Dempster Shafer Grid-based Hybrid Fusion of Virtual Lanes for Autonomous Driving

Ferit Uzer, Rachid Benmokhtar, Salma Moujtahid, Xavier Perrotton

Dempster Shafer Grid-based Hybrid Fusion of Virtual Lanes for Autonomous Driving

Ferit Uzer[1], Rachid Benmokhtar[1], Salma Moujtahid[1] and Xavier Perrotton[1]

***Abstract*— Lane estimation plays a central role in the context of self-driving vehicles, requiring clear road markings and a high quality infrastructure. To meet the safety requirements and manage difficult use cases (*i.e.* poor quality or absence of road-marking, traffic jams, severe weather conditions, *etc.*) without depending on cartography and lane markings, we propose a lidar-based Virtual Lanes (VL) generation system to provide a comfortable and safe ride. For this purpose, hybrid fusion based on Dempster-Shafer Theory (DST) coupled with a particle filter is introduced. Two VL strategies are merged: the first constructs virtual lanes based on independent road-borders detection, while the second uses moving objects trajectories. A novel lane similarity computation is also adopted in order to estimate an effective lane comparison. The performance is reported through extensive experiments with Valeo demo-car on highway and beltway roads. Experimental results demonstrate the accurate and robust performance of the proposed system.**

I. INTRODUCTION

Over the last few years, research in Autonomous Driving (AD) and Advanced Driver Assistance Systems (ADAS) has been very active in order to increase traffic safety. Adaptive Cruise Control (ACC), Lane Centering Assist (LCA) and Lane Keeping Assist (LKA) have been introduced into the market for restricted areas of operation (*e.g.* well structured road geometries).

Environment perception, and more precisely drive-able area detection, is an essential component of safe autonomous driving. Multiple sensors are usually employed such as ultrasonic, lidar, radar, camera, GPS, *etc.* Each sensor type has demonstrated its capacities and strengths for certain environments. Yet due to physical limitations, none of the existing sensor technologies are able to provide complete information on their own in all conditions.

To respond to these challenges, multi-sensor fusion has been widely adopted for object detection [1] or road-marking detection [2]. The redundancy not only insures a larger field of view but also a more reliable and robust unified information, and ultimately a safer detection and prevent a possible failure [3].

However, some scenarios with specific infrastructure events are not completely managed such as absence or poor quality of road-marking, working zones (where markings can be painted over or degraded), traffic jams (where markings are hidden by other vehicles), highway entries, exits and bifurcations. These represent road networks' most challenging areas and detecting them with a look-ahead distance, sufficient to take necessary actions, is therefore crucial for achieving a safer AD.

Virtual Lanes (VL) methods can be used in these environments to counter the absence of visible lanes, instead of employing predictive (yaw rate based) methods. Traditional VL approaches [4] can include :

- Using High-precision GPS and pre-built maps, which has the inconvenience of the price and the necessity of regular map updates.
- Free-space detection around the vehicle to determine minimal risk trajectories. The disadvantage is that it is oriented solely around safety (i.e. for collision avoidance systems).
- *Platooning* (i.e. tracking of the front vehicle). Park et al. [5] uses the position of primary target to generate the ego lane. The disadvantage of this approach is that it requires a vehicle in the host path and leads to changing lanes at the same time as the vehicle followed.

Other VL methods are oriented towards static obstacles (*e.g.* road-borders) or moving vehicle trajectories. Road-border (RB) detection is a well-studied topic and a key information for road modeling with optical sensors [6]. Due to the limits of vision sensors, radars and lidars are also leveraged to determine the course of the road. In the early examples of lidar based road boundary and course estimation, other traffic participants are completely excluded in order to detect the RBs by using raw laser points [7]. Then it is extended by including the measurements of closest vehicles [8]. Radar sensors [9] are also investigated to represent the RB in an occupancy grid map. Meis et al. [10] extended this by using tangential features. Fusing Radar data with vision is also exploited in [11].

In this paper, we address the challenge of VL fusion for a level 2 to 4 of autonomy [12] by exploiting the lidar cocoon. Our Contributions are the following:

- We propose to manage situations without road-marking and introduce redundancy information thanks to two independent VL frameworks : road-border VL and surround objects VL.
- We present a hybrid VL fusion based on DST because of its ability to represent both the accuracy and the reliability of inputs.
- A novel lane similarity computation is proposed in order to estimate global and effective lane parameters.

The rest of the paper is organized as follows: in Section II, the general framework and the road model are presented. VL generation systems and their fusion is detailed in section

[1]Valeo Vision - Driving Assistance Research (DAR), France

III. Experimental results and evaluation are described in section IV. Finally, we conclude with a summary of the most important results and our perspective.

II. GENERAL FRAMEWORK

A. System Architecture Overview

The global architecture of the proposed system is depicted in Fig. 1. The 360° Lidar cocoon provides a list of 3D-scan points, which are divided into stationary and dynamic.

Stationary points serve as input for the estimation of the RBs and VL road-borders are constructed based on that process. Dynamic points feed the multi object tracker which builds partial history trajectories using the ego vehicle odometry. Those trajectories constitute surround objects VL. Finally, DST fusion is employed in order to estimate relevant multi-lane parameters.

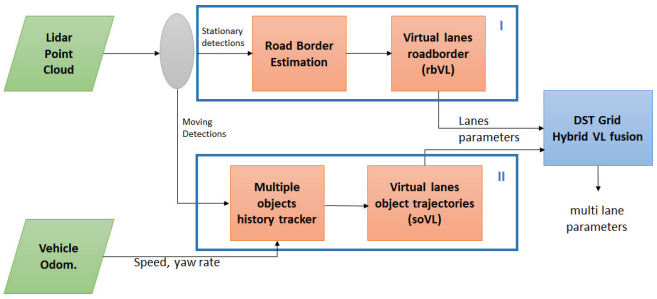

Fig. 1. General framework of the proposed system.

1) Road model: Road structure can be represented as multiple lanes delimited by road marking lines and surrounded by road-borders. Depending on the complexity needed, different geometrical models can be employed. We chose to use a $K = 3$ degree polynomial functions $y(x)$ to model lanes, lines or road-borders:

$$y(x) = c_3x^3 + c_2x^2 + c_1x + c_0, \tag{1}$$

where $c_0[meter]$ is the lateral offset to the ego vehicle, $\arctan(c_1)[radian]$ is the heading angle, $2 \times c_2[1/meter]$ is the curvature and $6 \times c_3[1/meter^2]$ is the curvature derivative. $y(x)$ represents the lateral position of the model.

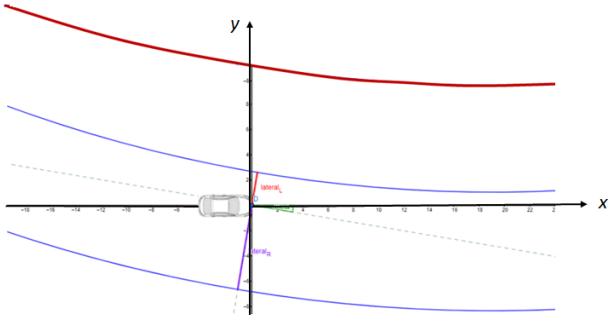

Fig. 2. Lane (in blue) and road-border (in red) representation.

2) Lane similarity function: Given two lane estimations, we need to find a measure of similarity between them. Contrary to the distance between two points or between a point and a curve, the distance between two curves is not straightforward. In [13], authors compare existing lane evaluation approaches for realistic (with human driven trajectory) and synthetic data considering different geometrical models. The most common strategies are based on the distance minimization of lateral offset with Velodyne lidar point alignment, measuring the angle deviation or GPS positioning. However, these methods need extra sensor configurations (*i.e.* an appropriate scaling regarding to vehicle coordinate system) and do not provide enough precision.

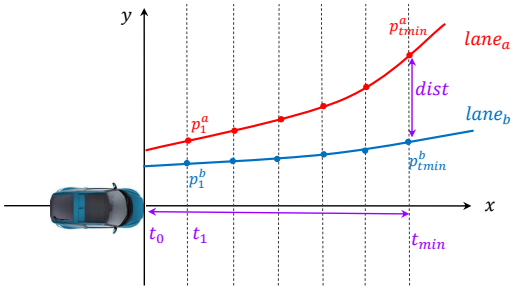

Fig. 3. The similarity function is applied to three trajectory examples with $\tau = 0.5m$ for safety goals.

Based the traffic safety regulations and intuition of what similarity between lanes could represent, we propose a metric which does not require tuning of multiple parameters. The center model for each lane estimation is taken for similarity comparison. To compare two lane models $lane_a$ and $lane_b$, they are defined as N time-stamped points $\{(t_1, p_1), (t_2, p_2) \ldots, (t_n, p_n)\}^{lane}$ as illustrated in Fig. 3. Then at a safety look-ahead time t_{min}, the distance $dist(lane_a, lane_b)$ is computed between the points $p^a_{t_{min}}$ and $p^b_{t_{min}}$ from each lane model such that:

$$dist(lane_a, lane_b) = \sqrt{(x^a_{t_{min}} - x^b_{t_{min}})^2 + (y^a_{t_{min}} - y^b_{t_{min}})^2}$$

$$sim(lane_a, lane_b) = \begin{cases} 1 & \text{if } dist \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

This similarity measure $sim()$ between lanes has the advantage of considering the lateral offset, the curvature as well as the heading angle in one distance value.

B. Virtual-Lanes Road-border (rbVL)

The rbVL framework is built based on the road-border detection algorithm. Its input is the 3D points cloud of lidar cocoon. The process is presented as follow:

- Scanning points are projected on ego-vehicle coordinates frame preserving their layer information and scanning order.

- Points from dynamic objects are filtered. The rest of the points are considered as static (*i.e.* coming from road surface and the infrastructure RB, traffic signs, *etc.*).
- Points from the road surface are eliminated by height-based filtering.
- A lateral histogram is computed to remove sparse areas in point cloud, giving better detection accuracy.
- The remaining points are clustered by *Mean-shift* [14] which iteratively seeks centroid of data in the kernel until convergence and clustering them considering similar lateral positions in our case (3).

$$\mathcal{K}_{l+1} = \frac{\sum_{i=1}^{n} x_i(\| \frac{\mathcal{K}_l - p_i}{h} \|^2)}{\sum_{i=1}^{n} \mathcal{G}(\| \frac{\mathcal{K}_l - p_i}{h} \|^2)} \tag{3}$$

$\mathcal{G}$ is the kernel, p is the data, $\mathcal{K}$ is the kernel location and h is the kernel half width. Clusters are estimated by adapting *mean-shift* segmentation to the histogram data. However in curves, clusters are over segmented due to the large curvature. Therefore, it is followed by a cluster merging process which checks the main directions of change in scanning direction of each point so that the wrongly classified points are corrected.

After the perception and clustering steps, the RB are mathematically represented by (1) where each side is modeled independently. In other words, there is no assumption that two sides are parallel to each other as it is shown in Fig. 5.

The polynomial fitting of RB models is based on a RANSAC algorithm. 4 random points are selected separately from each side. After a predefined number of iterations, the model with a maximum number of inliers represents the corresponding RB if it exists.

An Extended Kalman Filter (EKF) for each side is used to track the RB parameters (*i.e.* lateral offset, heading angle, curvature and curvature derivative) of the computed model. Roads/lanes have highly predictable geometry, therefore information about a road/lane estimation in time and space should provide significant information about the road at nearby points. Combining this with the non-linearity of our model and our noisy observations, EKF becomes the best candidate. That is, the requirement of linear equations for the measurement and state-transition models is relaxed; instead, they can be nonlinear and need only to be differentiable.

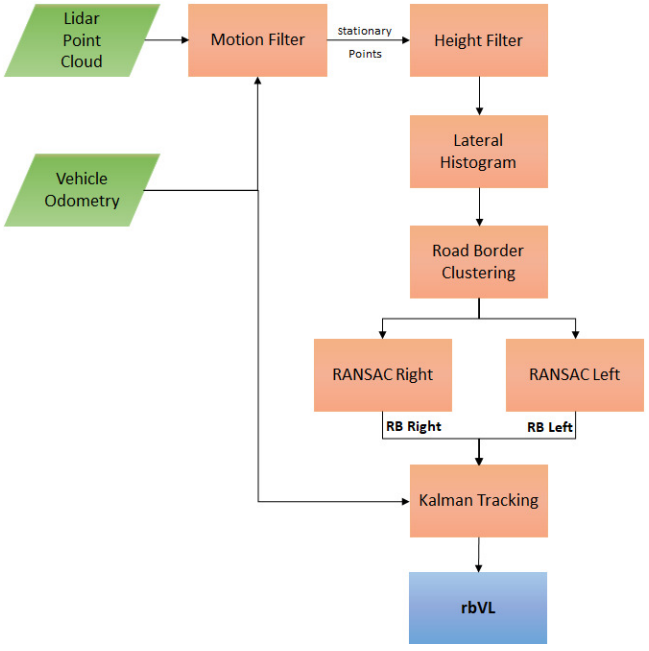

Fig. 4. Road-border detection flowchart.

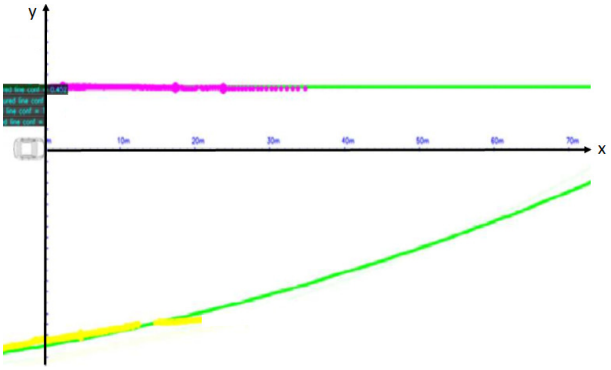

Fig. 5. Independent detection of right and left road-borders. Magenta represent the inlier points for left RB and yellow represents the inlier points for the right RB

Finally, similarity between left and right RBs is determined by the proposed lane similarity function (2) in order to construct rbVL. If $sim = 1$ as a result of (2) with a time range t_{max}, model parameters of rbVL are calculated by taking a weighted average (*i.e.* based on their confidence values) of right and left RB models within the given similarity range. In case of dissimilarity or unavailability of one side, the closest (to the ego vehicle) or the available RB's model is selected for rbVL construction.

C. Surround objects Virtual-Lanes (soVL)

Considering that detected dynamic surround objects are travelling on the same road as the ego-car, they represent important information of the road curvature, heading angle and lateral offset. Hence this information can be used in order to generate soVL. The flowchart of the proposed algorithm is given in Fig.6 and the major steps are the following:

- A trajectory is built for each moving object.
- History is updated using ego vehicle motion information (yaw rate, velocity, time displacement) at each step.
- High importance to the most recent positions is given by a weighting process. Doing that we prevent possible delays in curvature and heading estimations due to the past measurements in history track.
- Polynomial model (1) is fitted with RANSAC.
- A similarity matrix between each trajectory model is calculated. A measure of similarity between each pair is first determined by the equation (2). Then t_{max} which is the maximum time $sim = 1$ is converted to distance by multiplying with current speed of the ego vehicle. Finally, this distance is combined with the RANSAC

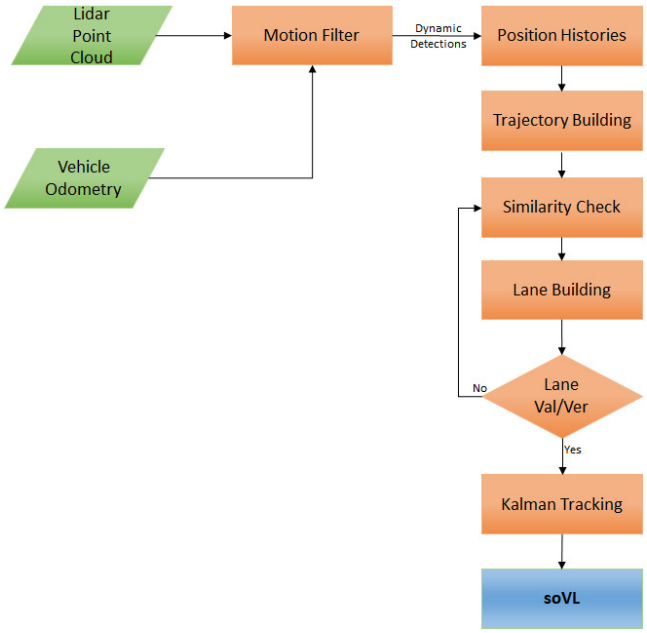

Fig. 6. Flowchart of soVL detection is given.

scores of each member of trajectory pair:

$$\mathbb{S}_{ij} = \frac{(r_i + r_j) * d_{ij}}{d_{max}} \tag{4}$$

where $\mathbb{S}_{ij}$ is the similarity score between the i^{th} and j^{th} trajectories. r_i and r_j are the RANSAC score of the i^{th} and j^{th} trajectories. d_{ij} is the calculated distance and d_{max} is the maximum distance to normalize the score.

- Estimated soVLs are tracked by an EKF.

III. Hybrid Virtual Lanes Fusion

After computing rbVLs and soVLs, they are fused following Dempster-Shafer Theory to output a unified and more complete Virtual Lanes representation of the road.

The lanes fusion system treats the lateral offset, heading angle and curvature of the left and right lines as separate variables to be fused. Belief masses for each sensor are computed and combined using Dempsters conjunctive combination rule.

A. State-of-the-art

There are a number of mathematical theories in literature to represent data imperfection, such as probability theory, fuzzy set theory, possibility theory or Dempster-Shafer evidence theory. Most of these approaches are able to model specific aspects of imperfect data. For example, a probabilistic distribution expresses data uncertainty. Fuzzy set theory can represent vagueness of data and evidential belief theory can represent uncertain as well as ambiguous data.

Bayesian fusion [15] provides a formalism for combining information using conditional probability according to the Bayes rules. It can be difficult estimating the conditional probabilities and it is ineffective in representing data imperfection. On other hand, Fuzzy logic [16], the membership of a set is more gradual rather than just member or not a member. It is guided by the notion of fuzziness, and uses the whole interval of real numbers to develop logic, as a basis for rules of inference. The main advantages of Fuzzy theory are the reasoning model simplicity and the ability to deal with uncertainty and non-linearity. DST [17], [18] provides a global formalism to represent incomplete knowledge, updating beliefs and a combination rule. It is also able to consider conflict between sources. Instead of assigning unknown evidence to prior probabilities, this evidence is assigned to ignorance, giving the fusion more flexibility. For these reasons, we chose to employ DST. For a more detailed review of data fusion theories, refer to [19].

B. Dempster-Shafer Grid-based fusion

The proposed approach is inspired from the evidential fusion approach proposed in [2]. The fusion is performed on the polynomial coefficients $\{c_0, c_1, c_2, c_3\}$ of each line independently. The main steps are :

*1) **Frame of discernment**:* It is defined as the set of intervals of the form $[c_{k_a}, c_{k_b}[$, which is a discretization of the coefficient c_k on a uniform grid.

*2) **Source modelling**:* The input data from different sources need to be represented as belief masses. To do so, the coefficients are first modelled as bell-shaped probabilities then converted into masses using the inverse pignistic transform $Bet^{-1}()$. This transform provides a method to transform a set of belief masses into a probability density, and inversely, to perform the reverse transform. A pyramidal activation function is used to model the probability density of each fusion variable as a function of source measurements.

For a coefficient c_k, the probability is formulated as:

$$\psi_k(c) = max(0, 1 - \frac{|c - c_k|}{3\ \sigma_k}). \tag{5}$$

This representation has the advantage of integrating the standard variation of σ_k of each {source, variable} which are computed in a characterization pre-step.

The inverse Pignistic Transform is applied to the probability densities $\psi_{k,n}$ to obtain a set of belief masses $m_{k,n}$ for each variable k and each source n.

$$m_{k,n}(H) = Bet^{-1}(\psi_{k,n}), H \subseteq 2^{\Theta}. \tag{6}$$

A discounting operation is also applied to the input masses, using a weakening parameter $\alpha_{k,n} \in [0, 1]$ to account for our confidence in the inputs reliability. The masses $m_{k,n}(\Theta)$, representing the total ignorance, are also normalized.

$$\begin{aligned} m'_{k,n}(H) &= \alpha_{k,n} \times m_{k,n}(H), \\ m'_{k,n}(\Theta) &= \alpha_{k,n} \times m_{k,n}(\Theta) + 1 - \alpha_{k,n}. \end{aligned} \tag{7}$$

*3) **Combination**:* The DS conjunctive combination rule is applied. And to decrease the computing power required for the combination, the belief masses $m'_{k,n}$ are converted

into commonality functions $q_{k,n}$ [20] before applying the combination rule:.

$$\begin{aligned} q_{k,n}(H) &= \sum_{A \in 2^{\Theta}, A \supseteq H} m'_{k,n}(A), \\ q_{k,f}(H) &= \prod_{n=1}^{N} q_{k,n}(H). \end{aligned} \tag{8}$$

After the combination rule, the commonalities $q_{k,f}$ are reconverted into belief masses $m_{k,f}$ and then, using the Pignistic transform, into probability densities $\psi_{k,f}$ of the fused coefficients for the tracking step.

C. Particle Filter Tracking

After the fusion of different sources, a filtering system is necessary to smooth the output and the possible "spatial" jumps. In our framework, we employ a Particle Filter (PF) operating on the geometric model of each lane, where the state is comprised of {lateral offset, lane width, heading angle, curvature and curvature derivative}. The state particles are predicted using the speed and yaw rate information from the vehicles odometry and the weights are updated based on the consistency of their state with the output of DST fusion.

The advantage of employing the the Particle Filter in the fusion framework, is that we can integrate the complete fused data, *i.e.* the probability density of fused data, instead of only the max value, in the sampling step of PF.

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

The presented framework was implemented on Valeo demo-car equipped with following sensors: Frontal camera, 360° coverage camera cocoon, lidar cocoon, radar cocoon. In this paper, we use only the lidar cocoon setup. To guarantee the quality of the evaluation process, 16 datasets including all use cases given in Table I are generated from beltways and highways around Europe, US and Japan.

TABLE I
TRANSCRIPT OF GROUND TRUTH DATASETS.

Use-cases		
Curves	Tunnels	Exit lanes
Lane changes	Cut ins	Cut outs
Traffic jam / low speed	Highway (80-130 kph)	

B. Results evaluation

Availability, recall, precision and F1-score are used to measure how robust is the overall performance of our system. In autonomous driving, any disconnection of lane information induces that the driver needs to take control of the steering wheel. The availability (or function activation) is considered as a key performance indicator. Moreover, the proposed lane similarity function (2) is used to compute false and true positives (FP, TP) which are necessary to calculate the recall, precision and F1-score of lane detections. Based on the safety regulations, parameters of function (2) are chosen as $\tau = 0.5m$ and $t_{min} = 1s$.

The Table II gives the comprehensive performance rates assessing the benefits of using Hybrid VL fusion as well as observing if fusing stationary and dynamic data introduces noise especially when there is no conflict in perception. As expected, *Hybrid VL fusion* provides an availability rate of **97.53%** which is very close to maximum of the two while fusion recall, precision and F1-score values are better than the rbVL and soVL alone.

TABLE II
PERFORMANCE RATES OF VIRTUAL LANES ESTIMATION ($\tau = 0.5m$)

Algorithm	Availability	Recall	Precision	F1-score
rbVL	84.4%	70.63%	83.08%	0.75
soVL	**98.70%**	77.91%	78.96%	0.78
Hybrid VL fusion	97.53%	**83.18%**	**85.33%**	**0.84**

C. Discussion Analysis

For visualization, the system pipeline is given at Fig. 7. As it is seen from the captured camera image (A), the test vehicle is turning a sharp curve with other cars in its FOV. Despite of the high curvature, the road border detection gives the road course precisely at (B). Another difficulty in this kind of cases is the uneven lateral offset of the vehicles towards their lanes' centers. As a result of our similarity matrix construction, soVL is achieved to minimize this offset as it is seen at (C). Finally, (D) shows the hybrid VL fusion output which is merged rbVLs and soVLs.

The Table II presents performance of each VL algorithm individually. The availability rate of *rbVL* is 84.4%. The loss is generated by the vehicles that are hiding the RB infrastructure. Although occlusions happening only on one side are well tolerated (*i.e.* right occlusions while doubling truck convoys), visibility of RBs becomes really poor on both sides in heavy traffic jams conditions. In fact, there are even some cases where borders are damaged or do not exist during the test. Availability rate of *soVL* **98.70%** is higher because there is always at least one detected object around us thanks to the surround FoV. The missing rate of 1.30% is generated by far objects and inconsistent object trajectories (*i.e.* changing lanes or exiting the road).

On the other hand, the *soVL* precision is lower due to the observed oscillations coming from the relative distance and speed between the ego-vehicle and other moving objects (especially for high speed). Due to the stationary characteristic of *rbVL's* input, it is more robust to this kind of situation, it reaches 83.08% of precision rate. It is observed that the loss in the precision rate of *rbVL* is mainly proportional to the detection range. An erroneous heading and curvature estimation is observed when the longitudinal detection range is below $30m$.

Considering the uneven distribution of FN in *rbVL* and FP in *soVL*, the F1-score is a better measure to represent the

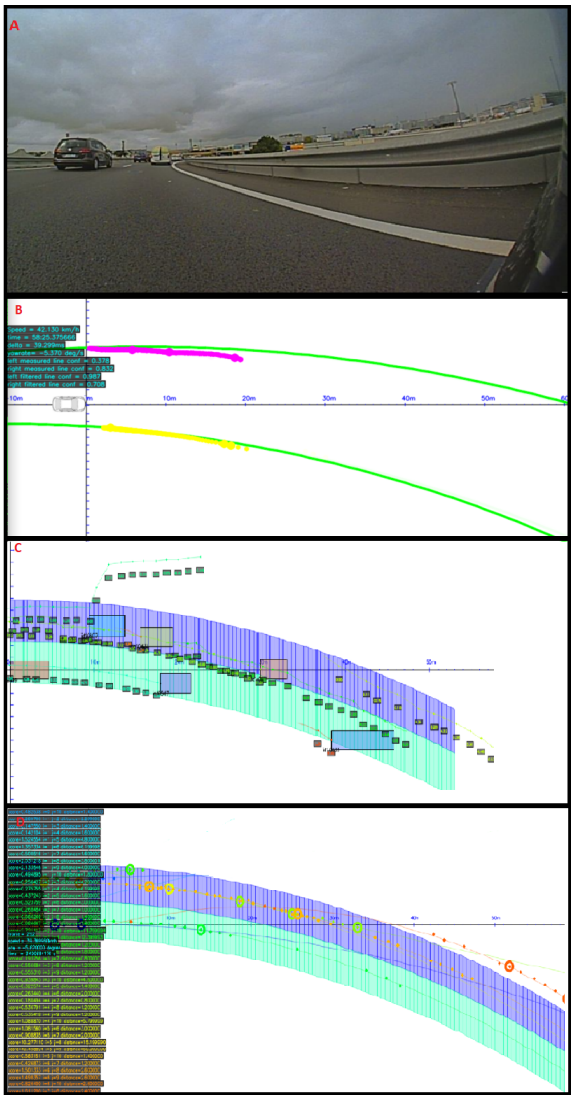

Fig. 7. VL system pipeline. (A) Frontal scene (B) *rbVL* (C) *soVL* (D) *Hybrid VL fusion*

performance of each and the improvement level obtained by Hybrid VL Fusion. The F1-score increases to 0.84, which proves the complementary between the two Virtual Lanes detections.

V. CONCLUSIONS

In this article, Hybrid DST fusion coupled with particle filtering is proposed to solve the lane estimation problem in non-structured situations. Using 360° lidar sensor set, two blocks: *rbVL* based on independent road-borders detection and *soVL* based on surround moving objects are merged. Through experimental comparative study we have demonstrated the potential of the VL fusion to manage multiple use-cases. With the resulting availability and accuracy rates, our model is a redundant source to other lane marking detection based solutions, and insures the lateral signal continuity for any minimum risk manoeuvres or AD backup solution.

In future work, an asymmetric modelling with other bell-shape activation functions will be tested in order to improve DST fusion. The empirically proposed weakening parameter is planned to be replaced by machine learning approaches. Moreover, introducing the HDmap cartography and static maps can increase the stability of the detection as a pre-validation step.

REFERENCES

[1] S. Durand, R. Benmokhtar, and X. Perrotton, "360 Multisensor object fusion and sensor-based erroneous data management for autonomous vehicles," in *IEEE Sensors Applications Symposium (SAS)*, 2019.

[2] S. Moujtahid, T. Liennard, and R. Benmokhtar, "Evidential multi-sensor fusion and erroneous management of lanes for autonomous driving," in *IEEE Sensors Applications Symposium (SAS)*, 2019.

[3] T. Stolte, G. Bagschik, and M. Maurer, "Safety goals and functional safety requirements for actuation systems of automated vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 2191–2198.

[4] A. Klotz, J. Sparbert, and D. Hötzer, "Lane data fusion for driver assistance systems," in *IEEE International Conference on Information Fusion*, 2004, pp. 657–663.

[5] S. Park, B. Kim, S. Jeong, H. Kang, and P. Yoon, "Lane estimation using lateral histogram in radar based ACC system," in *European Radar Conference*, 2014, pp. 193–196.

[6] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani, "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.

[7] A. Kirchner and T. Heinrich, "Model based detection of road boundaries with a laser scanner," in *IEEE International Conference on Intelligent Vehicles*, 1998.

[8] J. Kibbel, W. Justus, and K. Furstenberg, "Lane estimation and departure warning using multilayer laserscanner," in *IEEE Intelligent Transportation Systems*, 2005, pp. 607–611.

[9] C. Lundquist, U. Orguner, and F. Gustafsson, "Estimating polynomial structures from radar data," in *International Conference on Information Fusion*, 2010, pp. 1–7.

[10] U. Meis, W. Klein, and C. Wiedemann, "A new method for robust far-distance road course estimation in advanced driver assistance systems," in *IEEE International Conference on Intelligent Transportation Systems*, 2010, pp. 1357–1362.

[11] T. Kim and B. Song, "Detection and tracking of road barrier based on radar and vision sensor fusion," *Journal of Sensors*, pp. 1–8, 2016.

[12] T. Litman, *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, Canada, 2017.

[13] T. T. Nguyen, J. Spehr, J. Xiong, M. Baum, S. Zug, and R. Kruse, "A survey of performance measures to evaluate ego-lane estimation and a novel sensor-independent measure along with its applications," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2017, pp. 239–246.

[14] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[15] Á. F. García-Fernández, L. Hammarstrand, M. Fatemi, and L. Svensson, "Bayesian road estimation using onboard sensors," *IEEE Transactions on Intelligent Transportation systems*, vol. 15, no. 4, pp. 1676–1689, 2014.

[16] H. Zhu and O. Basir, "A novel fuzzy evidential reasoning paradigm for data fusion with applications in image processing," *Soft Computing*, vol. 10, no. 12, pp. 1169–1180, 2006.

[17] T. T. Nguyen, J. Spehr, J. Xiong, M. Baum, S. Zug, and R. Kruse, "Online reliability assessment and reliability-aware fusion for ego-lane detection using influence diagram and bayes filter," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent System (MFI)*, 2017, pp. 7–14.

[18] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor fusion using Dempster-Shafer theory," in *IEEE Instrumentation and Measurement Technology Conference*, vol. 1, 2002, pp. 7–12.

[19] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[20] P. Smets, "Belief functions on real numbers," *International Journal of Approximate Reasoning*, vol. 40, no. 3, pp. 181–223, 2005.