

# Real Time Lidar and Radar High-Level Fusion for Obstacle Detection and Tracking with Evaluation on a Ground Truth

Hatem Hajri, Mohamed-Cherif Rahal

**Abstract**—Both Lidars and Radars are sensors for obstacle detection. While Lidars are very accurate on obstacles positions and less accurate on their velocities, Radars are more precise on obstacles velocities and less precise on their positions. Sensor fusion between Lidar and Radar aims at improving obstacle detection using advantages of the two sensors. The present paper proposes a real-time Lidar/Radar data fusion algorithm for obstacle detection and tracking based on the global nearest neighbour standard filter (GNN). This algorithm is implemented and embedded in an automotive vehicle as a component generated by a real-time multisensor software. The benefits of data fusion comparing with the use of a single sensor are illustrated through several tracking scenarios (on a highway and on a bend) and using real-time kinematic sensors mounted on the ego and tracked vehicles as a ground truth.

**Keywords**—Ground truth, Hungarian algorithm, lidar Radar data fusion, global nearest neighbor filter.

## I. INTRODUCTION

**D**ATA fusion [1]–[3] is the process of combining multiple data sources in order to provide a better decision. Today data fusion is used in various fields such as military, medical and mobile robotics. In the field of autonomous driving, data fusion has multiple benefits. In fact autonomous vehicles are often equipped with different sensors through which they communicate with the external world. A multisensor fusion takes advantages of each sensor and provides more robust and time-continuous informations than sensors used separately.

**State of the art Lidars, Radars and Cameras fusion.** Several earlier works considered benefits of combining sensors such as Lidars, Radars and Cameras. For example, [4] presents a Lidar-Radar fusion algorithm based on Kalman filter and shows how fusion improves interpretation of road situations and reduces false alarms. Subsequently [5] uses Cramer-Rao lower bound to estimate performance of data fusion algorithms. The paper [6] considers Lidar-Radar fusion with applications to following cars on highways. In order to test the performance of their fusion algorithm, authors of [6] provide a study of mean square errors of relative distances and velocities in a highway tracking scenario using least squares polynomial approximation of

sensors data as a ground truth. Another Lidar, Radar and Camera fusion approach based on evidence theory appears in [7] with applications to the classification and tracking of moving objects. More recently, [8] focuses on fusion of multiple cameras and Lidars and presents tests on real world highway data to verify the effectiveness of the proposed approach.

**Contribution of the paper.** The present paper is concerned with data fusion between Lidar and Radar. In view of the state of the art, we can distinguish two different general fusion methods which were applied for Lidar and Radar: Kalman filter and evidence theory. We believe that approaches which apply one of these methods agree on the main steps. On the other hand, despite the previously mentioned works, the literature still clearly lacks a quantitative comparison between these sensors outputs such as relative coordinates, velocities, accelerations etc and their fusion result in the presence of a ground truth at least on one obstacle. Because of the lack of a ground truth, authors of these papers were led to work with simulated data or manually manage real data in order to create a ground truth and evaluate results.

The aim of this paper is twofold. First it proposes a real-time high-level fusion algorithm between Lidar and Radar based on the GNN filter which in turn is based on Kalman filter. This algorithm is presented with several mathematical and implementation details which go along with it. Second the paper proposes a ground truth generation method to evaluate the performance of the proposed algorithm while focusing on the main outputs of Lidar and Radar which are relative coordinates and speeds of obstacles. This method can be applied regardless scenarios types (highways, bends etc) and weather conditions. Our ground truth is generated using two synchronised autonomous cars equipped with real-time kinematic (RTK) sensors. Beyond data fusion, our method can be applied to test performance of sensors such as Lidars, Radars and Cameras.

Experiments are carried out on two prototypes of the autonomous vehicle of VEDECOM used as ego and obstacle vehicles. Fig. 1 displays the sensor architecture of this prototype. It has five Lidars ibeo LUX with horizontal fields of view of  $110^\circ$  mounted such that they provide a complete view around the car. These Lidars send measurements to the central computation unit (Fusion Box) which performs

Hatem Hajri and Mohamed-Cherif Rahal are with the Institut VEDECOM, 77, rue des Chantiers 78000 Versailles, France (e-mail: hatem.hajri@vedecom.fr, mohamed-cherif.rahal@vedecom.fr).

fusion of measured features, object detections and tracks at the frequency of 25 Hz. A long-range Radar ARS 308 of frequency 15 Hz is mounted at the front of the car with a horizontal view of  $-28^\circ, +28^\circ$ . The vehicle is moreover equipped with a RTK sensor with precisions 0.02 m on position and 0.02 m/s on velocity and a CAN bus which delivers odometry informations.

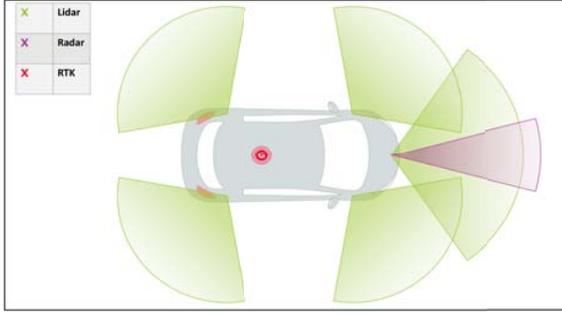


Fig. 1 Sensor configuration: Five lidars, one radar and one RTK sensor

The fusion algorithm uses informations from Lidar/Radar and the CAN bus. RTK sensors will be used for performance evaluation.

**Organisation of the paper.** The content of this paper is as follows. Section II develops the fusion method used in the paper. Section III gives more details on the implementation and integration into vehicle of the proposed algorithm. Section IV presents our ground truth generation method and the experiments (car followings on highways/bends) carried out using the two vehicles to collect data. The obtained ground truth data set will be published in a future paper with a more careful study and sensitivity analysis of the generation method. This ground truth is used to evaluate the mean square errors of Radars/Lidars and fusion measurements of relative positions and velocities of the obstacle vehicle. In addition several plots of temporal evolutions of these measurements are given showing interesting informations about their smoothness and unavailability periods. Results show advantages of data fusion comparing with one sensor.

## II. KALMAN FILTER FOR DATA FUSION

Consider a target (or state) which moves linearly in discrete time according to the dynamic  $x(k) = F(k)x(k-1) + v(k)$  and an observation  $z$  of it by a sensor  $S$  which takes the form  $z(k) = H(k)x(k) + w(k)$ . Assume  $v(k), k = 1, \dots$  and  $w(k), k = 1, \dots$  are two independent centered Gaussian noises with covariances  $P_S$  and  $N_S$  respectively. Kalman filter is known to give an estimation of  $x(k)$  when only  $z(1), \dots, z(k)$  are observed. From a mathematical point of view, the problem amounts to calculating the conditional mean  $\mathbb{E}[x(k)|Z^k]$  where  $Z^k = (z(1), \dots, z(k))$ . Introduce the notations  $\hat{x}(i|j) = \mathbb{E}[x(i)|Z^j]$  which is the conditional mean of  $x(i)$  knowing  $Z^j$  and its conditional covariance

$$P(i|j) = \mathbb{E}[(x(i) - \hat{x}(i|j))(x(i) - \hat{x}(i|j))^T | Z^j].$$

Kalman filter has an explicit solution which is determined recursively. Assume  $x(0)$  is a Gaussian distribution with mean  $\hat{x}(0|0)$  and covariance  $P(0|0)$ . Knowing the estimation  $\hat{x}(k-1|k-1), \hat{x}(k|k)$  is calculated following these two steps. (A) PREDICTION STEP. Compute  $\hat{x}(k|k-1)$  and  $P(k|k-1)$  by:

$$\begin{aligned}\hat{x}(k|k-1) &= F(k)\hat{x}(k-1|k-1) \\ P(k|k-1) &= F(k)P(k-1|k-1)F^T(k) + P_S\end{aligned}$$

This step requires the knowledge of  $z(1), \dots, z(k-1)$ .

(B) UPDATE STEP. When  $z(k)$  becomes available the final solution is obtained as follows

$$\begin{aligned}\hat{x}(k|k) &= \hat{x}(k|k-1) + W(k)(z(k) - H(k)\hat{x}(k|k-1)) \\ P(k|k) &= P(k|k-1) - W(k)S(k)W^T(k)\end{aligned}$$

with

$$\begin{aligned}W(k) &= P(k|k-1)H^T(k)S^{-1}(k) \quad (\text{Kalman gain}) \\ S(k) &= N_S + H(k)P(k|k-1)H^T(k)\end{aligned}$$

The matrix  $S(k)$  is the covariance of the innovation  $\nu(k) = z(k) - H(k)\hat{x}(k|k-1) = z(k) - \hat{z}(k|k-1)$ . The innovation measures the deviation between the estimates provided by the filter and the true observations. Its practical interest lies in the fact that (under the Gaussian assumption) the normalised innovation  $q(k) = \nu^T(k)S(k)^{-1}\nu(k)$  is a  $\chi^2$  distribution with  $\dim(z(k))$  degrees of freedom (see [3], [9]). As a consequence, with a high probability  $\alpha$ , the observation  $z(k)$  associated with the object  $x(k)$  belongs to the area

$$\{z : d^2 = (z - \hat{z}(k|k-1))^T S(k)^{-1}(z - \hat{z}(k|k-1)) \leq \gamma\}$$

where  $\alpha$  is such that  $\mathbb{P}(\chi^2 < \gamma) = \alpha$ . This area is known as the validation gate. When several measurements  $z$  are available for one specific object, the  $\chi^2$  distribution makes it possible to identify those measurements which may correspond to the underlying object by applying a hypothesis test as follows: If there are observations in the validation gate choose  $z$  which minimizes  $d^2$  as the only valid observation. Otherwise no observation is associated. This solution is known as the nearest neighbour filter. The solution which consists in averaging all observations which fall in the validation gate is known as the probabilistic data association filter (see [3] for more background on probabilistic filters). In practice, which is the case for Radar/Lidar, several tracked objects and new measurements can be available simultaneously. The task of associating new measurements with the underlying observed objects is known as the data association problem. This problem is dealt with in Section III by first solving a global optimization problem and second updating the list of objects using the nearest neighbour filter. The obtained filter is known as the GNN filter.

In the present paper each obstacle (old state or new observation) is represented as a vector of its relative coordinates and velocities  $[x \ y \ v_x \ v_y]^T$ . It is assumed that obstacles move according to the constant velocity motion model: if  $\Delta$  denotes the elapsed time between the  $k$  and  $k+1$

sensor data emissions, the matrices  $F$  and  $H$  describing the state and observation evolutions are given by

$$F(k) = \begin{pmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad H(k) = I_4$$

The content of this paper can be adapted to other motion models such as the constant acceleration motion model. We choose the constant velocity model for two reasons. First Lidars do not send informations about accelerations of obstacles. Second the constant velocity model requires less unknown parameters to estimate than the constant acceleration model.

### III. REAL TIME IMPLEMENTATION

This section gives details of the implementation of the real-time fusion module between the two sensors Lidar ibeo LUX and Radar ARS 308. This module was first implemented in Matlab/Simulink and then embedded in the automotive vehicle as a component generated by the software RTMaps (Real Time, Multisensor applications) [10]. The RTMaps software is widely used for real-time applications in mobile robotics as it allows to synchronise, record and replay data from different sensors. The proposed fusion module is an iterative algorithm which is run as soon as a sensor data arrives. Simulink delay functions offer a tool for this kind of problems as they allow to store the last value of a module. The following Fig. 2 shows the Simulink diagram used to generate the fusion module.

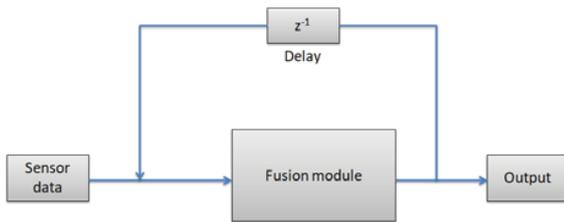


Fig. 2 The Simulink diagram

After a data by a sensor  $S_k$  (Lidar or Radar) is received at time  $t_k$ , this algorithm outputs a fused set of obstacles  $FO_k$ . The main characteristics of each obstacle (other outputs such as identity, age and so on will not be detailed) are 1) a vector  $[x, y, v_x, v_y]^T$  where  $x, y$  (resp.  $v_x, v_y$ ) are the relative (with respect to the ego-vehicle frame) coordinates (resp. velocities) of the object and 2) an uncertainty  $4 \times 4$  matrix of the object. The input of the algorithm at time  $t_k$  is the previous fused set of obstacles  $FO_{k-1}$ , the linear and angular velocities of the ego-vehicle at time  $t_k$  and the sensor data consisting in relative coordinates  $x, y$  and speeds  $v_x, v_y$  of the detected obstacles all at time  $t_k$ . The angular and linear velocities are available on the CAN bus. Since this sensor does not generate data at the same moments as Lidar or Radar, in practice an approximation value of these quantities at time  $t_k$  are taken (for example the last known ones before  $t_k$ ).

The fused list is initialized when the first sensor data arrives. For each object in this list, the vector  $[x, y, v_x, v_y]^T$  of relative coordinates and velocities given by the sensor is stored. The uncertainty of each object is initialized to the corresponding sensor's uncertainty  $N_S$ .

Assume known the last fused list of objects  $FO_{k-1}$  at time  $t_{k-1}$  updated after reception of data by the sensor  $S_{k-1}$  and assume a new sensor data arrives at time  $t_k$  by the sensor  $S_k$ . The update  $FO_k$  of  $FO_{k-1}$  follows these steps.

**(a) Tracking.** Call  $\mathcal{R}_{k-1}$  and  $\mathcal{R}_k$  the ego-vehicle frames at times  $t_{k-1}$  and  $t_k$ . Each object  $o^{k-1} = [x^{k-1}, y^{k-1}, v_x^{k-1}, v_y^{k-1}]^T \in FO_{k-1}$  with uncertainty  $I^{k-1}$  is first tracked in the frame  $\mathcal{R}_{k-1}$  under the constant velocity hypothesis. Then it is mapped to the new frame  $\mathcal{R}_k$  by a rotation of its relative coordinates and relative velocities. Call  $\omega$  and  $v$  the instantaneous angular and linear velocities of the ego-vehicle at time  $t_k$  and define the following estimates of the cap angle and travelled distance  $\theta = \omega \times \Delta, d = v \times \Delta$  where  $\Delta = t_k - t_{k-1}$ . Call  $o^{k-1,t} = [x^{k-1,t}, y^{k-1,t}, v_x^{k-1,t}, v_y^{k-1,t}]^T$  the new object and  $I^{k-1,t}$  its uncertainty in  $\mathcal{R}_k$ . More explicitly, the following identities hold:  $[x^{k-1,t}, y^{k-1,t}, v_x^{k-1,t}, v_y^{k-1,t}]^T$

$$= \begin{pmatrix} R_\theta & 0 \\ 0 & R_\theta \end{pmatrix} \begin{pmatrix} x^{k-1} + \Delta v_x^{k-1} - d \cos(\theta) \\ y^{k-1} + \Delta v_y^{k-1} - d \sin(\theta) \\ v_x^{k-1} \\ v_y^{k-1} \end{pmatrix}$$

and

$$I^{k-1,t} = \begin{pmatrix} R_\theta & 0 \\ 0 & R_\theta \end{pmatrix} (F I^{k-1} F^T + P_{S_k}) \begin{pmatrix} R_\theta & 0 \\ 0 & R_\theta \end{pmatrix}^T$$

where  $R_\theta$  is the rotation matrix of angle  $\theta$  and  $F$  is given in the previous section.

**(b) Association.** In this setp, each new sensor measurement is associated with at most one tracked object that corresponds best to it. For this, the classical Mahalanobis distance will be used as a similarity measure. This distance is defined for any tracked object  $o^{k-1,t}$  with uncertainty  $I^{k-1,t}$  in  $\mathcal{R}_k$  and new observation  $z^i$  by

$$d_{k,i}^2 = (o^{k-1,t} - z^i)^T (I^{k-1,t})^{-1} (o^{k-1,t} - z^i).$$

The association problem can be reformulated in the following optimization form: find the set  $(c_{k,i})$  which minimizes  $\sum_{k,i} c_{k,i} d_{k,i}^2$  subject to the constraints  $\sum_k c_{k,i} = 1$  for each  $i$ ,  $\sum_i c_{k,i} = 1$  for each  $k$  and  $c_{k,i} \in \{0,1\}$  for all  $k$  and  $i$ . A well known solution to this problem is given by the Hungarian algorithm [11]. This algorithm was implemented in Matlab/Simulink and subsequently used in the fusion module. Notice if there are less tracked objects than new observations, a tracked object is associated with exactly one observation and if not it may or may not be associated.

**(c) List update.** The fused list of objects is updated as follows. First for a tracked object  $o^{k-1,t}$  with uncertainty  $I^{k-1,t}$  which is associated with an observation  $z^i$ , the latter is accepted as an observation of  $o^{k-1,t}$  if it falls in the validation gate around  $o^{k-1,t}$  that is if  $d_{k,i}^2 < \gamma$  with  $\gamma$

a quantile of order 0.9 of the  $\chi^2$  distribution with 4 degrees of freedom. In this case both  $o^{k-1,t}$  and  $I^{k-1,t}$  are updated as in Kalman correction step based on the observation  $z^i$ . If the object is not associated with an observation, it is removed from the list. Another possibility is to continue tracking absent obstacles for a while. However we did not choose this option since Lidars and Radars already track absent obstacles. Finally new observations that are not associated with tracked objects are added to the list as for the first fused list. To summarize, the new list  $FO_k$  is composed of tracked objects associated with new observations and updated by Kalman filter and new observations not associated with tracked objects.

#### IV. EXPERIMENTS AND RESULTS

The last section is devoted to the generation of ground truth for the evaluation of the proposed data fusion algorithm (denoted Fusion in brief) and for comparisons between Lidar, Radar and Fusion. The focus will be on relative positions/velocities  $x, y, vx, vy$  of the target vehicle.

**Ground truth generation.** To generate the ground truth, we used two synchronised autonomous vehicles. The ego vehicle is equipped with Lidar and Radar and both vehicles are equipped with RTK sensors. The idea behind this generation process is that given two moving points  $A$  and  $B$  represented by their positions and velocities in a global frame  $R$  and knowing the angular velocity of  $A$ , it is possible to deduce the positions and velocities of  $B$  in the moving frame of  $A$  (composition of movements formula). In practice, these informations are used:

- (1) global coordinates and velocities in the reference frame  $\mathcal{R}_0$  of RTK sensors of the two vehicles during experiments all obtained from the RTK sensors.
- (2) heading of the ego vehicle in  $\mathcal{R}_0$  during experiments obtained from the RTK sensor mounted on this vehicle.

These informations in combination with the composition of movements formula provide ultra-precise estimations of the relative positions/velocities of the target vehicle in the ego vehicle frame. In fact, at each time  $t$ , informations (1) give the relative coordinates and speeds with the ego-vehicle identified to a point (that is without consideration of its heading). A rotation of angle given by the heading at time  $t$  (obtained from (2)) gives the desired estimations. In order to perform comparison, one has to find the target car characteristics  $(x_t, y_t, vx_t, vy_t)$  sent by Lidar/Radar/Fusion at a given time  $t$ . For this, the nearest obstacle to the ground truth position at time  $t$  which is non static was considered as the target car viewed by Lidar/Radar/Fusion. This method gives the desired obstacle most all the time. Since sensors have different frequencies, linear interpolation was used to get an estimate of any quantity (position/velocity) which is not available at a given time.

**Evaluation of the fusion algorithm.** In order to estimate the uncertainty matrices  $P_S$  and  $N_S$  for both Lidar and Radar involved in the algorithm, we used a ground truth collected during 10 minutes. We set all off-diagonal entries of these matrices to 0 and tolerate more error than the obtained

values. To evaluate the fusion algorithm, we generated new data in two challenging circuits: car followings on a highway and a highly curved bend. In each case, the same car-following scenario is repeated seven times. For each case, 4 figures displaying variations of the relative coordinates and velocities  $x, y, vx, vy$  in one scenario are shown along with the ground truth. We choose one scenario among seven because of space limitations. The two plots at the bottom (green and black referring to Radar and RTK) are the true ones (without translation). For better visualization, those at the middle (red and black referring to Lidar and RTK) and the top (blue and black referring to Fusion and RTK) correspond to the true ones + offset and true ones + 2\*offset. The offset is specified with each figure's title. Radar/Lidar/Fusion points are represented by small squares joined by lines with the same colors and RTK points are represented by continuous curves (linear interpolation of the values). In addition, two tables representing the mean square errors (MSE) of  $x, y, vx, vy$  for Radar/Lidar/Fusion are given for all the seven car following experiences. These errors are calculated with respect to the RTK output considered as a ground truth according to the formula

$$\text{MSE on } q = \frac{1}{N} \sum_{t=1}^N (q\text{Sensor}_t - q\text{RTK}_t)^2$$

In this formula, the notation  $q\text{Sensor}$  with  $q \in \{x, y, vx, vy\}$  and  $\text{sensor} \in \{\text{Radar, Lidar, RTK, Fusion}\}$  refers to  $q$  of the target as seen by sensor,  $q\text{Sensor}_t$  is the value of  $q\text{Sensor}$  at time  $t$  and  $N$  is the number of samples.

**Runtime performance.** Multiple tests show that the fusion algorithm is able to treat 50 obstacles in less than 15 microseconds. This period is negligible compared to periods of Lidar and Radar making the algorithm suitable for fusion in real time.

##### A. Car-Following on a Highway

The first scenario is a tracking, shown in Fig. 3, of the target vehicle on a highway at high speed (between 90 and 100 km/h). We repeated the same experience on the same portion of the highway seven times and obtained a record of more than 6 minutes.



Fig. 3 Tracking on a highway

The next figures display respectively the variations of  $x, y$  (in metre)  $vx, vy$  (in metre/second) as a function of time (in

second) of the target vehicle by Radar/Lidar/Fusion and RTK in one experience among the seven.

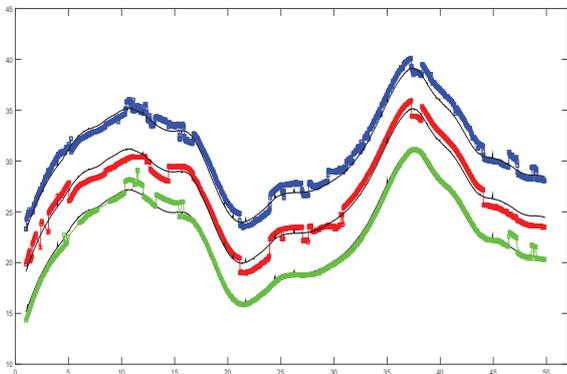


Fig. 4 Variations of  $x$  (in m) with Radar (green)/Lidar (red)/Fusion (blue) in comparison with RTK (black) as a function of time (in s). Offset=4

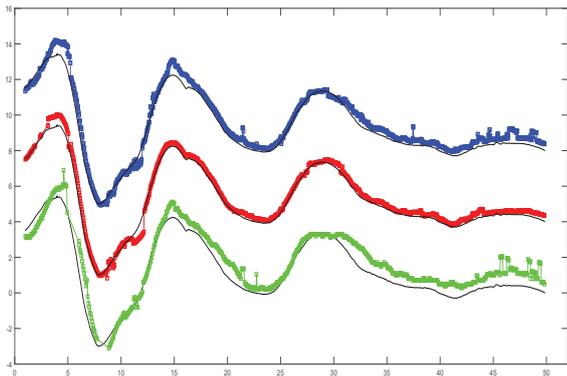


Fig. 5 Variations of  $y$  with Radar/Lidar/Fusion. Offset=4

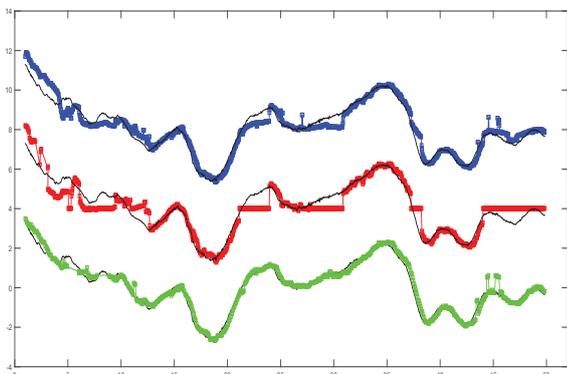


Fig. 6 Variations of  $v_x$  with Radar/Lidar/Fusion. Offset=4

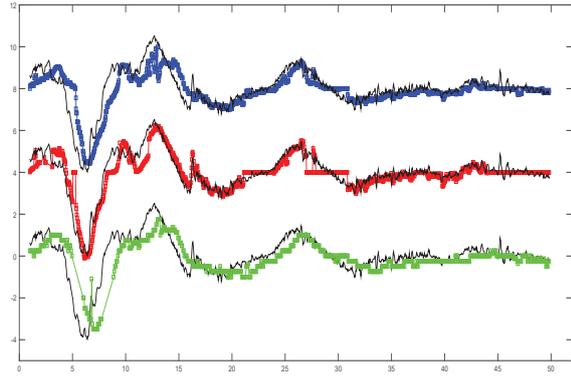


Fig. 7 Variations of  $v_y$  with Radar/Lidar/Fusion. Offset=4

Table I shows the mean square errors (MSE) on  $x, y, v_x, v_y$  by Radar, Lidar, Fusion corresponding to the seven car-following experiences.

TABLE I  
MSES OF  $x, y, v_x, v_y$  BY /RADAR/LIDAR/FUSION

	Radar	Lidar	Fusion
MSE on $x$	0.33	0.55	0.39
MSE on $y$	0.43	0.16	0.21
MSE on $v_x$	0.15	0.28	0.19
MSE on $v_y$	0.25	0.31	0.27

### B. Car-Following on a Bend

The second scenario is a tracking, shown in Fig. 8, on a succession of highly cruved bends. We repeated the same experience seven times getting a record of more than 6 minutes.



Fig. 8 Tracking on a bend

Fig. 9 shows the trajectory in the world frame of the ego vehicle obtained from the RTK sensor.

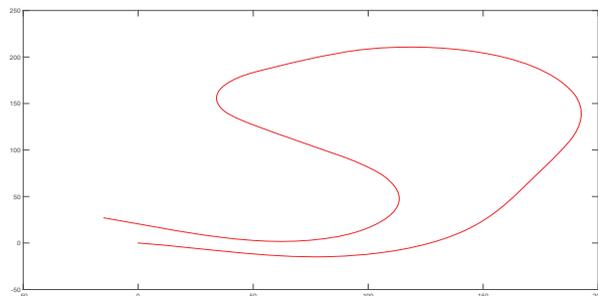


Fig. 9 Trajectory of the ego vehicle. The starting point is (0, 0)

Variations of the relative  $x, y$  (in metre)  $vx, vy$  (in metre/second) as a function of time (second) in one experience among the seven are displayed in the next 4 figures.

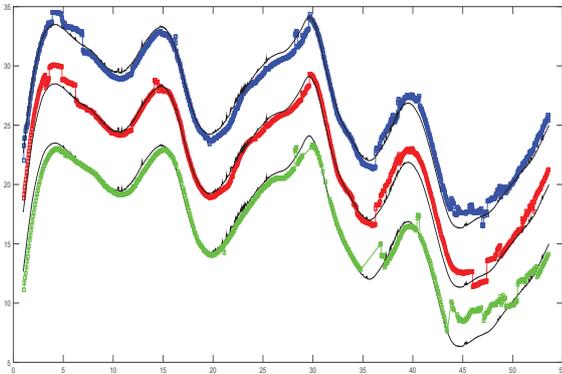


Fig. 10 Variations of  $x$  (in m) with Radar (green)/Lidar (red)/Fusion (blue) in comparison with RTK (black) as a function of time (in s). Offset=5

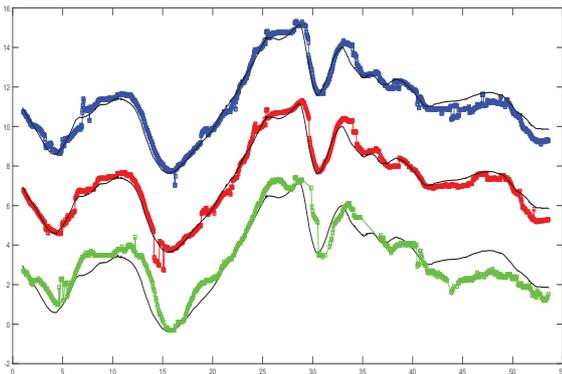


Fig. 11 Variations of  $y$  with Radar/Lidar/Fusion. Offset=4

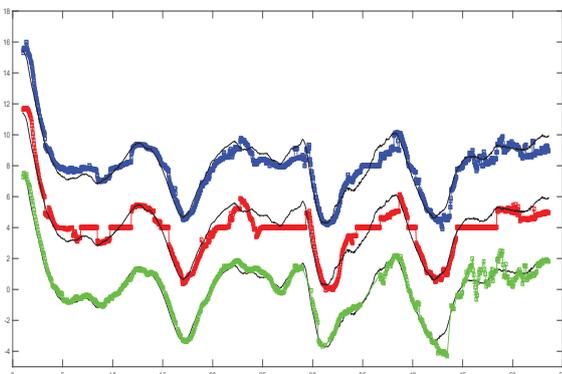


Fig. 12 Variations of  $vx$  with Radar/Lidar/Fusion. Offset=4

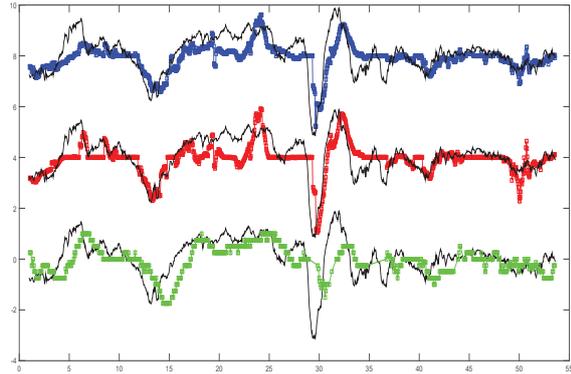


Fig. 13 Variations of  $vy$  with Radar/Lidar/Fusion. Offset=4

Table II shows the mean square errors (MSE) on  $x, y, vx, vy$  by Radar, Lidar, Fusion corresponding to the seven experiences.

TABLE II  
MSES OF  $x, y, vx, vy$  BY RADAR/LIDAR/FUSION

	Radar	Lidar	Fusion
MSE on $x$	0.8	0.61	0.65
MSE on $y$	0.5	0.15	0.19
MSE on $vx$	0.16	0.45	0.29
MSE on $vy$	0.33	0.38	0.36

### C. Comments

- The plots of  $x$  and  $y$  are relatively smooth for Lidar/Radar/Fusion in both scenarios. In contrast, the plots of  $vx$  and  $vy$  for Lidar present multiple brutal transitions and piecewise constancies. These drawbacks are minimal for Radar and Fusion.
- Multiple non detection periods of the target are observed for Radar especially in the second scenario. This can be explained by the fact that the target is not in the field of view of the sensor or also by a sensor malfunction.
- It is remarkable that Radar was more accurate on  $x$  than Lidar in the first scenario. This fact is supported by the experiment presented in [6] in which Radar was more accurate than Lidar on the relative distance  $= \sqrt{x^2 + y^2}$ .
- Experiments lead to the following conclusion. First, in terms of accuracy, Fusion provides a good compromise value between Radar and Lidar. Second, Fusion is more robust against unavailability of Lidar and Radar. Unavailability of information has dangerous impact in autonomous driving. This problem is very unlikely to occur for Fusion as the latter combines two sensors.

### D. Conclusion and Future Works

This paper presented a real-time fusion algorithm between Lidar and Radar which was implemented and successfully integrated into the autonomous car. This algorithm is based on the GNN filter and outputs multiple characteristics of the detected objects such as their relative positions/ velocities and uncertainties. Performance of Fusion in comparison with Radar and Lidar was evaluated through multiple tracking

scenarios (on a highway and a bend) using two synchronised vehicles and relying on data coming from ultra-precise RTK sensors as a ground truth. Benefits of Fusion were illustrated through two main central ideas: accuracy regarding the ground truth and robustness against sensors malfunctions. In future works, we plan to conduct experiences in challenging conditions and use ground truth to compare our approach with the evidence theory approach [7].

## REFERENCES

- [1] D. L. Hall and S. A. H. McMullen, *Mathematical Techniques in Multisensor Data Fusion*. Artech House, 2004.
- [2] H. B. Mitchell, *Multi-Sensor Data Fusion: An Introduction*, 1st ed. Springer Publishing Company, Incorporated, 2007.
- [3] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, ser. Mathematics in Science and Engineering. Academic Press, 1988, no. 179.
- [4] C. Blanc, B. T. Le, Y. Le, and G. R. Moreira, "Track to track fusion method applied to road obstacle detection," *IEEE International Conference on Information Fusion*, 2004.
- [5] C. Blanc, P. Checchin, S. Gidel, and L. Trassoudaine, "Data fusion performance evaluation for range measurements combine with cartesian ones for road obstacle tracking," *11th IEEE International Conference on Intelligent Transportation Systems*, 2007.
- [6] D. Gohring, M. Wang, M. Schnurmacher, and T. Ganjineh, "Radar/lidar sensor fusion for car-following on highways," *IEEE Int. Conf. Autom. Robot. Appl.*, pp. 407–412, 2011.
- [7] R. O. Chavez-Garcia and O. Aycard, "Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 252–534, 2015.
- [8] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using Cameras and Lidars," <https://arxiv.org/pdf/1802.08755.pdf>, 2018.
- [9] H. Durrant-Whyte, *Introduction to estimation and the Kalman filter*. Australian Centre for Fields Robotics.
- [10] F. Nashashibi, "Rtm@ps: a framework for prototyping automotive multi-sensor applications," pp. 99 – 103, 02 2000.
- [11] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society of Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, March 1957.