not estimated. These results also show that the estimate is more consistent when the car is perceived than when it perceives.

6.4   Computation time and scaling up

To evaluate the real-time performance of the system, the computation time was recorded. The code was developed in Python, and computation time could potentially be greatly optimized.

Table 5 presents the computation times for the different functions used in our global algorithm. The evolution steps of the data fusion take around 0.2 ms. The observation steps depend greatly on the size of the observation and on the filter used (EKF or CIF). The LDM data fusion with the CIF is seen to take more time than the others (15.42 ms). The processing needed for some observations is also recorded. The map matching takes 0.64 ms and the clustering and the PLICP take 4.09 ms and 10.07 ms for a total of 14.16 ms. As the computation time of the processing of the relative pose respects the real-time constraint with a large margin, it could run without any problem with five vehicles in the FoV.

The computation time for the observation steps of the LDM with our CIF implementation could, however, be a problem, given that the size of the observation is $5N$, with N the number of vehicles in the LDM. To examine this issue further, we simulated the computation time for the covariance matrix update of the LDM using covariance matrices corresponding to a fleet of between two and ten vehicles with four different CIF algorithms. 10000 couples of random covariance matrices were simulated for each number of vehicles, and the mean of the computation time was recorded. The classic CIF (see Equation 52 of Appendix B) and the Joseph form (used in this paper) were computed using the iterative optimization used in this paper, and using a fast optimization (Niehsen, 2002) that simply computes an approximation of the weight as follows:

$$\omega = \frac{\det\left(\Sigma_Z\right)}{\det\left(\Sigma_X\right) + \det\left(\Sigma_Z\right)}. \tag{48}$$

As the number of vehicles increases, the number of LDM updates increases (the computation time is multiplied by the number of neighbors, i.e. $N-1$).

Figure (14) shows that when the number of vehicles is greater than five, the computation time exceeds a 10 Hz communication limit (in red) with our CIF implementation. This implementation is therefore only able to handle five-vehicle LDMs.

Other considerations might be made for scaling. For example, the size of the exchanged LDMs might be reduced, or only the most informative vehicles used in the calculation. Consider for instance the LDM illustrated in Figure (15). The relative pose update needs only the vehicles on the dark background. The other vehicles do not improve the ego vehicle state directly, although they may improve the perceived vehicle state, which could subsequently improve the ego vehicle state. The other vehicles may be useful for some navigation tasks such as crossing intersections, given that they are in the LDM but can be removed from the CIF update to reduce computing time.

Table 4: *Position $\|\bar{\epsilon_p}\|$ and heading $|\bar{\epsilon_\theta}|$ error and coverage rate C when GNSS bias cannot be estimated from the lane borders by the follower or by the leader.*

	Lane borders undetected by the follower				Lane borders undetected by the leader			
	LDM of f		LDM of l		LDM of f		LDM of l	
	q_f	q_l	q_l	q_f	q_f	q_l	q_l	q_f
$\|\epsilon_p\|$ (m)	0.23	0.28	0.26	0.24	0.20	0.39	0.39	0.21
$|\epsilon_\theta|$ (°)	1.35	2.47	2.55	1.26	1.54	2.58	2.26	1.47
C (%)	90.4	93.7	93.2	90.6	96.4	94.0	93.5	96.7

Table 5: *Computation time for different elementary functions (ms).*

Function	Computation time (ms)
LDM extrapolation	0.24
Kinetic EKF update	1.24
GNSS EKF update	2.79
LDM CIF update	15.42
Relative pose EKF update	4.87
Lane borders EKF update	0.78
Map matching	0.64
Clustering	4.09
PLICP	10.07
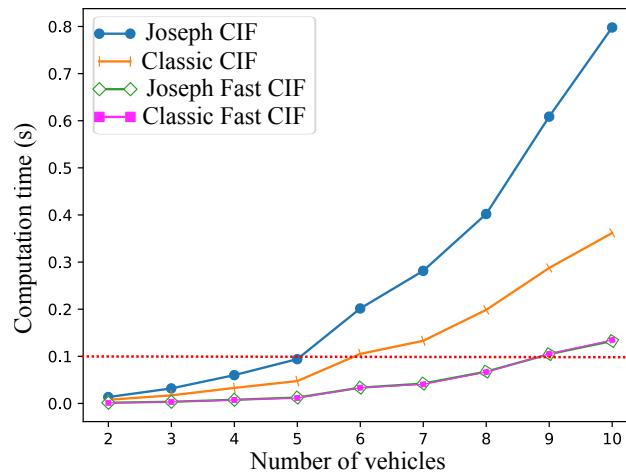
Figure 14: *Classic and Joseph CIF and their fast implementations. Computation times for two to ten vehicles with the 10 Hz limit (in red). The pink and green curves are very close to each other.*
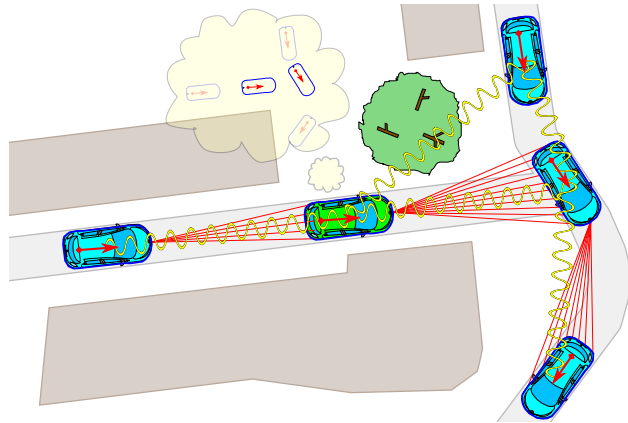
Figure 15: *LDM reduced to the ego and the perceived vehicles (in dark in the cloud).*

7 Conclusion and future work

This work has presented an embedded communicating cyberphysical system for cooperative localization by exchanging local dynamic maps between neighbor vehicles. Our method is based on extended Kalman filter and covariance intersection filter mechanisms to manage the correlation of error estimates between the LDMs of different vehicles. The states of vehicles in the neighborhood are included in the LDM, since this is essential for many navigation tasks. The algorithm merges multiple sources of information: kinematic data from the vehicle sensors, GNSS poses, relative poses between vehicles computed from LiDAR points, lane edges that can be detected by a camera, and LDMs of other vehicles received by communication. The importance of a GNSS bias estimation for good accuracy and consistency has been shown through lane edge detection. Experimental results have shown that the use of a consistent and robust LiDAR perception system to link the poses of different vehicles significantly improves the poses in the LDM, both absolutely and relatively. We have also shown that by paying attention to the consistency of each step, the resulting overall system remains consistent. Although we used only two vehicles for this paper, we were able to show that adding accurate relative measures between vehicles improves accuracy while maintaining consistency. Where this kind of system is deployed on a large scale, every observation between a perceiving and a perceived vehicle acts as a relative constraint that can be propagated throughout the whole information system. In a situation where all the vehicles in a fleet were able to perceive or to be perceived, these constraints could provide very accurate poses for them all. In future work, it would be interesting to add static elements to the LDM, such as buildings and traffic signs, for example. This approach could improve cooperative localization even when vehicles are unable to perceive each other but can perceive shared static landmarks.

Acknowledgment

A Poses transformations operators

As we use only direct orthonormal frames, the definition of one frame in its reference frame is equivalent to the frame transformation between these two frames and to the pose of this frame in its reference frame. The position corresponds to the origin and the orientation to the one of the x axes of the frame.

The compounding operator $\oplus$ is equivalent to the operator $\circ$ for functions composition. If we consider the poses as the equivalent frame transformation functions :

$$^kq_i =^k q_j \oplus {}^jq_i \tag{49}$$

where $^kq_i = \left(^kx_i,\ ^ky_i,\ ^k\theta_i\right)^T$ is the pose of the frame i in the frame k. One can see with these notations that the compounding can be applied similarly to the segment addition postulate. The operator $\oplus$ can be defined by the following rototranslation :

$$
\begin{aligned}
^kq_i &=^k q_j \oplus {}^jq_i \\
&=^k q_j + \begin{pmatrix} \cos\left(^k\theta_j\right) & -\sin\left(^k\theta_j\right) & 0 \\ \sin\left(^k\theta_j\right) & \cos\left(^k\theta_j\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} {}^jq_i
\end{aligned} \tag{50}
$$

The inverse operator $\ominus$ is equivalent to the inverse of a function, the two letters are reversed. This operator can be defined as follows:

$$
\begin{aligned}
^iq_j &= \ominus^j q_i \\
&= \begin{pmatrix} \cos\left(-^j\theta_i\right) & -\sin\left(-^j\theta_i\right) & 0 \\ \sin\left(-^j\theta_i\right) & \cos\left(-^j\theta_i\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} \left(-^jq_i\right)
\end{aligned} \tag{51}
$$

B Joseph form of the covariance intersection filter

In this work, we use the Joseph form of the CIF. The demonstrations presented in this work are based on the demonstrations of the different forms of the Kalman Filter with the weight $\frac{1}{\omega}$ and $\frac{1}{1-\omega}$ in front of the covariance matrices of the two elements to be fused.

To simplified the equation in this appendix, we use the following notations: $\widehat{X} = X\left[t|t\right]$, $\widehat{\Sigma}_X = \Sigma_X\left[t|t\right]$, $X = X\left[t|t_{-1}\right]$, $\Sigma_X = \Sigma_X\left[t|t_{-1}\right]$, $Z = Z\left[t\right]$ and $\Sigma_Z = \Sigma_Z\left[t\right]$.

When the observation matrix is the identity, one can use the first form of the CIF (Julier and Uhlmann, 1997) :

$$
\begin{cases} \widehat{\Sigma}_X^{-1} &= \omega\Sigma_X^{-1} + (1-\omega)\,\Sigma_Z^{-1} \\ \widehat{X} &= \widehat{\Sigma}_X\left(\omega\Sigma_X^{-1}X + (1-\omega)\,\Sigma_Z^{-1}Z\right) \end{cases} \tag{52}
$$

When the observation matrix is not the identity the following form can be used with the gain K_ω :

$$
\begin{cases} \widehat{\Sigma}_X^{-1} &= \omega\Sigma_X^{-1} + (1-\omega)\,H^T\Sigma_Z^{-1}H \\ K_\omega &= (1-\omega)\,\widehat{\Sigma}_X H^T\Sigma_Z^{-1} \\ \widehat{X} &= X + K_\omega\left(Z - HX\right) \end{cases} \tag{53}
$$

Proposition 1 : The form (53) is equivalent to the form (54) with $\omega \neq 1$ et $\omega \neq 0$:

$$
\begin{cases} K_\omega &= \frac{1}{\omega}\Sigma_X H^T\left(H\frac{1}{\omega}\Sigma_X H^T + \frac{1}{1-\omega}\Sigma_Z\right)^{-1} \\ \widehat{\Sigma}_X &= \left(I - K_\omega H\right)\frac{1}{\omega}\Sigma_X \\ \widehat{X} &= X + K_\omega\left(Z - HX\right) \end{cases} \tag{54}
$$

Proof : With the matrix identity of Woodbury:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(VA^{-1}U + C^{-1}\right)^{-1}VA^{-1}, \tag{55}$$

where $A \in \mathbb{R}^{n \times n}$, $U \in \mathbb{R}^{n \times k}$, $C \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{k \times n}$.

With $A = \omega\Sigma_X^{-1}$, $U = H^T$, $V = H$ and $C = (1 - \omega)\Sigma_Z^{-1}$:

$$\begin{aligned}
\widehat{\Sigma}_X &= \left(\omega\Sigma_X^{-1} + (1 - \omega)H^T\Sigma_Z^{-1}H\right)^{-1} \\
&= \left(I - \frac{1}{\omega}\Sigma_X H^T\left(H\frac{1}{\omega}\Sigma_X H^T + \frac{1}{1-\omega}\Sigma_Z\right)^{-1}H\right)\frac{1}{\omega}\Sigma_X
\end{aligned} \tag{56}$$

if $\omega \neq 1$ and $\omega \neq 0$.

Under this condition, one can develop K_ω :

$$\begin{aligned}
K_\omega &= (1 - \omega)\widehat{\Sigma}_X H^T\Sigma_Z^{-1} \\
&= \frac{1}{\omega}\Sigma_X H^T\left(I - \left(H\frac{1}{\omega}\Sigma_X H^T + \frac{1}{1-\omega}\Sigma_Z\right)^{-1}H\frac{1}{\omega}\Sigma_X H^T\right)(1 - \omega)\Sigma_Z^{-1}
\end{aligned} \tag{57}$$

By changing I into $\left(H\frac{1}{\omega}\Sigma_X H^T + \frac{1}{1-\omega}\Sigma_Z\right)^{-1}\left(H\frac{1}{\omega}\Sigma_X H^T + \frac{1}{1-\omega}\Sigma_Z\right)$:

$$K_\omega = \frac{1}{\omega}\Sigma_X H^T\left(H\frac{1}{\omega}\Sigma_X H^T + \frac{1}{1-\omega}\Sigma_Z\right)^{-1}$$

We have then

$$\widehat{\Sigma}_X = (I - K_\omega H)\frac{1}{\omega}\Sigma_X$$

Proposition 2 : The form (54) is also equal to the Joseph form of the CIF (58) with $\omega \neq 1$ and $\omega \neq 0$:

$$\begin{cases}
\Sigma_{Y,\omega} &= H\left(\frac{1}{\omega}\Sigma_X\right)H^T + \left(\frac{1}{1-\omega}\Sigma_Z\right) \\
K_\omega &= \frac{1}{\omega}\Sigma_X H^T\Sigma_{Y,\omega}^{-1} \\
\widehat{\Sigma}_X &= (I - K_\omega H)\left(\frac{1}{\omega}\Sigma_X\right)(I - K_\omega H)^T + K_\omega\left(\frac{1}{1-\omega}\Sigma_Z\right)K_\omega^T \\
Y &= Z - HX \\
\widehat{X} &= X + K_\omega Y
\end{cases} \tag{58}$$

Proof :   With $Y = Z - HX$ and $\Sigma_{Y,\omega} = H \left(\frac{1}{\omega} \Sigma_X \right) H^T + \frac{1}{1-\omega} \Sigma_Z$, one can obtain directly K_ω and $\widehat{X}$ from Equation 54. By developing $\widehat{\Sigma}_X$ from Equation 58:

$$
\begin{aligned}
\widehat{\Sigma}_X &= (I - K_\omega H) \left(\frac{1}{\omega} \Sigma_X \right) (I - K_\omega H)^T + K_\omega \frac{1}{1-\omega} \Sigma_Z K_\omega^T \qquad (59) \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X - \frac{1}{\omega} \Sigma_X H^T K_\omega^T + K_\omega \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1-\omega} \Sigma_Z \right) K_\omega^T \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X - \frac{1}{\omega} \Sigma_X H^T K_\omega^T + K_\omega \Sigma_{Y,\omega} K_\omega^T \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X - \frac{1}{\omega} \Sigma_X H^T K_\omega^T + \frac{1}{\omega} \Sigma_X H^T \Sigma_{Y,\omega}^{-1} \Sigma_{Y,\omega} K_\omega^T \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X
\end{aligned}
$$

We obtain $\widehat{\Sigma}_X$ of Equation 54.

References

Al Hage, J., Xu, P., and Bonnifait, P. (2019). High Integrity Localization With Multi-Lane Camera Measurements. In *IEEE Intelligent Vehicles Symposium*, pages 1232–1238, Paris, France.

Aufrère, R., Karam, N., Chausse, F., and Chapuis, R. (2010). A state exchange approach in real conditions for multi-robot cooperative localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4346–4351, Taipei, Taiwan.

Bahr, A., Walter, M. R., and Leonard, J. J. (2009). Consistent cooperative localization. In *IEEE International Conference on Robotics and Automation*, pages 3415–3422, Kobe, Japan.

Ben-israel, A. (2002). The Moore of the Moore-Penrose inverse. *Electronic Journal of Linear Algebra*, 9:150–157.

Bender, P., Ziegler, J., and Stiller, C. (2014). Lanelets: Efficient map representation for autonomous driving. In *IEEE Intelligent Vehicles Symposium*, pages 420–425, Dearborn, MI, USA.

Bengtsson, O. and Baerveldt, A.-J. (2003). Robot localization based on scan-matching—estimating the covariance matrix for the IDC algorithm. *Robotics and Autonomous Systems*, 44(1):29–40.

Biswas, P., Liang, T.-C., Toh, K.-C., Ye, Y., and Wang, T.-C. (2006). Semidefinite Programming Approaches for Sensor Network Localization With Noisy Distance Measurements. *IEEE Transactions on Automation Science and Engineering*, 3(4):360–371.

Bounini, F., Gingras, D., Pollart, H., and Gruyer, D. (2020). From Simultaneous Localization And Mapping to Collaborative Localization for Intelligent Vehicles. *IEEE Intelligent Transportation Systems Magazine*.

Brent, R. P. (1971). An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425.

Capellier, E., Davoine, F., Cherfaoui, V., and Li, Y. (2019). Evidential deep learning for arbitrary lidar object classification in the context of autonomous driving. In *IEEE Intelligent Vehicles Symposium*, pages 1304–1311, Paris, France.

Carrillo-Arce, L. C., Nerurkar, E. D., Gordillo, J. L., and Roumeliotis, S. I. (2013). Decentralized multi-robot cooperative localization using covariance intersection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1412–1417, Tokyo, Japan.

Censi, A. (2007). An accurate closed-form estimate of ICP's covariance. In *IEEE International Conference on Robotics and Automation*, pages 3167–3172, Roma, Italy.

Censi, A. (2008). An ICP variant using a point-to-line metric. In *IEEE International Conference on Robotics and Automation*, pages 19–25, Pasadena, CA, USA.

Challita, G., Mousset, S., Nashashibi, F., and Bensrhair, A. (2009). An application of V2V communications : Cooperation of vehicles for a better car tracking using GPS and vision systems. In *IEEE Vehicular Networking Conference*, pages 1–6, Tokyo, Japan.

Chieh-Chih Wang, Thorpe, C., and Thrun, S. (2003). Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 842–849, Taipei, Taiwan.

Coenen, M., Rottensteiner, F., and Heipke, C. (2017). Detection and 3d modelling of vehicles from terrestrial stereo image pairs. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 42, Nr. 1W1*, 42(1W1):505–512.

Cognetti, M., Oriolo, G., Peliti, P., Rosa, L., and Stegagno, P. (2014). Cooperative control of a heterogeneous multi-robot system based on relative localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 350–356, Chicago, IL, USA.

Cruz, S. B., Abrudan, T. E., Xiao, Z., Trigoni, N., and Barros, J. (2017). Neighbor-Aided Localization in Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(10):2693–2702.

Cunningham, A., Indelman, V., and Dellaert, F. (2013). DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *IEEE International Conference on Robotics and Automation*, pages 5220–5227, Karlsruhe, Germany.

Cunningham, A., Paluri, M., and Dellaert, F. (2010). DDF-SAM: Fully distributed SLAM using Constrained Factor Graphs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3025–3030, Taipei, Taiwan.

Delobel, L., Aufrère, R., Debain, C., Chapuis, R., and Chateau, T. (2019). A real-time map refinement method using a multi-sensor localization framework. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1644–1658.

ETSI (2011). Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization. Technical Report ETSI TR 102 863 V1.1.1.

Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A Probabilistic Approach to Collaborative Multi-Robot Localization. *Autonomous Robots*, 8(3):325–344.

Gao, C., Zhao, G., and Fourati, H. (2019). *Cooperative Localization and Navigation: Theory, Research, and Practice*. CRC Press.

Grime, S. and Durrant-Whyte, H. (1994). Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849 – 863.

Héry, E., Masi, S., Xu, P., and Bonnifait, P. (2017). Map-based curvilinear coordinates for autonomous vehicles. In *IEEE Intelligent Transportation Systems Conference*, pages 16–19, Yokohama, Japan.

Héry, E., Xu, P., and Bonnifait, P. (2018). LiDAR based relative pose and covariance estimation for communicating vehicles exchanging a polygonal model of their shape. In *Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, Madrid, Spain.

Hoang, G. M., Denis, B., Härri, J., and Slock, D. (2018). Cooperative Localization in VANETs: An Experimental Proof-of-Concept Combining GPS, IR-UWB Ranging and V2V Communications. In *Workshop on Positioning, Navigation and Communications*, pages 1–6, Bremen, Germany.

Hoang, G. M., Denis, B., Härri, J., and Slock, D. T. M. (2015). Select thy neighbors: Low complexity link selection for high precision cooperative vehicular localization. In *IEEE Vehicular Networking Conference*, pages 36–43, Kyoto, Japan.

Hoang, G. M., Denis, B., Härri, J., and Slock, D. T. M. (2017). Mitigating unbalanced GDoP effects in range-based vehicular Cooperative Localization. In *IEEE International Conference on Communications Workshops*, pages 659–664, Paris, France.

Howard, A., Mataric, M. J., and Sukhatme, G. S. (2003). Putting the 'I' in 'team': An ego-centric approach to cooperative localization. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 868–874, Taipei, Taiwan.

Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001). *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer London Ltd.

Jo, K. H. and Lee, J. (2008). Cooperative localization of multiple robots with Constraint Propagation technique. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3477–3482, Nice, France.

Julier, S. and Uhlmann, J. (2001). General Decentralized Data Fusion with Covariance Intersection (CI). In Hall, D. and Llinas, J., editors, *Multisensor Data Fusion*, volume 3. CRC Press.

Julier, S. J. and Uhlmann, J. K. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conference*, volume 4, pages 2369–2373 vol.4, Albuquerque, NM, USA.

Karam, N. (2009). *Agrégation de données décentralisées pour la localisation multi-véhicules*. PhD thesis, Clermont-Ferrand 2.

Karam, N., Chausse, F., Aufrere, R., and Chapuis, R. (2006a). Cooperative Multi-Vehicle Localization. In *IEEE Intelligent Vehicles Symposium*, pages 564–570, Meguro-Ku, Japan.

Karam, N., Chausse, F., Aufrere, R., and Chapuis, R. (2006b). Localization of a Group of Communicating Vehicles by State Exchange. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 519–524.

Khammari, A., Nashashibi, F., Abramson, Y., and Laurgeau, C. (2005). Vehicle detection combining gradient analysis and AdaBoost classification. In *IEEE Intelligent Transportation Systems*, pages 66–71, Vienna, Austria.

Kieffer, M., Jaulin, L., Walter, É., and Meizel, D. (2000). Robust Autonomous Robot Localization Using Interval Analysis. *Reliable Computing*, 6(3):337–362.

Kim, M., Kim, H. K., and Lee, S. H. (2020). A Distributed Cooperative Localization Strategy in Vehicular-to-Vehicular Networks. *Sensors*, 20(5):1413.

Kurazume, R., Nagata, S., and Hirose, S. (1994). Cooperative positioning with multiple robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1250–1257, San Diego, CA, USA.

Lassoued, K., Bonnifait, P., and Fantoni, I. (2017). Cooperative Localization with Reliable Confidence Domains between Vehicles sharing GNSS Pseudoranges Errors with no Base Station. *IEEE Intelligent Transportation Systems Magazine*, 9(1):22–34.

Li, H. (2012). *Cooperative Perception : Application in the Context of Outdoor Intelligent Vehicle Systems*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris.

Li, H. and Nashashibi, F. (2012). Cooperative multi-vehicle localization using split covariance intersection filter. In *IEEE Intelligent Vehicles Symposium*, pages 211–216, Alcal de Henares, Spain.

Li, H. and Nashashibi, F. (2013). Cooperative Multi-Vehicle Localization Using Split Covariance Intersection Filter. *IEEE Intelligent Transportation Systems Magazine*, 5(2):33–44.

Lin, Y., Vernaza, P., Ham, J., and Lee, D. D. (2005). Cooperative relative robot localization with audible acoustic sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3764–3769, Las Vegas, NV, USA.

Liu, J., Rizos, C., and Cai, B.-g. (2018). Integrity Monitoring of Vehicle Positioning using Cooperative Measurements under Connected Vehicles Environments. In *International Global Navigation Satellite Systems Symposium*, Sydney, Australia.

Low, K.-L. (2004). Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. Technical Report TR04-004, Department of Computer Science University of North Carolina at Chapel Hill.

Madhavan, R., Fregene, K., and Parker, L. E. (2002). Distributed heterogeneous outdoor multi-robot localization. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 374–381, Washington, DC, USA.

Martinelli, A., Pont, F., and Siegwart, R. (2005). Multi-Robot Localization Using Relative Observations. In *IEEE International Conference on Robotics and Automation*, pages 2797–2802, Barcelona, Spain.

Montesano, L., Gaspar, J., Santos-Victor, J., and Montano, L. (2005). Cooperative localization by fusing vision-based bearing measurements and motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2333–2338, Edmonton, AB, Canada.

Mu, H., Bailey, T., Thompson, P., and Durrant-Whyte, H. (2011). Decentralised Solutions to the Cooperative Multi-Platform Navigation Problem. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):1433–1449.

Niehsen, W. (2002). Information fusion based on fast covariance intersection filtering. In *Proceedings of the Fifth International Conference on Information Fusion.*, volume 2, pages 901–904, Annapolis, MD, USA.

Peker, A. U. and Acarman, T. (2017). VANET-Assisted Cooperative Vehicle Mutual Positioning: Feasibility Study. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E100.A(2):448–456.

Prakhya, S. M., Bingbing, L., Rui, Y., and Lin, W. (2015). A closed-form estimate of 3D ICP covariance. In *IAPR International Conference on Machine Vision Applications*, pages 526–529, Tokyo, Japan.

Reinhardt, M., Noack, B., and Hanebeck, U. D. (2012). Closed-form optimization of covariance intersection for low-dimensional matrices. In *International Conference on Information Fusion*, pages 1891–1896, Singapore.

Rife, J. and Xiao, X. (2010). Estimation of Spatially Correlated Errors in Vehicular Collaborative Navigation with Shared GNSS and Road-Boundary Measurements. *International Technical Meeting of the Satellite Division of the Institute of Navigation*.

Rohani, M., Gingras, D., Vigneron, V., and Gruyer, D. (2015). A New Decentralized Bayesian Approach for Cooperative Vehicle Localization Based on Fusion of GPS and VANET Based Inter-Vehicle Distance Measurement. *IEEE Intelligent Transportation Systems Magazine*, 7(2):85–95.

Rosa, L., Cognetti, M., Nicastro, A., Alvarez, P., and Oriolo, G. (2015). Multi-task Cooperative Control in a Heterogeneous Ground-Air Robot Team. *IFAC-PapersOnLine*, 48(5):53–58.

Roumeliotis, S. I. and Bekey, G. A. (2000). Collective localization: A distributed Kalman filter approach to localization of groups of mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2958–2965, San Francisco, CA, USA.

Roumeliotis, S. I. and Bekey, G. A. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795.

Roumeliotis, S. I. and Rekleitis, I. M. (2004). Propagation of Uncertainty in Cooperative Multirobot Localization: Analysis and Experimental Results. *Autonomous Robots*, 17(1):41–54.

Safaei, A. and Mahyuddin, M. N. (2019). Adaptive Cooperative Localization Using Relative Position Estimation for Networked Systems With Minimum Number of Communication Links. *IEEE Access*, 7:32368–32382.

Severi, S., Wymeersch, h., Härri, J., Ulmschneider, M., Denis, B., and Bartels, M. (2018). Beyond GNSS: Highly accurate localization for cooperative-intelligent transport systems. In *IEEE Wireless Communications and Networking Conference*, pages 1–6, Barcelona, Spain.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating Uncertain Spatial Relationships in Robotics. In Cox, I. J. and Wilfong, G. T., editors, *Autonomous Robot Vehicles*, pages 167–193. Springer New York, New York, NY.

Stegagno, P., Cognetti, M., Rosa, L., Peliti, P., and Oriolo, G. (2013). Relative localization and identification in a heterogeneous multi-robot system. In *IEEE International Conference on Robotics and Automation*, pages 1857–1864, Hong Kong, China.

Tischler, K. and Hummel, B. (2005). Enhanced environmental perception by inter-vehicle data exchange. In *IEEE Intelligent Vehicles Symposium*, pages 313–318, Las Vegas, NV, USA.

Vaughan, R. T., Sukhatme, G. S., Mesa-Martinez, F. J., and Montgomery, J. F. (2000). Fly spy: Lightweight localization and target tracking for cooperating air and ground robots. In Parker, L. E., Bekey, G., and Barhen, J., editors, *Distributed Autonomous Robotic Systems 4*, pages 315–324. Springer Japan, Tokyo.

Wanasinghe, T. R., Mann, G. K. I., and Gosine, R. G. (2014). Decentralized Cooperative Localization for Heterogeneous Multi-robot System Using Split Covariance Intersection Filter. In *Canadian Conference on Computer and Robot Vision*, pages 167–174, Montréal, QC, Canada.

Worrall, S. and Nebot, E. (2007). Using Non-Parametric Filters and Sparse Observations to Localise a Fleet of Mining Vehicles. In *IEEE International Conference on Robotics and Automation*, pages 509–516, Roma, Italy.

Zelazo, D., Franchi, A., Bülthoff, H. H., and Robuffo Giordano, P. (2015). Decentralized rigidity maintenance control with range measurements for multi-robot systems. *International Journal of Robotics Research*, 34(1):105–128.

Zelazo, D., Franchi, A., and Giordano, P. R. (2014). Rigidity theory in SE(2) for unscaled relative position estimation using only bearing measurements. In *European Control Conference*, pages 2703–2708, Strasbourg, France.

Zhu, J. and Kia, S. S. (2019). Cooperative Localization Under Limited Connectivity. *IEEE Transactions on Robotics*, 35(6):1523–1530.