

Multisensor Tracking of Lane Boundaries based on Smart Sensor Fusion

Federico Camarda^{1,2}, Franck Davoine¹, Véronique Cherfaoui¹, and Bruno Durand²

Abstract—Lane detection plays a crucial role in any autonomous driving system. Currently commercialized vehicles offer lane keep assist and lane departure warning via integrated smart cameras, deployed for road markings detection. These sensors alone, however, do not generally ensure adequate performance for higher autonomy levels. In this paper, a multi-sensor tracking approach for generic lane boundaries is proposed. This solution is based on well-established filtering techniques and supports a flexible clothoid spline representation. It relies on fine-tuned measurement models, tailored on collected data from both off-the-shelf and prototype smart sensors. The implementation takes into account real-time constraints and ADAS ECUs scarcity of resources. The result is finally validated against lane-level ground truth and experimental data acquisitions.

I. INTRODUCTION

Perception is an essential stage in the development of autonomous driving systems. It precedes and lays the basis for path planning and decision making. In this key phase, the goal is to pursue two main outcomes: an overview of other users in the driving scene and a solid representation of the surrounding environment. The latter is tackled in this work and it is addressed as a lane boundaries estimation problem. Hereinafter, the term "lane boundary" equivalently identifies lane markings, barriers and any other road element relevant to the partition of the roadway into lanes. This environment modeling approach aims at identifying the edges of these fundamental travel corridors where vehicles can drive safely and efficiently. A reliable and accurate characterization of the roadway and its lanes would enormously benefit a lane-based navigation and positively support further tasks such as lane assignment of other vehicles, prediction of their future paths and smooth lateral control. To achieve this, it is proposed in this work to exploit multiple sources of smart sensor data, in an attempt to face the problem in challenging conditions, e.g. poor lane markings quality or unfavorable weather.

Our methodology focuses on highway scenarios where lanes exist and supposedly orient and organize the traffic flow. Some current automotive trends in ADAS development shall also be considered in this research. Nowadays, in fact, car manufacturers are used to integrate ready-to-use lane detection solutions (in the form of so-called smart cameras) from external suppliers. Therefore, towards higher levels of autonomy where redundancy and perception diversity will be essential, a multi off-the-shelf sensor fusion approach is here presented for tracking and estimating lane boundaries.

This paper is organized as follows. Section II describes the

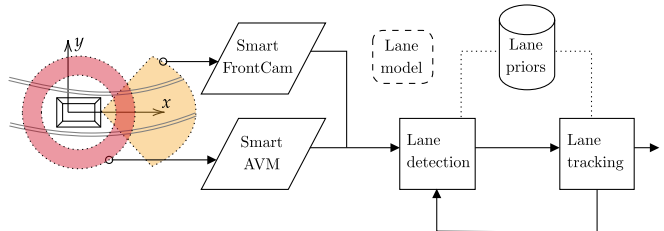


Fig. 1: Lane detection pipeline from [3].

related literature. Section III, Section IV and Section V detail the key points of our solution. Experimental results are presented in Section VI and commented in Section VII, together with future works perspectives.

II. RELATED WORK

In the general lane detection problem, the aim is to identify the number of the surrounding lanes and their geometric shape. The approaches in the literature differ on the level of the supposed prior knowledge available. In the case of precise localization solutions, the work of [1] is a valid example, the lane detection problem is addressed by precisely localizing (with centimetre accuracy) the vehicle in the reference frame a so-called high definition map, a lane-level representation of the environment. In the mentioned work and others [2], the lane detection problem finds its solution accessing the geographic representation of the environment and querying this prior knowledge (usually stored in the form of a database).

In our case, however, this prior is not considered to be available or reliable. The chosen approach is therefore more similar to the one proposed in [3]. Here, the main steps of the lane detection problem in general are described. The design choices made in these steps determine the final result, the working conditions and the scope of the possible application. An overview of the process, as proposed in [3] and its illustration (reported in Fig. 1), can be given as follows:

- 1) Lane model
- 2) Feature detection
- 3) Lane detection and tracking

These three main axes of the problem are taken as guidelines for the exploration of the literature.

A. Lane model

The choice of the lane model affects which types of lanes can be represented and the applicable estimation algorithms

¹Sorbonne universités, Université de Technologie de Compiègne, CNRS UMR 7253 Heudiasyc, France

²Renault S.A.S, Guyancourt, France

to fit it. Lane and road models can be categorized into three classes: parametric models, non-parametric models, and semi-parametric models.

1) Parametric models: a finite number of parameters maps into simple geometric curves. Common representations are straight lines, parabolic curves, circumferences arcs and hyperbolas. This first category has limited degrees of representation but it is suitable for efficient fitting algorithms such as RANSAC, Hough transform and vanishing point [4]. This enables simple applications but strong outlier and noise resistance, as in [5] where RANSAC and Kalman Filter are exploited along with a straight line model. On the other hand, if the intended application is more complex than a LKA (Lane Keep Assist), more flexibility is needed and strictly parametric solutions are not viable.

2) Non-parametric models: without an a priori specified structure the result is determined by the data. It may consist of a continuous representation (e.g. continuous pixels), but it might not contemplate any smoothness or differentiability. This paradigm is preferred in all of those applications where features are extracted directly from the image, still in the camera point of view. The most interesting implementations of this kind are generally based on deep learning [6].

3) Semi-parametric models: in the presence of both parametric and non-parametric components, these models do not assume any specific global path geometry. This category primarily includes different declinations of piece-wise defined functions, so-called splines. [7] presents a generic lane model, which also implements constraints and priors between lanes (parallelism, minimal lane width, etc.) as form of edges in a Graph-SLAM approach.

B. Feature detection

Sensor data is processed to extract environmental features useful for the actual estimation. Accordingly with the nature of the sensor, there is a variety of possible features to consider and extract.

For cameras and vision-based sensors, several existing works focus on extracting features based on colours, shapes and textures. The classic work [8] bases its lane detection module on the horizontal brightness variation of the input stereo-image, after the application of an Inverse Perspective Mapping (IPM) transformation. Nowadays instead, [6] represents the state of the art for lane detection, according to the TuSimple Benchmark Lane Detection Challenge [9]. Its deep neural network approach takes into account wide spatial relationships and results effective on extended objects with few appearance clues, i.e. traffic lanes, poles, and walls.

Different data-related attributes are considered for sensors of different nature, such as reflectivity of the material for Lidars [10] and echoes coming from road barriers for radars [11]. In [12], lane marking features are detected according to the intensity of the laser reflection, transformed into an intensity image and classified by the means of a Convolutional Neural Network. [13] bases his work on the same feature, with a simple straight line model in order to enhance the cross track localization of the vehicle.

C. Lane detection and tracking

Selected features are processed in the detection step to infer the presence of one or more lanes, or lane boundaries as in our case. Features are fit into the chosen lane model and an initial estimate is generated. If these detections are successively confirmed by fresh measurements, they are validated as actual tracks and at tracking stage we can exploit spatial and temporal continuity constraints to update the estimates as the vehicle moves and new observations are available. The algorithm resistance to outliers is tested, as well as its reaction to false positives. Additionally, new observations of already detected lane boundaries have to be correctly associated to existing tracks and contribute to their estimate and update. In these steps, the representation of uncertainties plays a crucial role when classical association techniques are applied. [14] estimates the geometry of a railway track using a spline model and localization samples from GNSS receivers installed on the trains. Each point of the curve has an associated uncertainty, which is taken into account in the estimate update.

In view of the existing literature and the selected focus of our research, our contributions are located in the lane detection pipeline (Fig. 1) as subsequent to the feature detection step. The solution presented in the following, in fact, is suitable for any kind of sensor technology or feature detection method (potentially including the previously mentioned state of the art techniques) if properly characterized in the measurement model introduced in Section II-B.

III. LANE BOUNDARY MODEL

A semi-parametric model is preferred to assure a more general representation for the tracked lane boundaries. The chosen model for each track $T_i \in \hat{\mathbf{x}}_t$ (track collection at instant t) is the clothoid spline, each curve is composed of a variable finite number of clothoid segments defined as follows:

$$S_j = [x_0, y_0, \psi_0, \kappa_0, \kappa_1, l, \Sigma_S] \in T_i \quad (1)$$

where the continuous description of each segment is given at curvilinear abscissa s by the Fresnel integrals [15]:

$$x(s) = x_0 + \int_0^s \cos\left(\frac{1}{2}\kappa_1\tau^2 + \kappa_0\tau + \psi_0\right) d\tau, s \in [0, l] \quad (2)$$

$$y(s) = y_0 + \int_0^s \sin\left(\frac{1}{2}\kappa_1\tau^2 + \kappa_0\tau + \psi_0\right) d\tau, s \in [0, l] \quad (3)$$

The uncertainty on this representation is given by the covariance matrix Σ_S with respect to each parameter of each segment. This transcendental functions require approximation methods to be handled but they accurately reflect the techniques used for road infrastructure design [15]. This representation is also immediately suitable for curvature-based control of the vehicle.

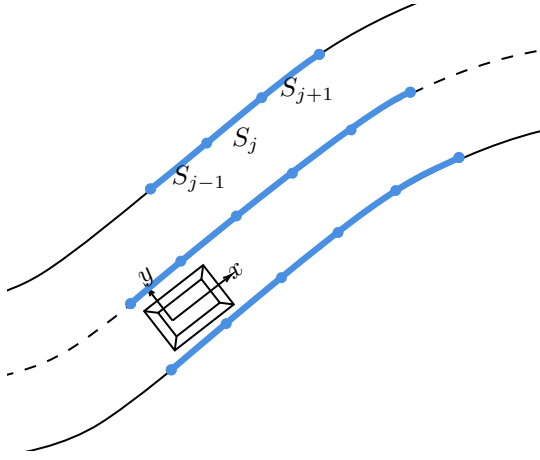


Fig. 2: Lane boundary model.

IV. MEASUREMENT MODEL

The sensor set of the target architecture is supposedly composed of smart sensors, capable of extracting meaningful features for lane detection. Independently of the sensor nature, their measurements are detection of lane boundaries in the driving scene. Measures $M_i \in \mathbf{z}_t$ at each iteration are represented in the body frame and modeled with polynomial curves, as follows:

$$M_i = [c_0, c_1, c_2, c_3, x_{min}, x_{max}, \Sigma_M] \in \mathbf{z}_t \quad (4)$$

where the continuous description of the measurement is given by:

$$P(x) = c_0 + c_1x + c_2x^2 + c_3x^3, x \in [x_{min}, x_{max}] \quad (5)$$

This polynomial model for measurements reflects a standard output provided by off-the-shelf devices adopted in the automotive industry. The measurement uncertainty is given by the covariance matrix Σ_M with respect to each parameter. However, no guarantee is given that the sensor will be able and willing to indicate the uncertainty for each detection. Therefore, a measurement noise evolution is modeled as follows:

$$\Sigma_M(d) = \exp(\alpha_M d) \Sigma_M(0) \quad (6)$$

where:

$$\Sigma_M(0) = \begin{pmatrix} \sigma_{xx}^2 & 0 & 0 \\ 0 & \sigma_{yy}^2 & 0 \\ 0 & 0 & \sigma_{\theta\theta}^2 \end{pmatrix} \quad (7)$$

In this model, the measurement noise scales an initial covariance matrix $\Sigma_M(0)$ by a factor that grows exponentially with the euclidean distance d from the sensor to the sample $(x, P(x))$. A graphical representation is given in Fig. 3, where measurements and corresponding covariance ellipses are shown in red. The model coefficients have been empirically tuned and adapted for different sensors.

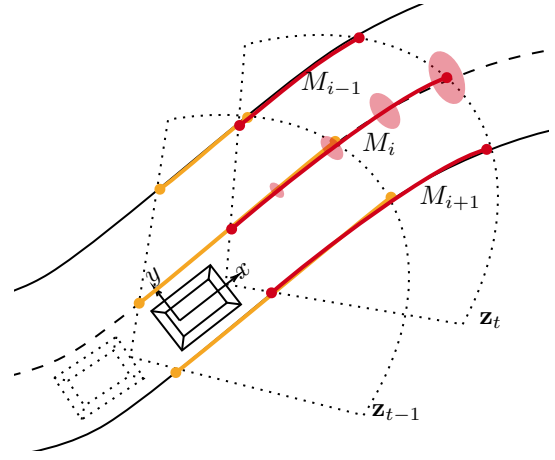


Fig. 3: Measurement model.

V. DETECTION AND TRACKING

This section describes the core of the proposed fusion architecture. The basic idea is that a set of ordered control points jointly with an appropriately chosen interpolation method can completely describe a geometric curve, representing lane bounding elements, defined on the clothoid spline model of Section III. In our framework, these control points are tracked and referred to as features. A curve C_i consists of a finite number of features F_j defined as:

$$F_j = [x_j, y_j, \theta_j, \Sigma_F] \in C_i \quad (8)$$

This representation describes a curve specifying abscissa x_j , ordinate y_j and heading θ_j of its control points. Considering the application domain and the uncertain nature of the described entity, these values constitute a random vector and Σ_F is its covariance matrix. In order to track and cumulatively refine the estimation of a feature set, the adopted procedure consists of the following steps:

- Initialization
- Prediction
- Association
- Update

For each detected and initialized feature, a filtering process is carried on using a traditional Kalman filter. This method is selected for its low computational cost and optimality under Gaussian noise assumption, which is supposed to be our case. Ultimately, the above mentioned spline representation is attained at each iteration interpolating the curve between two consecutive features by means of the algorithm proposed in [16]. This procedure builds the clothoid spline by determining the appropriate initial curvature κ_0 , curvature rate κ_1 and clothoid segment length l for each given couple of F_j and F_{j+1} . This method guarantees a G^1 degree of continuity of the spline, which is a mandatory requirement for an exploitable result. Therefore, for two consecutive clothoid segments S_j and S_{j+1} , it is verified that $x_j(l_j) = x_{j+1}(0)$, $y_i(l_j) = y_{j+1}(0)$ and $\psi_j(l_j) = \psi_{j+1}(0)$. Attaining G^2 (curvature) continuous curves would require

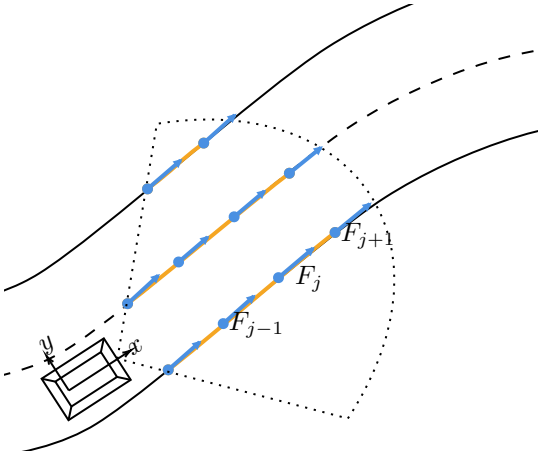


Fig. 4: Feature set initialization. Measurement in orange and sampled features in blue.

more computationally expensive algorithms [17] exceeding our working hypotheses.

A. Initialization

Whenever no curve is being tracked or the measurements delivered by the sensor have not been associated to any existing track, a new curve is initialized. Samples are extracted from the continuous measurement in order to initialize a new feature set. These samples are located at constant inter-distance (in term of curvilinear abscissa) and they define the initial conditions of each state variable. For a given $x_j \in [x_{min}, x_{max}]$:

$$y_j = P(x_j) = c_0 + c_1x_j + c_2x_j^2 + c_3x_j^3 \quad (9)$$

$$\theta_j = atan(P'(x_j)) \quad (10)$$

Σ_F is set in accordance with the measurement model in Section IV also taking into account which sensor is providing the measurement.

B. Prediction

Before being allowed to compare the existing tracks with the latest measurements, they both have to be defined in the same reference frame. At this step, the appropriate rotation and translation is applied to each curve in such a way that tracks are temporally and spatially coherent with the freshest sensor delivery. Where this would usually be handled with an ordinary geometric transformation, in this context it consists of a prediction. In fact, the actual parameters required for this operation are as well outcome of an estimation process. In the case of state prediction, an evolution model is generally introduced and its definition strongly impacts the final outcome. Within this context, however, the evolution model of our targets is perfectly known and trivial: control points of lane boundaries do not (in our scope) evolve in time, exclusively the reference frame is moving. This movement is separately estimated and defined at each iteration as:

$$\Delta Ego_t = [dx, dy, d\theta, \Sigma_E] \quad (11)$$

In view of this, the affine transformation (operator \oplus) of the state vector ${}^{t-1}\hat{\mathbf{x}}_{t-1}$ is firstly performed:

$$\hat{\mathbf{x}}_{t-1} = {}^tT_{t-1} \oplus {}^{t-1}\hat{\mathbf{x}}_{t-1} \quad (12)$$

where ${}^tT_{t-1}$ describes the translation applied along with a rotation of $d\theta$:

$${}^tT_{t-1} = \begin{pmatrix} \cos(d\theta) & -\sin(d\theta) & dx \\ \sin(d\theta) & \cos(d\theta) & dy \\ 0 & 0 & 1 \end{pmatrix} \quad (13)$$

The prediction step of our Kalman filter follows, according to the simple evolution model:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{w}_t \quad (14)$$

The prediction of the state variables is here subject to the uncertain estimation of the ego-movement, accordingly appearing in the filter prediction step, given that:

$$\mathbf{w}_t \sim \mathcal{N}(0, \Sigma_E) \quad (15)$$

C. Association

The association between measurements and tracks is crucial to successively integrate fresh upcoming information. In order to do this, an appropriate metric to express the distance between a feature set C_i and a measurement M has been defined. It is supposed that, between $t-1$ and t , the ego-motion is sufficiently small to allow at least the orthogonal projection of one feature on the measure, if they represent the same lane boundary. Under this assumption, each feature is projected on each measure. Each successful projection of $F_j \in C_i$ appoints to a correspondent feature $p_{\perp}(F_j)$ sampled along the measure M at (x_{\perp}, y_{\perp}) . The Mahalanobis distance between the two can be computed as:

$$d(p_{\perp}(F_j), F_j) = \sqrt{(p_{\perp}(F_j) - F_j)^T (\Sigma_M(x_{\perp}, y_{\perp}) + \Sigma_F)^{-1} (p_{\perp}(F_j) - F_j)} \quad (16)$$

The distance between a measure and a track is finally defined selecting the highest value, where multiple projections exist:

$$d(M, C_i) = \max_{F_j \in C_i} d(p_{\perp}(F_j), F_j) \quad (17)$$

A distance matrix built on these values allows the use of a Global Neighbour Neighbour (GNN) algorithm to finalize the association.

D. Update

Once the association between the measure M and the feature set C_i is confirmed, the aim is to update the state of existing features and to possibly extend the track with newly discovered elements. Existing features are updated in the filter update step with their projection $p_{\perp}(F_j)$ on the associated measurement. Where available, the remaining length of the measurement is exploited to initialize, as in Section V-A, newly discovered features and extend C_i . Fig. 5 shows unchanged, updated and new features respectively displayed in grey, green and blue.

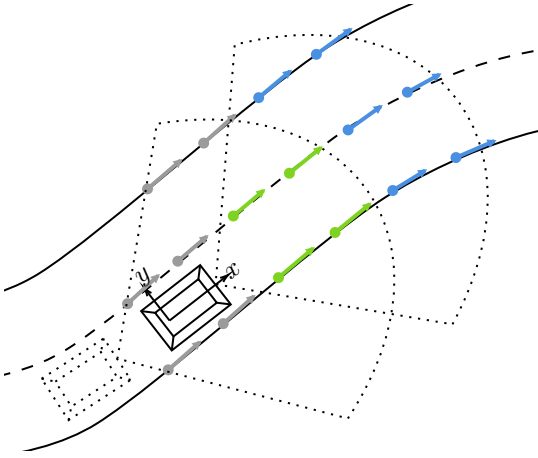


Fig. 5: Feature set update. Unchanged, updated and new features respectively displayed in grey, green and blue.

VI. EXPERIMENTAL RESULTS

The aim of the following experiments is to test the effectiveness of the proposed solution, using smart sensors as input. The feasibility of the on-board fusion also has been verified, targeting highway scenarios in fluid traffic at low medium speed (up to 60 km/h). The specificity of our research focus did not allow for an evaluation on public well-known datasets. For this reason, we collected custom-tailored data to be replayed in the framework described hereinafter.

A. Setup

The presented method is implemented in a proprietary environment for ADAS design. On this platform, the development is carried out mostly in C language, following MISRA C [18] guidelines for safe, reliable and portable code for embedded systems. Additionally, a proper management of message timestamps (measure dates, buffer dates and meaningful instants of the process) allows the monitoring of effective execution time. The presented solution is, in fact, specifically designed to comply with real-time constraints and low resources availability typical of ADAS ECUs. An interface example of the platform is reported in Fig. 6. The video stream shows a view of the driving scene at each instant and it can be used for a qualitative evaluation of the results, accordingly displayed. Data fusion algorithms are here designed and tested off-line, replaying recordings of data obtained on different prototype vehicles, where they can finally be exported for on-board testing.

The deployed vehicle for our experiments (an ad-hoc configured Renault Espace) is equipped for the perception of its surroundings with:

- Smart FrontCam, 30Hz, FoV: $53^\circ \times 120 \text{ m}$
- Smart AVM (4 cameras), 20Hz, FoV: $360^\circ \times 20 \text{ m}$

These sensors implement device-specific data processing algorithms and issue measurements in the format specified in Section IV. Our solution processes asynchronous sources equivalently and updates the lane boundaries state estimate at each delivery. Detections from the AVM sensor, however, are



Fig. 6: On the left, tracking results (blue) and measurements (orange for FrontCam, brown for AVM) are displayed in a bird-eye view. On the right, the context cameras.

not allowed to initialize new tracks. Being a smart sensor still in its early stage of development, this limitation is necessary to minimize false detections. Its contribution is anyway taken into account for existing tracks, narrowing our analysis focus to the FrontCam (which is not dissimilar to a mass-produced device) and to the impact of the fusion process. A lane-level ground truth is essential to achieve this evaluation. The vehicle has been additionally equipped with a GNSS-RTK precise positioning system and sensor data has been recorded on the French A86 (Créteil-Versailles). An HD-map of this highway (section of approximately 20 km long) has been specifically realized for the purpose of self-driving applications testing.

B. Results

A prior qualitative analysis of the fusion result has been done through the development platform output interface (Fig. 6) and with the aid of context cameras, confirming the soundness of the result on highway scenarios. In fact, the deployed association criteria can robustly discriminate measurements and tracks that refer to different lane boundaries (covering also close together road edges and markings). It has also been reasonably observed that, processing the two data sources, a fresher estimation is available at a higher rate. This corresponds in average to the sum of the sensors frequency, namely 50 Hz in our experiment.

The quantitative evaluation follows and assumes the topological information in the map to be accurately geo-referenced. Jointly with the RTK pose estimation and expressed in the body frame, it accounts for the ground-truth representation \mathbf{x}_t of our lane boundaries estimation $\hat{\mathbf{x}}_t$. Under these conditions, the FrontCam error \mathbf{e}_t^{FC} (accounting for our baseline) and fusion error $\mathbf{e}_t^{\text{Fusion}}$ are described by:

$$\mathbf{e}_t^{\text{FC}} = \mathbf{x}_t - \mathbf{z}_t^{\text{FC}} \quad (18)$$

$$\mathbf{e}_t^{\text{Fusion}} = \mathbf{x}_t - \hat{\mathbf{x}}_t \quad (19)$$

where appropriate format transformations are applied to perform these comparisons. In the scope of this work,

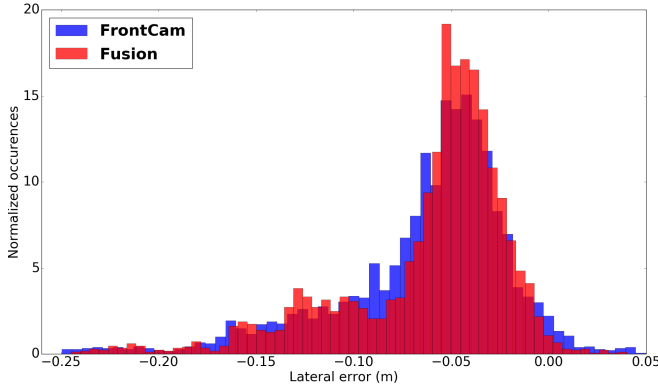


Fig. 7: Superposed, normalized error distributions e_0^L for FrontCam (blue) and Fusion (see-through, red)

the analysis focus has been limited to the following error indicators:

$$\mathbf{e}_t = [e_0^L, e_1^L, e_0^R, e_1^R] \quad (20)$$

where, for left L and right R markings, a separated lateral error is considered for different interval ranges of longitudinal distance (specifically, in $[0,10]$ m and $[10,20]$ m). These range intervals cover the FoVs intersection, where the multi-sensor fusion occurs.

At first, a characterization of the sensor set has been carried on. In Fig. 7, the histogram in blue shows an indicator of the FrontCam error distribution. On the one hand, the reported error density supports the Gaussian hypothesis on the measurement noise and the model presented in Section IV. This analysis also guided the fine-tuning of the model's covariance matrix and α_M coefficient. On the other hand, we can observe that this result does not appear to be *zero mean* as supposed. This offset (in the centimeter magnitude) can be attributed to the relative mismatching between the lane-level ground truth and our global positioning system. In fact, this gap occurs likewise in the error distribution of the fusion result which is presented in red, always Fig. 7. This lightly narrower density shows the smoothing effect of the filtering process and the contribution of the complementary AVM sensor. More precisely, a comparison in term of mean, variance and Root Mean Square Error (RMSE) of the selected indicators has been finalized. The produced benchmark is presented in Table I and confirms previous observations.

| | FrontCam only | | | FrontCam + AVM fusion | | |
|---------|---------------|------------------------------|----------|-----------------------|------------|--------|
| | μ [m] | σ^2 [m ²] | RMSE [m] | μ | σ^2 | RMSE |
| e_0^L | -0.0638 | 0.0020 | 0.0781 | -0.0620 | 0.0019 | 0.0755 |
| e_1^L | -0.0875 | 0.0027 | 0.1018 | -0.0773 | 0.0022 | 0.0906 |
| e_0^R | -0.1277 | 0.0039 | 0.1421 | -0.1018 | 0.0024 | 0.1131 |
| e_1^R | -0.1393 | 0.0044 | 0.1543 | -0.1254 | 0.0037 | 0.1394 |

TABLE I: Lateral error benchmark

VII. CONCLUSIONS AND PERSPECTIVES

In this paper, a multi-sensor tracking approach for generic lane boundaries is proposed. Although the experimental fusion of two smart sensors reported only a slight improvement

in term lateral RMSE, the solution confirmed its validity and coherency w.r.t. the lane-level ground truth. Its real-time implementation can support potentially any multi-modal smart sensor set, providing redundancy and perception diversity in the overall lane geometry estimation.

In future works, the integration of additional a priori from navigation maps is planned, along with a closer focus on road curvature estimation which remains a major challenge.

VIII. ACKNOWLEDGMENT

This work has been carried out within SIVALab, a shared laboratory between Renault and Heudiasyc (UTC/CNRS).

REFERENCES

- [1] F. Poggenschans, N. O. Salscheider, and C. Stiller, "Precise Localization in High-Definition Road Maps for Urban Regions," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 2167–2174.
- [2] F. Li, P. Bonnifait, J. Ibanez-Guzman, and C. Zinoune, "Lane-level map-matching with integrity on high-definition maps," in IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 1176–1181.
- [3] A. S. Huang and S. Teller, "Probabilistic lane estimation for autonomous driving using basis curves," *Autonomous Robots*, vol. 31, no. 2–3, pp. 269–283, Oct. 2011.
- [4] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2009, 96–103.
- [5] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and kalman filter," in IEEE Conf. Image Processing (ICIP), 2009, 3261–3264.
- [6] Pan, X., et al. "Spatial as deep: Spatial cnn for traffic scene understanding." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
- [7] A. Abramov, C. Bayer, C. Heller, and C. Loy, "A Flexible Modeling Approach for Robust Multi-Lane Road Estimation," arXiv:1706.01631 [cs], Jun. 2017.
- [8] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, Jan. 1998.
- [9] "TuSimple/tusimple-benchmark," GitHub. [Online]. Available: <https://github.com/TuSimple/tusimple-benchmark>.
- [10] S. Kammel and B. Pitzer, "Lidar-based lane marker detection and mapping," in IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, 2008, pp. 1137–1142.
- [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] B. He, R. Ai, Y. Yan, and X. Lang, "Lane marking detection based on Convolution Neural Network from point clouds," in IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 2475–2480.
- [13] F. Ghallabi, F. Nashashibi, G. El-Haj-Shhade, and M.-A. Mittet, "LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map," in IEEE International Conference on Intelligent Transportation Systems, 2018.
- [14] C. Hasberg and S. Hensel, "Online-estimation of road map elements using spline curves," 11th International Conference on Information Fusion, Cologne, 2008, pp. 1–7.
- [15] G. Marinelli, M. Bassani, M. Piras, and A. M. Lingua, "Mobile mapping systems and spatial data collection strategies assessment in the identification of horizontal alignment of highways," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 257–273, Jun. 2017.
- [16] E. Bertolazzi and M. Frego, "G1 fitting with clothoids," *Mathematical Methods in the Applied Sciences*, vol. 38, no. 5, pp. 881–897, Mar. 2015.
- [17] E. Bertolazzi and M. Frego, "Interpolating clothoid splines with curvature continuity," *Mathematical Methods in the Applied Sciences*, vol. 41, no. 4, pp. 1723–1737, Mar. 2018.
- [18] Motor Industry Software Reliability Association, Ed., MISRA-C:2004: guidelines for the use of the C language in critical systems, 2. ed, [Stand]: October 2004. Nuneaton: MIRA, 2008.