LiDAR based relative pose and covariance estimation for communicating vehicles exchanging a polygonal model of their shape

Elwan Héry, Philippe Xu, Philippe Bonnifait

LiDAR based relative pose and covariance estimation for communicating vehicles exchanging a polygonal model of their shape

Elwan Héry, Philippe Xu and Philippe Bonnifait
Sorbonne universités, Université de Technologie de Compiègne, CNRS UMR 7253 Heudiasyc
57 Av. Landshut CS 60319, 60203 Compiègne cedex, France
{elwan.hery - philippe.xu - philippe.bonnifait}@hds.utc.fr

***Abstract*— Relative localization between autonomous vehicles is an important issue for accurate cooperative localization. It is also essential for obstacle avoidance or platooning. Thanks to communication between vehicles, additional information, such as vehicle model and dimension, can be transmitted to facilitate this relative localization process. In this paper, we present and compare different algorithms to solve this problem based on LiDAR points and the pose and model communicated by another vehicle. The core part of the algorithm relies on iterative minimization tested with two methods and different model associations using point-to-point and point-to-line distances. This work compares the accuracy, the consistency and the number of iterations needed to converge for the different algorithms in different scenarios, e.g. straight lane, two lanes and curved lane driving.**

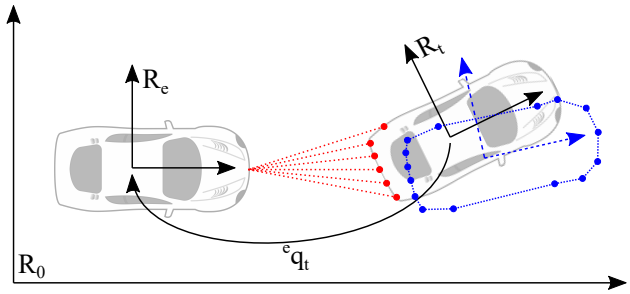


Fig. 1: *The ego-vehicle estimates the rigid transformation ${}^{e}q_t$ that maps the reference frame R_t of a target vehicle into its own reference frame R_e. The target vehicle communicates an estimate of its pose along with a polygonal model of its shape (illustrated in blue with an unavoidable error).*

I. Introduction

Vehicle detection and tracking are a key features for autonomous driving which have led to many research work [7]. Knowing the relative pose of a detected vehicle in the ego-vehicle reference frame is essential for tasks such as obstacle avoidance or platooning.

The emergence of wireless communication capabilities for vehicles in the recent years has given rise to new possibilities. Having access directly to information such as pose or vehicle dimensions, e.g., from the European standard CAM (Cooperative Awareness Message) [5], a vehicle can have a better understanding of its surroundings. Moreover, if a vehicle can receive perception information from other vehicles, it can have an augmented perception of the environment enabling it to see much further. However, in order to transpose perception information of one vehicle into the reference frame of another, the relative pose between these two vehicles is needed.

Many vision based vehicle detection algorithm can be found in the literature with recent deep learning based detectors having impressive performances [4]. However, these methods often only return a bounding box in the image frame and fail at providing a metric estimate of the pose of the detected vehicle. On the contrary, LiDAR based vehicle detection are much more adapted for relative pose estimation. Vehicle detection by fitting a geometrical model such as L-shape fitting [10] provides a good estimate of the relative pose. Because the true shape of a detected vehicle is not known a priori, only simple geometric models, i.e., box, are usually used for model fitting. However in the context of communicating vehicles, it is possible for a vehicle to send an accurate model of its own geometric shape.

An Iterative Closest Point (ICP) algorithm is often used for scan matching [3] but can also be used to fit a scan to a model [9]. ICP provides a good estimate of the relative pose but the associated covariance matrix is often not computed which is crucial to obtain an information that can be used in a data fusion process [2], [8], [1]. With that objective in mind, we present in this paper several algorithms for relative localization based on model matching with covariance estimation.

The paper is organized as follows. In Sec. II we introduce the relative localization problem. In Sec. III, the iterative minimization algorithm is presented with four different matching methods to associate LiDAR points to a geometric model. We also introduce two different minimization methods. Finally, the results of the different matching and minimization methods are compared in Sec. IV.

II. Problem statement

In this work, we aim to estimate the relative pose between two vehicles and quantify its uncertainty. We assume that an ego-vehicle, equipped with a LiDAR sensor, perceives a target vehicle resulting in set of 2D points $P = \{p_i = [x_i, y_i],\ i = 1, \dots, n\}$, in the ego-vehicle reference frame R_e. There exists many algorithms in the literature to compute this cluster of points from a LiDAR point cloud. This computation is out of the scope of this work. In our study,

we also suppose that the target vehicle communicates an estimate of the pose of its reference frame R_t along with a 2D polygonal model, $M = \{m_j = [x_j, y_j],\ j = 1, \dots, N\}$, representing its geometrical shape (see Fig. 1). The points m_j represent the vertices of the model and the edges are defined by two consecutive vertices $(m_j; m_{j+1})$.

This problem can be solved in 3D using a multilayer LiDAR and a 3D model with facets, e.g. STL model. Nevertheless a monolayer LiDAR is less expensive and gives already very good results with smaller computation time and information to communicate. The 2D hypothesis cannot always be respected, i.e. if the vehicles are not driving on a flat road or if the LiDAR scan and the 2D model are not on the same plane, e.g. when the model is at the height of the bumper of a truck and the LiDAR at the height of the bumper of a car. In these cases a 2D polygonal model can be computed from the intersection of a 3D model and the plane of the LiDAR scan.

The problem we aim at solving is to estimate the relative pose ${}^e q_t = [{}^e x_t, {}^e y_t, {}^e \theta_t]$ (θ being the heading) of the target vehicle in the reference frame of the ego-vehicle. In other words, ${}^e q_t$ represents the rigid transformation, i.e., translation and rotation, that maps R_t to R_e. Any point $p_t = [x_t, y_t]$ in R_t can be transformed into R_e as

$$\overline{p}_e = {}^e T_t \overline{p}_t = \begin{bmatrix} \cos({}^e\theta_t) & -\sin({}^e\theta_t) & {}^e x_t \\ \sin({}^e\theta_t) & \cos({}^e\theta_t) & {}^e y_t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} \quad (1)$$

where ${}^e T_t$ is the transformation matrix associated to ${}^e q_t$ and $\overline{p} = [p^T\ 1]^T$ is the homogeneous vector associated to p.

This problem can also be formulated as finding the transformation that would map the target model M to the perceived set of points p_i minimizing a positive scalar error E:

$${}^e\hat{q}_t = \arg\min_q E(q) = \arg\min_q \sum_{i=1}^{n} d(q; p_i, M), \quad (2)$$

where $d(q; p_i, M)$ represents a distance from point p_i to the model M corresponding to the transformation q.
By supposing that the minimization problem is convex, one can also compute the covariance matrix of the error by using an empirical estimate of the variance of the residuals, as proposed in [1]:

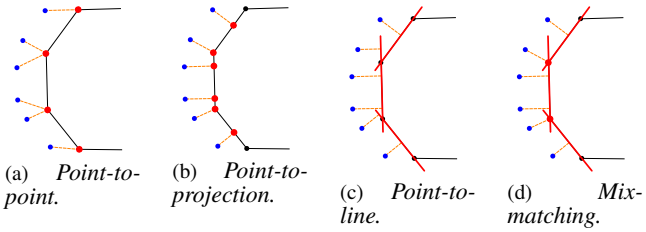
$${}^e\Sigma_t = 2\frac{E({}^e\hat{q}_t)}{n-k}\left(\frac{\partial^2 E}{\partial q^2}({}^e\hat{q}_t)\right)^{-1}, \quad (3)$$

where k is the dimension of ${}^e q_t$, i.e., $k = 3$. This computation needs at least four LiDAR points in the scan ($n \geq 4$).

In this paper, we compare several ways to compute the distance metric d, two different minimization methods within an Iterative Closest Point framework and we evaluate the consistencies of the estimated covariance matrices.

III. Computation of the Relative Pose

ICP is often used to match a LiDAR scan with another one. We used a similar method to estimate the relative pose between the model of the target vehicle and the scan. The objective is to find the pose that minimizes the error between the scan and the shape model.



(a) *Point-to-point.* (b) *Point-to-projection.* (c) *Point-to-line.* (d) *Mix-matching.*

Fig. 2: *Matching between the LiDAR points (in blue) and the model (in black). The matched points and lines are illustrated in red.*

Algorithm 1 *Overview of the proposed method.*

1: Compute a first rough relative pose using bounding boxes of the scan and of the received shape model
Loop
2: Match the clustered scan points with the model
3: Find the pose that minimizes the error
4: Break if the variation of the error divided by the number of LiDAR points is smaller than a threshold
End loop
5: Compute the covariance matrix

Algorithm 1 summarizes the method. In the following, we study four matching methods and two minimization strategies.

A. LiDAR points to model matching

We introduce four different ways to match a set of LiDAR points to a polygonal model.

- Point-to-point, ICP (Fig. 2a): each LiDAR point is matched with the closest vertices of the model. This method does not take into account the edges of the model. A sparsely discretized model may lead to a large distance between the LiDAR points and the model.
- Point-to-projection, ICPP (Fig. 2b): to have a tighter matching, one can match a LiDAR point to the nearest point of the model considering both its vertices and edges. This point can be projected onto the model using the smallest distance : the orthogonal distance to an edge or the Euclidean distance to a vertex.
- Point-to-line, PLICP (Fig. 2c): by matching a point to its orthogonal projection may result in an increase of the error since the matched point remains fixed during the minimization. One way to take this into account is to match the point directly to the line defined by its edge matched using the point-to-projection approach.
- Mix-matching, mixICP (Fig. 2d): the last method is a mix-matching using point-to-point matching when the smallest distance to the model is an Euclidean distance to a vertex and point-to-line matching when it is an orthogonal distance to an edge.

The difference between mixICP and ICPP is subtle. With point-matching methods the matched points are constant

whereas the scan can slide along the model with line-matching.

In the case of a matching between a LiDAR point p_i and a model point m_j (a vertex or an orthogonal projection):

$$d(M, p_i; q) = \|Tm_j - p_i\|^2 = \left\|m_j - T^{-1}p_i\right\|^2, \quad (4)$$

where the T is the transformation matrix associated to q. It should be noted that fitting the model points m_j to the LiDAR points p_i using T is equivalent to fit p_i to m_j using T^{-1}.

In the case of a matching between a LiDAR point p_i and an edge $(m_j; m_{j+1})$ with a unit normal n_j:

$$d(M, p_i; q) = \left(\left(m_j - T^{-1}p_i\right) \cdot n_j\right)^2. \quad (5)$$

B. Minimization using polynomial roots

The error function (2) to minimize is non linear. Censi [3] proposed to change the variable $q = [x\, y\, \theta]^T$ to $q_{4D} = [x\, y\, c\, s]^T = [x\, y\, \cos\theta\, \sin\theta]^T$. By doing so, the minimization of (2) using the distances (4) or (5) can be rewritten as a constrained quadratic problem:

$$\begin{cases} \min_{q_{4D}} & E(q_{4D}) = q_{4D}^T A q_{4D} + B q_{4D} + C \\ \text{subject to} & q_{4D}^T W q_{4D} = 1 \end{cases}, \quad (6)$$

where A, B and C depend on the matched points and

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

To solve this problem, the Lagrangian multiplier λ can be used, resulting in the following function to minimize:

$$L(q_{4D}) = q_{4D}^T A q_{4D} + B q_{4D} + C + \lambda(q_{4D}^T W q_{4D} - 1). \quad (8)$$

The global minimum can then be found by finding the roots of a four degree polynomial in λ. Final, a 3D pose can be computed using

$${}^e\hat{q}_t = [x\, y\, \theta]^T = [x\, y\, \text{atan2}(s, c)]^T. \quad (9)$$

The expression of the covariance matrix ${}^e\Sigma_t$ can be computed from the derivatives of ${}^e\Sigma_t(q_{4D})$ in function of ${}^e\hat{q}_t$.

C. Minimization using pseudo-inverse matrix

Another way to solve (2), proposed by Low [6], is to assume that the angular variation between two consecutive iterations of the ICP is small. Therefore, we can approximate $\cos\theta \approx 1$ and $\sin\theta \approx \theta$. Using this approximation the problem becomes a linear least-squares problem:

$$\min_q E(q) = \min_q \|Aq - b\|^2, \quad (10)$$

where A and b depend on the matched points. A pseudo-inverse matrix can be used to solve the minimization problem 10:

$${}^e\hat{q}_t = \text{pinv}(A)b. \quad (11)$$

TABLE I: *Comparison of the two minimization methods with the four matchings. The accuracy is evaluated from the mean of the norms of the position errors $\|\epsilon\|$ and the mean of the absolute value of the orientation error $|\epsilon_\theta|$.*

		ICP	ICPP	PLICP	mixICP
Polynomial Minimization	$\|\epsilon\|$ (cm)	8.2	7.8	13.7	11.0
	$\|\epsilon_\theta\|$ (°)	2.97	2.84	5.97	5.26
	consistency (%)	85.5	58.8	69.8	70.0
Pseudo-inverse Minimization	$\|\epsilon\|$ (cm)	8.2	7.8	11.5	10.8
	$\|\epsilon_\theta\|$ (°)	2.94	2.83	5.64	5.24
	consistency (%)	93.5	83.9	91.6	89.8

The covariance matrix as defined by (3) is easy to compute here ($n > 3$):

$${}^e\Sigma_t = \frac{E(\hat{q})}{n-3}\left(A^T A\right)^{-1}. \quad (12)$$

IV. Simulation results[1]

We used simulated data to test different parameters with the four matching and the two minimization algorithms. Within the reference frame of the ego-vehicle, the target vehicle is placed 10 meters ahead, i.e., ${}^eq_t = [10\,0\,0]^T$. Gaussian noise has been added to these poses with the standard deviation: $\sigma_x = \sigma_y = 0.5\,\text{m}$ and $\sigma_\theta = 5°$. The LiDAR points have also been simulated with a Gaussian noise added to the range of the LiDAR beams, $\sigma_\rho = 0.1\,\text{m}$.

To test the consistency of the estimated covariance matrix ${}^e\hat{\Sigma}_t$ associated to an estimated relative pose ${}^e\hat{q}_t$ at a given risk $\alpha = 5\%$, we check if the ratio of epochs where the following inequality holds is equal to $1 - \alpha$:

$$({}^eq_t - {}^e\hat{q}_t)^T {}^e\hat{\Sigma}_t^{-1} ({}^eq_t - {}^e\hat{q}_t) < \chi^2_{3,1-\alpha}, \quad (13)$$

where $\chi^2_{3,1-\alpha} = 7.81$ for a three dimensional problem with an error probability $\alpha = 5\%$.

A. Comparison of the two minimization algorithms

One can see on table I the average position error $\|\epsilon\|$ is similar for the two minimization methods excepted for the point-to-line matching where the pseudo-inverse approach is more accurate. The pseudo-inverse method with the point-to-line matching becomes also more consistent. ICP converges to ICPP when the model is very discretized. Its large uncertainty ellipse comes from the minimization error, which is not computed with the shortest distances to the model, but with distances to the points of the model.

B. Bounding box model

In the results of table I, we used the polygonal model, in blue in figure 3. In many works, only the bounding box, in red in figure 3, is known and used.

One can see in table II that the relative poses found are less accurate in position but more accurate in orientation. The back of the bounding box is indeed not curved like the back of the polygonal model. The LiDAR scan is more

[1]The Matlab source-code used for this paper is available at: `https://www.hds.utc.fr/~heryelwa/dokuwiki/en/start`

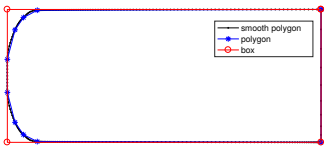


Fig. 3: *Bounding box model, in red, polygonal model, in blue, and smooth polygonal model used to simulate the LiDAR scan. As this model is used for platooning, i.e. only the back and the sides of the vehicle are in the field of view of the LiDAR, only the rear of the vehicle is detailed in the discretization of these models.*

TABLE II: *Results with a bounding box. The pose is supposed to be found when the covariance matrix can be computed without numerical singularity, when the problem has enough constraints.*

		ICP	ICPP	PLICP	mixICP
Polynomial Minimization	$\|\epsilon\|$ (cm)	12.1	9.5	9.9	9.9
	$\|\epsilon_\theta\|$ $(^\circ)$	2.36	3.39	2.97	2.97
	consistency (%)	99.9	31.1	59.9	59.9
	found (%)	100	100	56.4	56.4
Pseudo-inverse Minimization	$\|\epsilon\|$ (cm)	17.1	9.5	11.0	11.0
	$\|\epsilon_\theta\|$ $(^\circ)$	2.39	3.41	4.32	4.32
	consistency (%)	78.1	60.1	64.5	64.4
	found (%)	100	100	99.6	99.6

constrained in orientation and less constrained in position when a bounding box is used instead of a polygonal model.

When using a bounding box model, the problem is less constrained and only one segment can be matched by all the LiDAR points. In this case, the point-to-line matching is not appropriate for the first minimization. With this model and this matching, the pseudo-inverse approach found a pose with a computable covariance matrix more often.

C. Convergences of different iterative methods

Figure 4 shows an example of convergence for one epoch for the four different matching methods. When a threshold of $1cm^2$ is used, all the matching methods converge with three or four iterations on average. The ICP error is larger because it does not use the smallest distance to the model like the other methods. Sometimes, the ICPP seems to converge but after a small variation of the error during several iterations, a smaller error may still be found. When the threshold is $0cm^2$, the ICPP need 243 iterations on average to converge. Indeed, the point-to-point matching methods like the ICP or the PLICP limit the motion of the LiDAR points during the minimization of one iteration. The ICPP recomputes the points to match at every iteration, these points are increasingly closer to the final result, but the error is also increasingly smaller. When point-to-line matchings are used in the PLICP, the LiDAR points have more freedom and the convergence is faster.

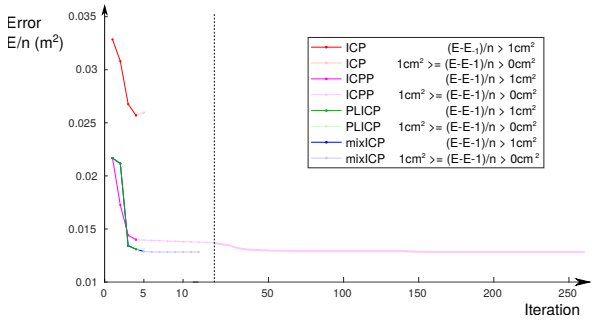
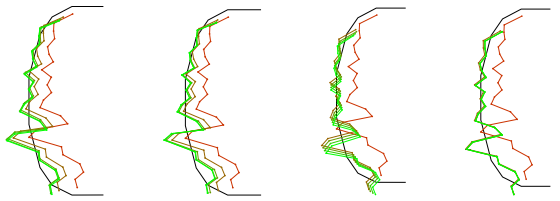


Fig. 4: *Convergences of the minimization errors for the second minimization method and the four different matchings methods. The dark color corresponds to the error before the convergence, for the $1cm^2$ threshold used to stop the algorithm. The variation of the error between two iterations divided by the number matched LiDAR points is compared to this threshold. The light color shows the error before the convergence if the threshold is $0cm^2$.*



(a) ICP (b) ICPP (c) PLICP (d) mixICP

Fig. 5: *LiDAR points convergence on the geometrical polygonal model for the different matchings. The model is shown in black and the LiDAR points change from red, before the first iteration, to green, once the algorithm has converged. 0.0001 used as threshold.*

Figure 5 shows the convergence of the LiDAR points for the four matchings. When a point-to-line matching is used in the PLICP or in the mixICP, the LiDAR points can slide along the model as shown by subfigure c.

D. Scenarios

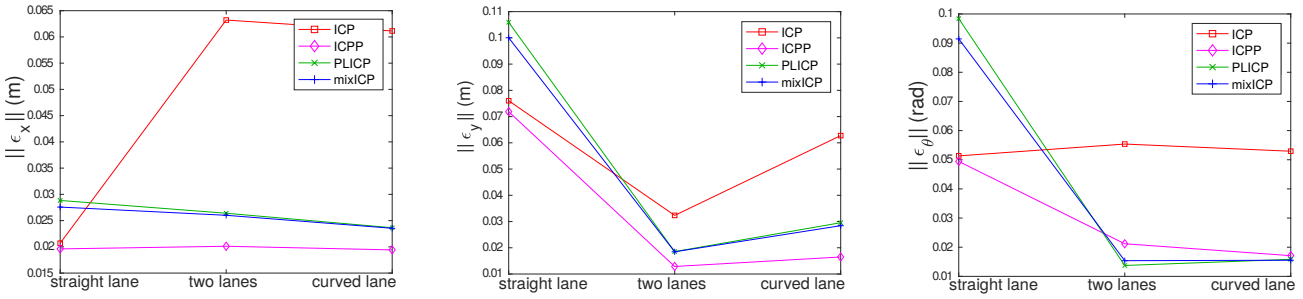
Three different scenarios have been tested on a straight road, a two lanes road and curved road. In the straight road, only the back of the vehicle is detected. The back is slightly curved, a rotation and translation invariance are present. The problem is here badly conditioned. The estimated pose is therefore not very accurate and it is more difficult to obtain the consistency. In a curved road driving, the back and one side of the vehicle are in the field of view of the LiDAR. The estimated pose is accurate. In the two lanes scenario, the back and one side of the vehicle are detected (like for the curved road) and the results are similar.

Figure 6 validates the previous hypothesis. The accuracy for y and θ increases largely when two faces of the vehicle are in the field of view of the LiDAR, e.g., in the two lanes and the curved lane scenarios.

In the figure 7, one can see that the consistency is higher for the two lanes driving than for the two other scenarios.

E. Inter-distance dependency

The previous results has been computed for an inter-distance between the leader vehicle and the follower of 10m. We have also tested the dependency of the error and the consistency for different inter-distances. When the inter-distance increases, the number of LiDAR points on the leader decreases. The error and the consistency become larger (Fig. 8).



(a) $|\bar{\epsilon_x}|$ (b) $|\bar{\epsilon_y}|$ (c) $|\bar{\epsilon_\theta}|$

Fig. 6: *Mean of the absolute value of the errors for x, y and θ for the four matchings and for the straight lane, the two lanes and the curved lane drivings.*

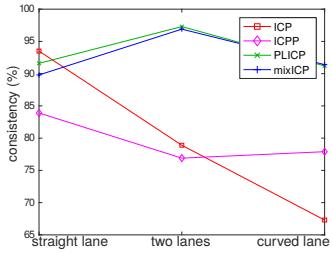
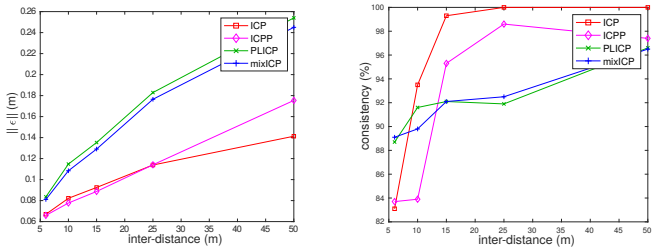


Fig. 7: *Consistency for the four matchings and for the three scenarios: the straight lane, the two lanes and the curved lane drivings.*



(a) *Norm of the position error* $\|\bar{\epsilon}\|$. (b) *Consistency.*

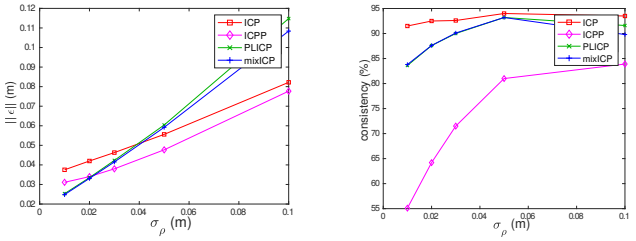
Fig. 8: *Mean of the norm of the relative position error $\|\epsilon\|$ and consistency for the four matchings depending on the inter-distance.*

F. LiDAR noise dependency

The previous results are tested with a range noise on the LiDAR points with a standard deviation of 10cm. We test in this section the error and the consistency when the LiDAR becomes more accurate. The error and the consistency increase when the LiDAR noise increases. (Fig. 9).

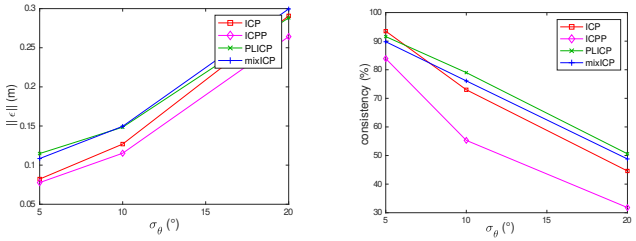
G. Noise on the poses of the leader and the follower

The standard deviations $[\sigma_x\, \sigma_y\, \sigma_\theta] = [0.5m\, 0.5m\, 5°]$ are applied on the poses of the follower and the leader vehicles on the previous results. We test here the effect of less accurate poses on the error and the consistency. Even if the initial localization corrects greatly the position error, the iterative minimization is very sensitive to the orientation



(a) $\|\bar{\epsilon}\|$ (b) *Consistency.*

Fig. 9: *Mean of the norm of the relative position error $\|\epsilon\|$ and consistency for the four matchings depending on the LiDAR range noise with a standard deviation σ_ρ.*



(a) $\|\bar{\epsilon}\|$. (b) *Consistency.*

Fig. 10: *Mean of the norm of the relative $\|\epsilon\|$ and consistency for the four matchings depending on the noise apply to the poses of the follower and the leader vehicles with the standard deviations:* $[\sigma_x\, \sigma_y\, \sigma_\theta] = [0.5\, 0.5\, 5]$, $[\sigma_x\, \sigma_y\, \sigma_\theta] = [2.5\, 2.5\, 10]$ *and* $[\sigma_x\, \sigma_y\, \sigma_\theta] = [5\, 5\, 20]$ *($[\sigma_x] = [\sigma_y] = m$ and $[\sigma_\theta] = °$).*

noise. When it increases, the error increases (Fig. 10a) and the consistency decreases drastically (Fig. 10b). If the relative orientation error becomes very large (near 45°) some ambiguity can appear and the LiDAR points can match on the wrong side of the vehicle.

In this simulation, only one other vehicle was present. If two or more vehicles are present the position error can creates some ambiguity when the algorithm has to choose which points matches with which vehicles.

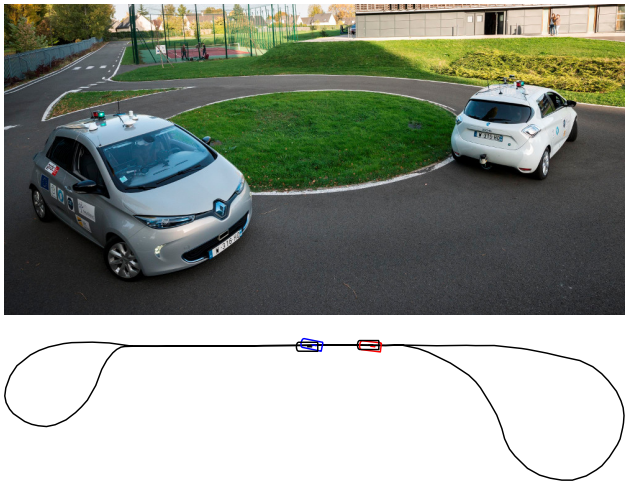


Fig. 11: *Real platooning scenario on the test track Seville, Compiègne, France, in black. The ground truth of the two vehicles are shown in black with their models. The pose and the model are in red for the leader and in blue for the follower.*

TABLE III: *Real platooning scenario errors $\|\epsilon\|$ and consistency for the four matchings, for second minimization method.*

	ICP	ICPP	PLICP	mixICP
$\|\epsilon\|$ (cm)	8.5	5.3	7.1	6.8
consistency (%)	75.1	68.6	92.5	91.8

H. Real platooning scenario

The relative localization have been tested on a real platooning scenario. Two vehicles of the laboratory were driving together on a test track (Fig. 11). This track has two roundabouts and one straight lane between them.

Both vehicles were equipped with an IMU (Span CPT) with a GNSS receiver using RTK corrections for the ground truth. In practice, we have noticed that this ground truth was not accurate enough for relative localization compared to the high quality of the LiDAR measurements. Therefore, the LiDAR of the follower was not used but simulated to correspond perfectly with the pose given by the GNSS receiver. The poses of the follower and of the leader used for the relative localization algorithms were the ground truth with Gaussian noise such as : $[\sigma_x\, \sigma_y\, \sigma_\theta] = [0.5m\, 0.5m\, 5°]$. The LiDAR was simulated with a Gaussian range noise with a 10cm standard deviation.

In this scenario, the inter-distance was evolving between 6m and 16m. The vehicles were following each other on the curved road of the roundabout and on the straight lane.

The consistency and the accuracy are similar to the other results : 92.5% of consistency and 71 mm of error for the point-to-line matching.

V. Conclusion

This work has presented different relative localization methods based on LiDAR points. An estimated pose received from the detected vehicle is used for initialization. A first localization is computed using the bounding boxes of the LiDAR points and of the communicated model. This is used to reduce the position error, the orientation error being unchanged in this stage. An iterative minimization algorithm is then applied using this first localization. We have presented and compared two minimization methods and four different points to polygonal model matchings. First, we have noticed on different scenarios that the second minimization method using the pseudo-inverse matrix gives a better accuracy and consistency. Secondly, a point-to-line matching allows a better estimation of the covariance matrix. This matching gives more freedom to the LiDAR points which can slide along the model. Moreover, this matching needs less iterations to converge. We have also observed that, when two sides of a vehicle are in the field of view of the LiDAR, the problem is better conditioned and the accuracy is higher.

In future work, we will use these algorithms to compute the absolute pose of the follower using the estimated pose of the leader like a deported GNSS antenna. We will test these algorithms with experimental data with two vehicles and more.

Acknowledgment

This work was carried out in the framework of Equipex ROBOTEX (ANR-10- EQPX-44-01) and Labex MS2T (ANR-11-IDEX-0004-02). It was also carried out within SIVALab, a shared laboratory between Renault, CNRS and UTC.

References

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

[2] A. Censi. An accurate closed-form estimate of ICP's covariance. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3167–3172, April 2007.

[3] A. Censi. An ICP variant using a point-to-line metric. In *2008 IEEE International Conference on Robotics and Automation*, pages 19–25, May 2008.

[4] T. Chateau, F. Chabot, C. Teulière, M. Chaouch, and J. Rabarisoa. Deep MANTA: A Coarse-to-Fine Many-Task Network for Joint 2d and 3d Vehicle Analysis from Monocular Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, United States, 2017.

[5] ETSI. Intelligent transport systems (ITS); Vehicular communications; Basic set of applications; Part 2: specification of cooperative awareness basic service. Technical Report ETSI EN 302 637-2 v1.3.1, Sep. 2014.

[6] K.L. Low. 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, February 2004.

[7] A. Petrovskaya and S. Thrun. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26(2-3):123–139, April 2009.

[8] S. M. Prakhya, L. Bingbing, Y. Rui, and W. Lin. A closed-form estimate of 3d ICP covariance. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 526–529, May 2015.

[9] A. Shetty. *GPS-LiDAR Sensor Fusion Aided by 3D City Models for UAVs*. M. S. Thesis, University of Illinois, Urbana-Champaign, 2017.

[10] X. Zhang, W. Xu, C. Dong, and J. M. Dolan. Efficient L-shape fitting for vehicle detection using laser scanners. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 54–59, June 2017.