HAL open science

End-to-End Probabilistic Ego-Vehicle Localization Framework

Abderrahim Kasmi, Johann Laconte, Romuald Aufrère, Dieumet Denis, Roland Chapuis

To cite this version:

Abderrahim Kasmi, Johann Laconte, Romuald Aufrère, Dieumet Denis, Roland Chapuis. End-to-End Probabilistic Ego-Vehicle Localization Framework. IEEE Transactions on Intelligent Vehicles, 2020, pp.1-1. 10.1109/TIV.2020.3017256 . hal-03049396

HAL Id: hal-03049396

https://uca.hal.science/hal-03049396v1

Submitted on 9 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

End-to-End Probabilistic Ego-Vehicle Localization Framework

Abderrahim KASMI[1,2], Johann LACONTE[2], Romuald AUFRERE[2], Dieumet DENIS[1], Roland CHAPUIS[2]

Abstract—Locating the vehicle in its road is a critical part of any autonomous vehicle system and has been subject to different research topics. In most works presented in the literature, ego-localization is split into three parts: Road level-localization consisting in the road on which the vehicle travels, Lane level localization which is the lane on which the vehicle travels, and Ego lane level localization being the lateral position of the vehicle in the ego-lane. For each part, several researches have been conducted. However, the relationship between the different parts has not been taken into consideration. Through this work, an end-to-end ego-localization framework is introduced with two main novelties. The first one is the proposition of a complete solution that tackles every part of the ego-localization. The second one lies in the information-driven approach used. Indeed, we use prior about the road structure from a digital map in order to reduce the space complexity for the recognition process. Besides, several fusion framework techniques based on Bayesian Network and Hidden Markov Model are elaborated leading to an ego-localization method that is, to a large extent, robust to erroneous sensor data. The robustness of the proposed method is proven on different datasets in varying scenarios.

Index Terms—Autonomous vehicle, localization, lane marking, map matching, road prior, fusion framework.

1 Introduction

Over the past decades, the automotive industry has been growing strongly. The demand for the driver and pedestrians security coupled with the technological advances are the main factors behind this exponential growth. In that regard, Advanced Driver Assistance Systems (ADAS) have spread out. The main mission of these systems is to ensure that driver safety is constantly guaranteed. For this purpose, multiple applications have been deployed, such as lane departure warning, lane keeping assist, pedestrian detection, collision avoidance, or lane change assist system. To achieve this mission, the faultless knowledge of the localization of the ego-vehicle with regards to the surrounding environment is necessary.

The ego-localization of a vehicle is a mandatory component for a safe autonomous driving. Direct applications range from the decision-making system [1] regarding the decisions to make in order to keep lane or to change lane, to navigation: including path planning and vehicle control [2]. Ultimately, the lane/road understanding demands in terms of precision and false alarm rate [3] vary from one application to another. Therefore, the ego-localization solution must suit perfectly the localization requirement for each application.

The ego-localization task has been widely tackled over the years. The current literature is teeming with solutions that address this issue in a variety of manners. However, one interpretation of ego-localization consists of the knowledge of three key components:

I **Road level localization:** the road on which the vehicle travels.

II **Lane level localization:** the position of the host lane within the road (the lane on which the vehicle travels).

III **Ego-lane level localization:** the position of the vehicle in the lane in terms of lateral and longitudinal position.

For the road level localization, digital maps (Google, OpenStreetMap (OSM) or Waze) are used to perform this task, GPS receivers are used to retrieve the geographic (latitude, longitude) coordinates, and map-matching procedure is performed in order to match the position of the ego-vehicle with the correct road ('link'). However, the accuracy of the localization obtained is in the order of meters. Indeed, according to the Federal Aviation Administration (FAA) GPS Performance Analysis Report [4], the accuracy of a standard GPS device is within $3m$ with a 95% confidence, which can not be sufficient for most ADAS that require a more precise localization.

For some applications like lane-keeping, knowing the road on which the vehicle is traveling is not sufficient. These systems must be informed about the position of the host lane in the road to provide the adequate maneuver instruction and maintain the vehicle safety.

Further, autonomous vehicle applications need a more accurate localization, which can be translated by the knowledge of the lateral and longitudinal position of the vehicle in the ego-lane. For instance, overtaking maneuvers need a faultless knowledge of the lateral position of the ego-vehicle with respect to the ego-lane marking in order to decide whether the vehicle should overtake the obstacle or not.

The task of vehicle localization is still challenging for an autonomous vehicle, a complete ego-vehicle localization must perform all the three key components described above. Thus, in this work, an end-to-end solution for ego-localization from Road level localization to Ego-lane level

*This work has been sponsored by Sherpa Engineering and ANRT (Conventions Industrielles de Formation par la Recherche).

[1] *Sherpa Engineering, R&D Department, 333 Avenue Georges Clemenceau, 92000 Nanterre, France.* [a.kasmi, d.denis]@sherpa-eng.com

[2] *Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont-Ferrand, France.* FirstName.Lastname@uca.fr

localization is presented.

2 Related Work

Due to the vastity of this topic, this section will discuss the main vehicle localization techniques relevant to our presented work. To do so, the current literature is divided into three distinct topics: Localization on a map, Localization on a road and Localization on a lane.

2.1 Localization on a Map

For Autonomous Vehicles, map-matching (MM) algorithms are a critical piece on any localization system. In essence, MM algorithms integrate the geographical position from a GPS receiver with the spatial road network to identify the correct road on which the vehicle is traveling. In [5] a comprehensive literature survey of common MM methods is presented. The basic method combines an inertial measurement unit (IMU) with a classic GNSS receiver (e.g., GPS, Galileo) into a Kalman Filter to choose the correct road [6] [7]. In the same manner, authors in [8] propose a probabilistic MM that takes into account both the spatial (geographical coordinates) and temporal information (speed and time) to determine the likelihood road on which the ego-vehicle is traveling. However, in more complicated cases, such as urban scenarios, multi-path GPS signals lead to a lack of accuracy. In this case, the use of topological and geographical information enhances the proposed algorithm. Towards this end, many studies proposed a Hidden Markov Model (HMM) to take into account the measurement noise [9]. The authors in [10] introduce a HMM to robustify the proposed MM algorithm. The HMM incorporates the topological and the geographical road network information in the transition state to ensure the temporal connectivity between links. A more recent approach presented in [11] utilizes the HMM for the MM task. In essence, the two approaches are similar, except for the HMM modeling. Indeed, in [11] the transition from one 'link' or segment to another must obey some rules, these rules include the dynamics of the vehicles and the legal and logical connectivity of the road map. In a more specific way, the latest developments use the OpenStreetMap (OSM) database to perform the MM. In our previous work [12], a multi-criteria map-matching algorithm based on multiple probabilistic criteria has been introduced. Nevertheless, the road map topology has not been properly operated in the MM process, which consists of one of the contributions of this paper.

2.2 Localization on a road

Knowing the position of the ego-lane is still a challenging task for any localization system and subject-matter of research. Indeed, lane-level localization has been widely discussed in recent literature. According to [13], ego-lane localization can be performed in two different manners: model-driven and deep leaning approaches.

2.2.1 *Model-driven approaches*

The current literature abounds with ego-lane recognition techniques based on model-driven approaches. In these approaches, road level features are extracted from images via feature extraction, mainly two features (edge and color) are used. For the first feature, filters are used to extract lanes edges, then the resulted outputs are fed into Hough Transforms [14] or Ransac methods [15] to detect lanes in the road. For the color features, it exploits the primary color or the direction of lane components (arrow, zebra marking). Once these features extracted, they are fed into a high-level fusion framework. As an example, in [16] ego-lane recognition is achieved from multiple-lanes detection. Adjacent lanes are first hypothesized assuming the same curvature and lane width in the road, then tested using a video-based system. This approach shows interesting results. However the occlusion of lanes marking by obstacle vehicles can not be explained and will jeopardize the output of the ego-lane. More high level features systems fuse visual features from the road scene: surrounding vehicles [17], lanes marking [18] [19], road clues such as arrows marking [20], lane marking colors [21], lane marking and adjacent vehicles [22]. These visual cues are fed into a fusion framework: Bayesian filter (BF) [17], Bayesian Network (BN) [20] [18] [21], Hidden Markov Model (HMM) [19] or combination of BN and HMM [22]. A worth mentioning work is the one presented in [23] in which a lane-level accurate map is used in order to match the correct localization of the ego-vehicle. The map contains center lines of every lane of the road in addition to the lanes marking. Therefore to know the lane on which the vehicle is travailing a map matching procedure is presented. The presented method deals with the ambiguities encountered in choosing the right 'link'. Nevertheless, this method is tributary to the accurate map. In contrast, we present a method for lane level localization that relies only on a camera and a coarse map (OSM).

2.2.2 *Deep learning approaches*

Convolutional Neural Networks (CNNs) have also been widely adopted for lane level localization. These techniques do not take any prior about the surrounding environment, which makes these unrelated to the type of road-scenario (e.g., highway, urban). However, a learning phase is needed to set the weights of the used network.
In [24] local context of the scene is used to consider occlusion to determine the end point of the local lane segment, ego-lane is deduced once lane segments are extracted. Another research [25] uses a combination of CNN and RNN to detect lane boundaries. Similar to this approach a CNN is trained in an end-to-end way to estimate the ego-lane [26]. In [13] an end-to-end ego lane estimation is presented using a deep learning network SegNet. For all the mentioned methods before, no road prior is exploited. However, a learning stage is needed in all cases, which may require a large database to perform it.

2.3 Localization within a lane

In order to guarantee the ego-lane level localization, several data fusion methods have been investigated. The first one

uses a GPS receiver with a digital map to locate the ego-vehicle in the lane. The lack of accuracy provided by a classic GPS that can be caused by poor satellite signals, high degree dilution of precision, or multi-path in urban scenes, is first compensated with proprioceptive sensors, such as Inertial Measurement Unit (IMU). These methods are well known as Dead Reckoning [27]. Nevertheless, the accuracy provided by this kind of method is not sufficient for autonomous navigation, where centimetric accuracy is needed. Furthermore, dead reckoning methods suffer from integration errors that tend unbounded in time.

Leafing through the literature, it appears that most researchers use lane marking detection from a camera to provide an accurate ego-lane level localization. There are two main advantages of using the camera: currently, this sensor is the cheapest and most versatile modality for automotive applications and it provides dense information of the environment. Common approaches are based on a two-step process [3]. First, a road marking feature detection [28], then a fitting procedure [29] is used. This fitting procedure can be done with a polynomial fitting model based on the road geometry design [30] or non-parametric model [31]. The above-mentioned approaches suffer from the camera's weaknesses: high luminosity variation and occlusion issues.

In an attempt to provide a more robust and more accurate localization, the trend for manufacturers and researchers is to equip the ego-vehicle with multiple sensors such as Velodyne lidars, multiples cameras, IMUs and high defined digital map that contains the precise location of high features such as the lane marking or landmarks [32]. In that way, authors in [33] used two lateral cameras combined with an IMU and a GPS receiver to accurately determine the lateral distance of the ego-vehicle with respect to the lane marking. To do so, a vision-based ego-lane marking detector was used [34] and fused with the IMU sensors through an Extended Kalman Filter (EKF) to locate the ego-vehicle. Furthermore, a sub-decimeter accuracy was achieved by adding lanes marking from a created map. Yet again, an EKF was introduced for the fusion stage between the lane marking extracted from the map and the marking obtained through the vision process. In order to bound the integration errors coming from dead reckoning, authors in [35] present a localization system based on the fusion of multi-sensor and a high definition map. The integrity of the localization is insured by a Student's distribution that replaces the classical Gaussian distribution used in fusion filter like Kalman filters in an Informational Filter (IF). Close to our work, an atypical approach presented in [36] relies on the use of road priors and contextual information for road detection (roadway detection). OSM map is used to create the road backbone depending on the lanes number and the width of the lane. This road shape is then projected on the image taking into account the uncertainties related to the ego-vehicle pose. The result is used as prior to roadway detection. The road shape is then delimited by estimating the vanishing point and the lane marking. Naturally, the mentioned approaches are just a small part of the ego-lane localization techniques, an interesting survey is presented in [3] for more detailed approaches.

To the best of the authors' knowledge, there is no end-to-end localization solution proposed that tackles every aspect of the ego-localization task. Thus, in this paper, we present a complete solution for the overall localization problem.

Unlike existent work, this contribution distinguishes by the information-driven approach. Indeed, we exploit the available data from the OSM map in order to extract prior about the road geometry. This strategy allows us to focalize the region of interest in the recognition procedure and ultimately reduce the computation complexity. In addition, we propose a probabilistic model that will ensure the geometric coherency between lanes marking. Finally, the main contributions can be summarized as follows:

1) An enhanced map matching method based on OSM datasets and a Hidden Markov Model
2) An ego-lane detection based on a prior OSM map and a vision based recursive recognition ego-lane marking in an information-driven fashion.
3) Once the ego lane marking is detected and the lateral distance between the ego-vehicle and the lane marking estimated, we proceed to the ego-lane determination (the postion of the ego-lane in the road) using a combination of a Bayesian Network and a Hidden Markov Model.

We perform the complete localization steps in a sequential fashion. Thereby, the robustness of the proposed approach is highly improved. In addition to that, the split between the different parts of the localization allows the use of any part of the complete solution individually.

3 Overall localization algorithm

In this work, we present an end-to-end algorithm for ego localization as highlighted on Figure 1. The algorithm is intentionally divided into three modules: Road-level localization (RLL), Ego-lane level localization (ELL) and Lane level localization (LLL). This split allows us to be in phase with the current literature. Moreover, the pipeline of the algorithm allows to have modular algorithms that can be changed in the future without changing the entire algorithm architecture. In the following, we will first introduce our map-matching module based on OSM datasets, then we will use this map as a *prior* for our ego-lane marking detection in order to locate the vehicle within its lane. Once the ego-lane level localization is performed, we will proceed to the identification of the ego-lane in the road.

3.1 Road-level localization (RLL)

In this section, we present our road-level localization algorithm using OSM datasets. As shown in Figure 2 the proposed module is an upgrade of our work presented in [12]. Indeed, an HMM is added to robustify the proposed Map-Matching. Therefore, the algorithm is divided into two different stages:

- **Discrimination stage** elimination of incongruent links based on distance, orientation difference between vehicle steering, and links heading. Additionally, a third factor was implemented, it is based on the maximum speed allowed on the road.
- **Selection stage** in the case where the remaining candidate links are greater than one, we proceed into a selection procedure based on multiple probabilistic criteria to eliminate the ambiguity.

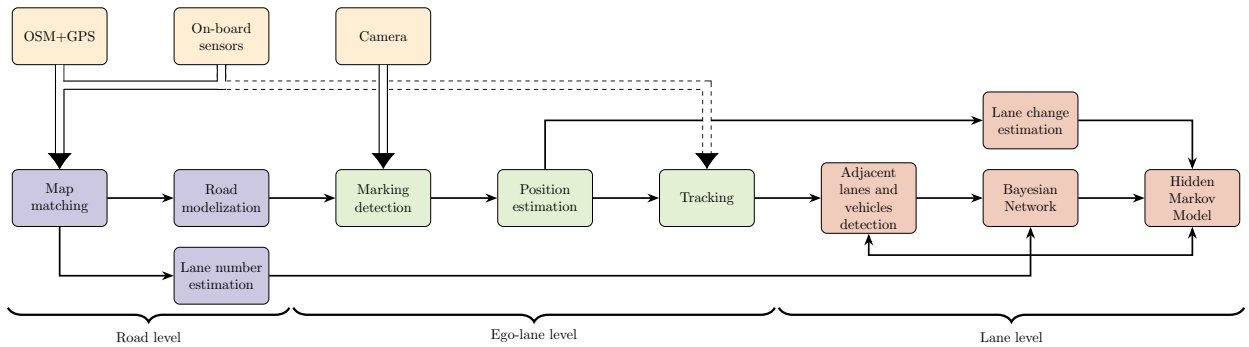

Fig. 1: Overall algorithm for complete localization of the ego-vehicle starting from the Road-level localization (RLL): to locate the ego-vehicle on the map Using OSM and GPS datasets, to the Lane-level localization (LLL): to locate the ego-vehicle within its own lane using prior road model from OSM and camera images and the Ego-lane level localization (ELL): to determine the ego-lane position in the road.

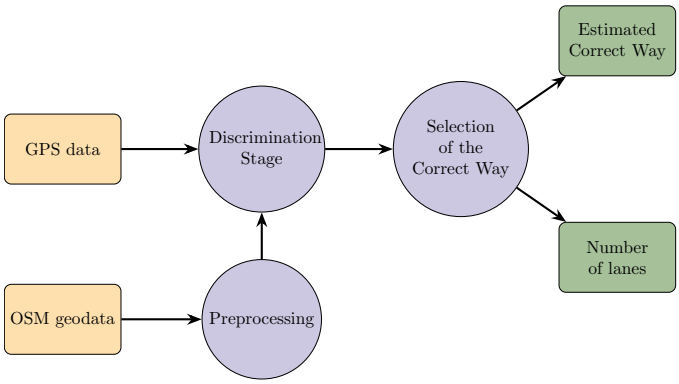

Fig. 2: Overall algorithm for the RLL using GPS data and OSM in order to extract the correct road and the corresponding number of lanes.

3.1.1 Discrimination stage

Before starting with this stage, we must present the OSM data. OSM data are composed of three keys components: Nodes, Ways, and Relations [37]. Nodes are the geometrical elements that represent GPS points. Ways are an ordered list of nodes that represents roads network, every Way has a number of tags, the latter provide information about the characteristics of the Way (number of lanes, limitation speed...). In addition, each Way is composed of a set of segments. In other words, belonging to a segment is equivalent to belonging to an OSM Way. So the map matching task can be reformulated as matching a GPS point with a segment. In the remaining sections, the words segment and Way are switchable.

Thus, the discrimination stage consists of the exclusion of segments which are not compatible with the current vehicle and the road network topology configuration. To do so, we introduced three discrimination criteria [12]:

- The first one is based on the distance between the ego-vehicle position estimated with the GPS data and the segment.
- The second one is based on the angle difference between the ego-vehicle steering angle and the segment heading (which expresses the traffic flow).
- The third one is based on the speed limit. The assumption made is that the ego-vehicle is respecting the limitation of speed to some extent i.e. the ego speed can not be higher than the limitation plus $40\,\mathrm{km\,h^{-1}}$.

Once the discrimination stage completed, if the number of candidates segment is higher than one, we proceed to the selection of the correct Way.

3.1.2 Selection of the correct Way

We define the state of ego-vehicle at time step t as $\mathbf{x_t} = (x_w, y_w, \theta_w)^T$. The selection procedure can be formulated as finding the highest conditional probability of belonging to a Way W_i knowing the pose of the ego-vehicle $\mathbf{x_t}$ at each time t:

$$\arg\max p(W_{i_t}|\mathbf{x_t}) \tag{1}$$

The main goal of this formulation is to select the correct Way on which the vehicle is traveling. Consequently, a selection criterion must be introduced to determine the correct Way. Based on equation (1) three probabilistic criteria [12] were developed:

- C_e: Criterion based on Euclidean distance.
- C_m: Criterion based on Mahalanobis distance.
- C_p: Criterion based on the probability of belonging to a segment.

Depending on the data available, one of these criteria is calculated for each GPS data and for each candidate Way. Furthermore, to take into account the relation between two consecutive frames, a convenient and effective approach is to use a HMM. Indeed, the change in Ways over time is governed by topology constraints that can be embedded in the transition state of the HMM. Thus we enhance the proposed multi-criteria algorithm with an HMM. The HMM in this work is an upgrade of the proposed HMM by [38] [39]. We fitted the HMM to the way selection problem. However, the big difference lies in the probabilistic reasoning. Indeed, we use criteria that represent probability to model the observation probability, which is not the case in [38] [39], where arbitrary functions have been introduced leading to non-intuitive criteria.

By definition a HMM consists of two stochastic processes, the first one is a Markov Chain to model the change of a state vector over time. This change is governed by a probability that describes the transit probability over time, which is called the transition probability. In the HMM, the states of the chain are not visible but observable, for this reason they are called "hidden". The second process is called the observation space, it produces emission of the observation at each time. Although the elements of the state vector are hidden, there is a relation between the hidden elements of the state and the observations, this relation is referred as an emission probability. Figure 3 describes the proposed HMM for the map-matching task.

So, in order to model the HMM-MM, three components must be defined:

State space the state of the system describes the list of the Ways candidates for each observation $\mathbf{x_t}$. We will use S_t to denote the set of candidates Ways at time t $S_t = \{W_1, W_2, ..., W_{n_t}\}$ with n_t the number of candidates Ways at each time t, S_t is a form of categorical distribution, $S_t \sim Cat(\xi)$.

Observation space for each candidate Way composing the state space, an emission probability is made. This emission probability is directly calculated from the multi-criteria algorithm. For each Way candidate the emission probability $\mathbb{P}_e(W_i)$ is calculated as follows:

$$\mathbb{P}_e(W_i) = \frac{C_k(W_i)}{\sum\limits_{j \in S_i} C_k(W_j)} \tag{2}$$

where C_k is the probabilistic criterion used with $k \in \{e, p, m\}$.

Transition probability The transition probability reflects the probability of the transition matrix that a state will move from one state to another. In the map matching procedure, the topology of the network is used in order to determine the transition matrix. Indeed, the vehicle can only move on Ways that are physically connected.

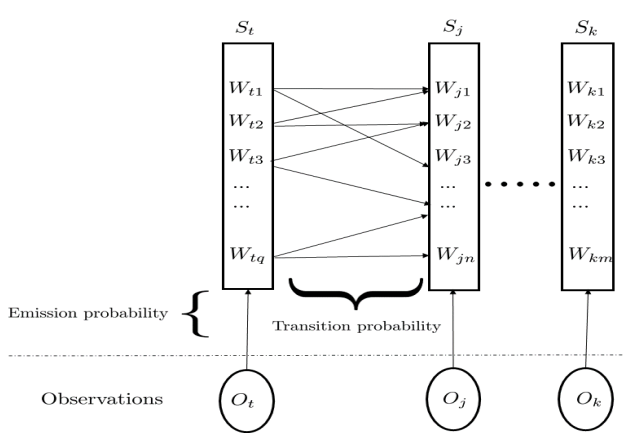

Fig. 3: Modeling of the Hidden Markov model Map-Matching (HMM-MM) algorithm, with S_t the state space at time t, W_t the Way candidate at time t and O_t the observations at time t. Note that the number of Way candidates at each time may vary.

TABLE 1: Transition matrix for example shown on Figure 4

	W_1	W_2	W_3	W_4	W_5
W_1	s_{11}	0	c_{13}	0	c_{15}
W_2	0	s_{22}	0	0	0
W_3	c_{31}	0	s_{33}	c_{34}	c_{35}
W_4	0	0	c_{43}	s_{44}	0
W_5	c_{51}	0	c_{53}	0	s_{55}

We will refer to $\mathbb{P}(W_i^t, W_j^\tau)$ as the transition probability from Way W_i^t to Way W_j^τ given the state space S_t and S_τ for time t and τ:

$$\mathbb{P}(W_i^t, W_j^\tau) = \frac{w_i}{\sum\limits_{j \in S_t} w_j} \tag{3}$$

Where w_i is a criterion calculated for each Way, this criterion obeys some rules defined as follows:

1) $w_i = 0$ if the way W_i^t and W_j^τ are not connected;
2) $w_i = s_{ij}^{t\tau}$ if the Way W_i^t and W_j^τ are the same;
3) $w_i = c_{ij}^{t\tau}$ if the Way W_i^t and W_j^τ are connected.

Figure 4 shows a simplified road network to illustrate the transition matrix probability. These probabilities are shown on Table 1. It can be noticed that in this example the number of Ways is the same for two connected states, which is not always the case. Indeed, the number of Ways may vary from two consecutive frames.

The transition probability $s_{ij}^{t\tau}$ is used whenever the two Ways are the same. Let $P(x_w, y_w)$ be the vector describing the Cartesian coordinates of the ego-vehicle in the Universal Transverse Mercator coordinate system frame (UTM), we note P' the projection of the point P on the Way and ψ the Way's heading, the Way W_i^τ is composed of two nodes n_1, n_2 as represented on Figure 5. The criterion s_{ij} is computed as a Gaussian distance to the middle of the segment $[n_1, n_2]$. Indeed, the probability to change the Way will be higher if P' is in the middle and will be lower in the limits, thus it is calculated as follows:

$$s_{ij} = e^{-\frac{1}{2}\left(\frac{t - u_0}{\sigma_0}\right)^2} \tag{4}$$

With $P' = t.n_1n_2$, $u_0 = 0$ and we take $\sigma_0 = 0.25$.

If the Way W_i and W_j are connected then we introduce a criterion based on [39], this criterion depends on the

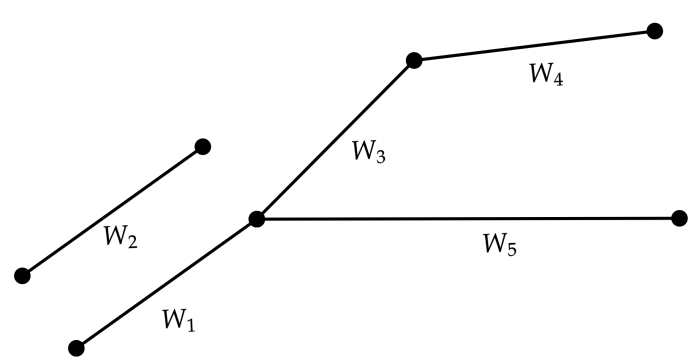

Fig. 4: Example of road network to illustrate the transition probability calculation. In this example the number of Way candidates is the same between two consecutive frames

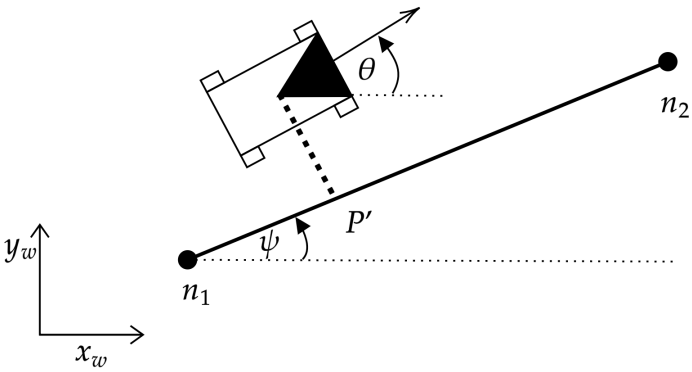

Fig. 5: Example of calculation for transition probability $s_{ij}^{t\tau}$, the vehicle is illustrated by the box with black triangle, ψ represents the Way heading and θ the vehicle's heading

difference between the vehicle's heading and the Way's heading :

$$c_{ij}^{t\tau} = e^{-\beta|\Delta\theta - \Delta\psi|} \tag{5}$$

with $\Delta\theta$ being the vehicle's heading change over time, $\Delta\psi$ the Way's heading change between the two Ways W_i and W_j and β a chosen coefficient. Once the Way selected, the number of lanes is extracted from the OSM database.

3.2 Ego-lane level localization (ELL)

Once the RLL has been performed, the ELL is initiated. We choose to start the ELL before the LLL for several reasons: the first is that the ego-lane marking is, for the most part, the easiest one to detect. The second one is due to the use of the information driven approach. Indeed, when the ELL is completed we have an estimate of the ego-vehicle's lateral position in its own lane. So, knowing the lane's width and the lanes number allows us to interpolate research zones for other lane marking.

To perform the ELL, the road is modeled in a 2D vehicle's frame (x_v, y_v) as a cubic polynomial [40] [41]:

$$x_v = \frac{1}{6}c_0 y_v^3 + \frac{1}{2}c_1 y_v^2 + c_2 y_v + c_3 \tag{6}$$

The parameters c_0 is the curvature's derivative of the road, c_1 the curvature of the road, c_2 the vehicle's heading with respect to the tangent of the road, c_3 the lateral shift of the ego-vehicle with regards to the road model. Once the road has been modeled in the vehicle's frame, the next step is to project this model on the image taking into account the intrinsic and extrinsic parameters of the camera. According to [42] the projection of the road model (left and right lines) in the image frame (u_i, v_i) is defined as follows:

$$u_i = e_u \left(\left(\frac{e_v Z_0}{(v_i - e_v\alpha)} \right)^2 \frac{c_0}{6} - \frac{e_v Z_0}{2(v_i - e_v\alpha)} c_1 + \frac{v_i - e_v\alpha}{e_v Z_0} (x_0 \pm \frac{L_w}{2}) - x_0 \right) \tag{7}$$

where $e_u = f/du, e_v = f/dv$, f is the focal distance of the camera, du and dv are the width and height of a pixel in the image, Z_0 is the height of the camera, x_0 the lateral distance of the ego-vehicle with respect to the ego-lane marking, the $\pm$ sign indicates whether the ego-marking is right $(+)$ or left $(-)$, α is the camera tilt angle and L_w is the road width. $i = 1, ..., n_{Roi}$ with n_{Roi} the number of Region Of Interest (ROI) for each lane marking.

As mentioned in [12], the OSM map does not provide information about the accuracy of its data. However, it is well known that OSM is a collaborative project in which volunteers provide the geospatial data. If we consider that most of the volunteers have a classic GNSS receiver, the accuracy of the OSM is thus metric. That being said, the polynomial representation of the road will be affected by this unknown accuracy. For the parameters c_0 and c_1, they represent the shape of the road. Given that we are working mostly with highway roads, their values will not be affected by the inaccurate geospatial accuracy of the OSM. In contrast, the parameter c_2 is more sensitive to inaccurate accuracy of the OSM, and its value will be affected. Hence, to compensate for the inaccuracy of its value, we will define $X_v = [x_0, c_2]^T$ as the state vector of the road model in the ego-vehicle frame. We associate a covariance matrix to this vector C_{X_v}. This matrix expresses the allowed dispersion around the average parameters previously defined by the vector X_v. Concerning the parameters c_3, the value extracted from equation (6) is not used. Indeed, it express the lateral shift of the vehicle to the road. Hence it is not used in equation (7), as we use x_0, which is the lateral shift regarding the center of the ego-lane. It assumed that the ego-vehicle travel in the center of the ego-lane thus x_0 is around 0.

Using equation (7), we can express the probabilistic model (X_v, C_{X_v}) into the image space. We will note this probabilistic model as u', C'_u, with u' being the average values for the pixel in the image and C'_u its corresponding covariance matrix:

$$C_u = J_u C_{X_v} J_u^T \tag{8}$$

with:

$$J_u = \begin{bmatrix} \frac{\partial u_1}{\partial x_0} & \frac{\partial u_1}{\partial c_2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \frac{\partial u_n}{\partial x_0} & \frac{\partial u_n}{\partial c_2} \end{bmatrix}, \; C_{X_v} = \begin{bmatrix} \sigma_{x_0}^2 & 0 \\ 0 & \sigma_{c_2}^2 \end{bmatrix} \tag{9}$$

The resulting projection of the equation (7) is shown in Figure 6. Taking into account the prior about the road geometry allows focalizing the zones of research in a Top-Down process fashion. Indeed, not all the image is used in order to perform the processing task. Consequently, the recognition process is faster, and subject to less noise considering it takes into account only the regions in the image that most likely contains a lane marking.

Once the probabilistic model is defined, we proceed to the recognition phase. This stage is inspired by [42]. For clarity's sake, we will discuss each aspect of this stage as presented in Figure 7.

i **Initialization** in this stage we define the probabilistic model (u', C'_u) as already presented in Figure 6.

ii **Selection of the best Region Of Interest (ROI)** once the probabilistic model is defined, we proceed to the selection of the most informative ROI in the Image in

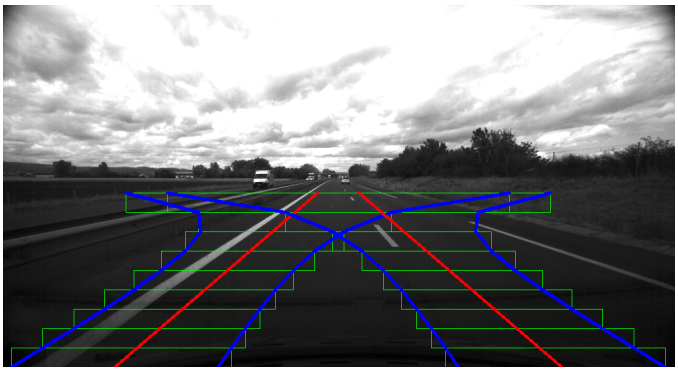

Fig. 6: Projection of the initial probabilistic model (u', C'_u) on the image. The mean values of the lane marking u' are presented in Red, Blue illustrates the dispersion around these values. Finally, green boxes are the ROI for each lane marking, in this example the number of ROI is 9 for each lane marking.

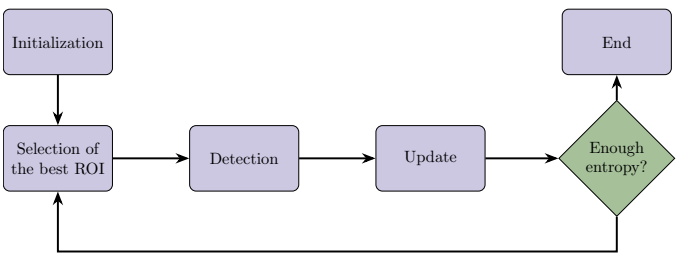

Fig. 7: All different steps of the Recognition stage for ego-lane marking detection.

a **Top-Down fashion** To do so, we present an informational criterion based on the Shannon entropy [43]. For each ROI, we use an *a priori* selection based on a entropic criterion H_{sel}:

$$H_{sel} \triangleq H_v^- - H_v^* \tag{10}$$
$$= p(Dk = 1, Db = 1).\frac{1}{2}\left[log|2\pi\, e\, C_{X_v}^-| - log|2\pi\, e\, C_{X_v}^*|\right]$$

with:

$$\begin{cases} X_v^- & \textit{The state vector before simulation of the detection,} \\ C_{X_v}^- & \textit{Covariance before simulation of the detection,} \\ X_v^* & \textit{The state vector after simulation of the detection,} \\ C_{X_v}^* & \textit{Covariance after simulation of the detection.} \end{cases}$$

For the probabilities, $p(D_b), p(D_k)$, a Bayesian Network (BN) is introduced. The presented nodes are an adaptation of the BN presented in [44]. The scheme of the network is illustrated in Figure 8. The BN is composed of the following nodes:

- X_{k_-} the confidence before the detection is attempted.
- Z_0 is the chosen landmark (the white strip) is observable in the ROI.
- D_k a landmark is detected in the focal zone,
- D_b the correct landmark has been detected (which manages landmark ambiguity).

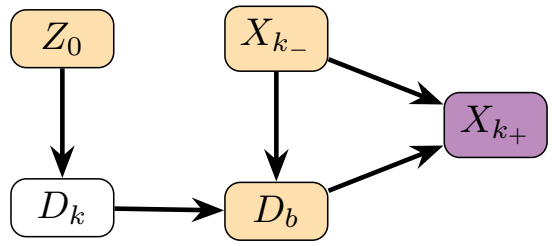

Fig. 8: Bayesian Network used for the confidence estimation. Yellow nodes are the input nodes, Purple node is the node we are seeking to estimate.

- X_{k_+} the confidence after the detection is attempted.

This network has two uses, the first one is to calculate the *a priori* probabilities $p(D_k)$ and $p(D_b)$, the second one is to determine the confidence $p(X_{k_+})$ obtained after proceeding to a detection.

iii **Detection** Once the most informative ROI is chosen, we proceed to the detection. The implemented method is based on a row filter as presented in [42]. The idea is to compute the gradient of the image row by row with the aim to find pixels that correspond to the lane marking. Once the patterns have been selected for the complete ROI, a Ransac method is used in order to detect the segment in the ROI. Thus, for each ROI two points $p_1(u_u, v_u)$ and $p_2(u_d, v_d)$ are defined.

iv **Update** for the state vector $x_p = (u_u, u_d)$ we associate a covariance matrix C_p :

$$C_p = \begin{pmatrix} \sigma_{u_u}^2 & 0 \\ 0 & \sigma^2 u_u \end{pmatrix} \tag{11}$$

With $\sigma_{u_u}^2$ being the accuracy of the segment detected. In order to update the probabilistic model, we use a Kalman filter:

$$\begin{cases} u^+ = u^- + K_u\left[\hat{x}_p - x_p\right] \\ C_u^+ = C_u^- - K_u H_u C_p \end{cases} \tag{12}$$

K_u being the Kalman Gain, (u^-, C_u^-) is the model before detection and (u^+, C_u^+) is the model after detection.

v **End** after each detection, we compute the corresponding entropic gain H_{gain}:

$$H_{gain} = p\left(X_{k_+}\right).\frac{1}{2}\left[log|2\pi e C_u^+| - log|2\pi e C_u^{'}|\right] \tag{13}$$

With $C_u^{'}$ the covariance from the initialization stage. The ego-lane marking has been detected if the following condition is satisfied :

$$H_{gain} \geqslant \lambda H_{max} \tag{14}$$

with

$$H_{max} = p\left(X_{k_+}\right).\frac{1}{2}\left[log|2\pi\, e\, C_u^+| - log|2\pi\, e\, C_u^{'}|\right]$$

With λ a fixed coefficient (< 1). If the condition is satisfied, it means that enough detections have been attempted successfully, and the obtained probabilistic model (u^+, C_u^+) is sufficiently precise to consider the ego-lane recognition finished. Otherwise, the ego-lane

recognition process carried out, and the second most informative ROI is selected.

The end of the marking detection involves the end of the ego-lane level localization. Indeed, the update of the model u, C_u leads to the update of the model X_v, C_{X_v}. This means that an estimation of the road parameters c_2, x_0 is performed.

3.3 Lane level localization (LLL)

Once the ego-lane localization is estimated, we have to perform the lane-level localization to correctly choose the right lane on which the vehicle travels. To do so, we proposed in [22] a probabilistic framework that is split into three stages. Thus, the presented LLL algorithm is an extension of our work proposed in [22]. Indeed, the Hidden Markov Model's architecture is the same. The difference lies in the input of the algorithm, as we take the information about the road from OSM. In addition, the parameters of the road, such as the curvature, are also extracted from OSM, which will be useful for interpolating other lanes marking.
In the first stage, we extrapolate adjacent lanes by assuming that lanes in the same road have the same width L_w and the variation in the curvature c_1 is very small. The second step consists of a Bayesian Network (BN), that takes as input the results of the hypothesized adjacent lane-marking detection, whether these detections succeed or fail. Furthermore, since the proposed BN is modular, we also use an adjacent vehicle detector based on deep learning. The third step includes a filtering process, using a Hidden Markov Model (HMM).

3.3.1 Adjacent lanes extrapolation

The adjacent lanes are extrapolated by taking advantage of the estimated ego-lane localization. Thus, each adjacent lane is described by a probabilistic model $\{X_l, C_{X_l}\}$, where X_l contains the parameters $(x_0, c_2)^T$ described in Section 3.2 and C_{X_l} refers to the corresponding covariance matrix. The assumption made is that the curvature and lane width stays constant for all the lanes in the same road. Thereby, the value of the vector X_l remains the same as x_v described in equation (6), only the value of x_0 will be shifted by a $(\pm i L_w)$, with $(-i)$ indicates that the edge is at the right of the ego-lane and $(+i)$ the left. The number of adjacent lanes extrapolated is equal to the lanes number from OSM. Thus, from a perspective view, the hypothesized lanes are shown in Figure 9.
As mentioned in the ELL section, the model $\{X_l, C_{X_l}\}$ can be transferred to the model image $\{u_l, C_{u_l}\}$. As a consequence, u_l represents the horizontal pixel of the edges in the image and C_{u_l} its interval confidence. In Figure 10, the adjacent lane regions of interest resulting from the extrapolation are shown.

For each adjacent lane marking extrapolated, the detection is performed and the results are fed into a Bayesian Network (BN).

3.3.2 Bayesian network for ego-lane determination

The proposed BN is designed to be flexible and modular for other detection results from any type of sensor, i.e. vehicle detector, guardrail detector. In order to show the flexibility of the BN, we will first use only adjacent lanes detection to determine the ego-lane. Thereafter, an adjacent vehicle detector based on Deep Learning (YOLO [45]) is introduced. The general architecture of the BN used is illustrated in Figure 11. The described nodes are the following:

- Z_{k_i} the element i is observable, this element can represent any element of the road scene: {vehicles, lane-marking, traffic signs...},
- D_{k_i} the detection of the element of type k is successful. Later we will use D_l for adjacent lanes and D_v for vehicles,
- L_{BN} the lane on which the ego-vehicle is traveling, L_{BN} is a form of categorical distribution, $L_{BN} \sim Cat(\xi)$ with $L_{BN} = \{l_1, ..., l_n\}$, where l_1 indicates the leftmost lane and n the lanes number.

Depending on the results of the detection, we infer the probability to belong to a lane l_i as follows:

$$P(L_{BN=l_i}) = P(L_{BN=l_i} | D_{k_1}, Z_{k_1}, ..., D_{k_n}, Z_{k_n}) \quad (15)$$

The lane with the highest probability is chosen to be the lane on which the vehicle travels. As mentioned before, the proposed BN is aimed to be modular for other detectors. To start, we only use the Adjacent Lanes Detection (ALD) as input into our BN (BN + ALD), where D_{l_i} indicates whether the detection of adjacent lane i is successful. After, we add the information about the adjacent vehicles D_{v_i} using the Yolo detector, where D_{v_i} indicates whether the detection of the adjacent vehicle i is successful. Knowing that, this detection is performed in the regions of the image bounded by the neighboring marking-lanes.

3.3.3 HMM for ego-lane determination

The proposed BN in the previous section is applied on a per-frame basis. However, the dynamic relationship between two consecutive frames is not taken into account. Indeed, the BN does not take into consideration the dynamic constraints of the ego-vehicle (i.e. the ego-vehicle can only change lane to an adjacent lane). In order to take into account these constraints, we filter the output of the BN by a Dynamic Bayesian Network which is the Hidden Markov Model (HMM).
We will use L_t to denote the set of ego-lane state variables at time t, which depends on the lanes number n_{lanes} and

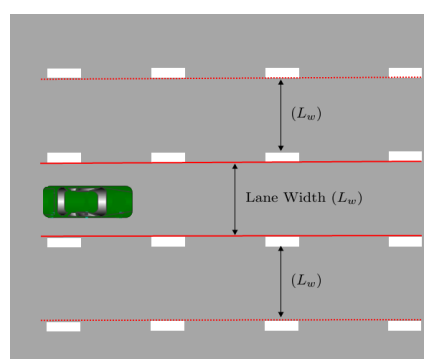

Fig. 9: Detected ego-lane in solid red and the hypothesized lanes in dash

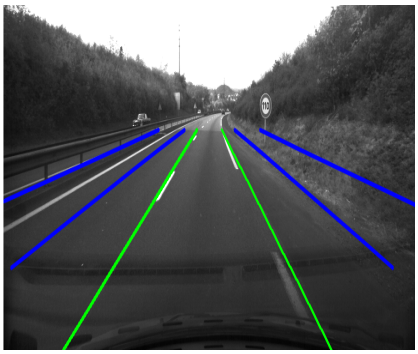

Fig. 10: Results of the adjacent lane regions of interest resulting from the extrapolation

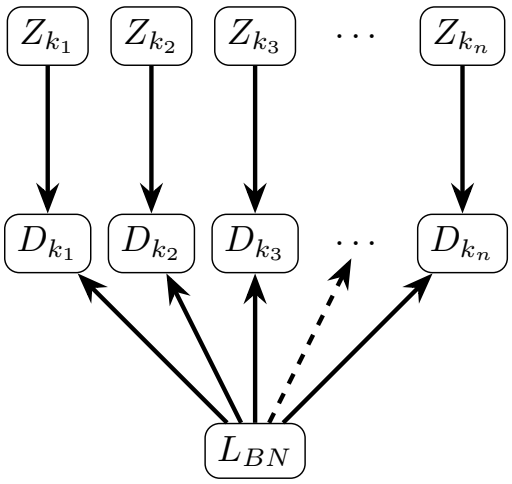

Fig. 11: Final architecture Bayesian Network (BN) used for ego-lane determination.

which are assumed to be observable. e_t denotes the observable evidence variable. The aim of the filter algorithm is to estimate the probability $P(L_{t+1}|e_{1:t+1})$. According to [46], this probability can be formulated as follows:

$$\begin{aligned} P(L_{t+1}|e_{1:t+1}) &= P(L_{t+1}|e_{1:t}, e_{t+1})_{\text{(dividing up the evidence)}} \\ &= \eta P(e_{t+1}|L_{t+1}, e_{1:t}) P(L_{t+1}|e_{t+1})_{\text{(Baye's rule)}} \\ &= \eta P(e_{t+1}|L_{t+1}) P(L_{t+1}|e_{t+1})_{\text{(Markov assumption)}} \end{aligned} \tag{16}$$

Where η denotes a normalizing constant to make probabilities sum equal to 1. By arranging Equation (16):

$$\begin{aligned} P(L_{t+1}|e_{1:t+1}) &= \eta P(e_{t+1}|L_{t+1}) \sum_{l_t} P(L_{t+1}|L_t, e_{1:t}) P(L_t, e_{1:t}) \\ &= \eta P(e_{t+1}|L_{t+1}) \sum_{l_t} P(L_{t+1}|L_t) P(L_t, e_{1:t}) \end{aligned} \tag{17}$$

The probability $P(e_{t+1}|L_{t+1})$ comes from the observation model. Hence, in this paper from the BN described previously, thus:

$$P(e_{t+1}|L_{t+1}) = P(L_{BN}) \tag{18}$$

With regard to the probability $P(L_{t+1}|l_t)$, it comes from the transition model. It expresses the probability of the ego-lane to change its current state, which is the lane change probability. Finally, the third term $P(L_t, e_{1:t})$ expresses the current state distribution. Graphically, we can illustrate the corresponding HMM as in Figure 12.

With the recursive formulation obtained in Equation (17),

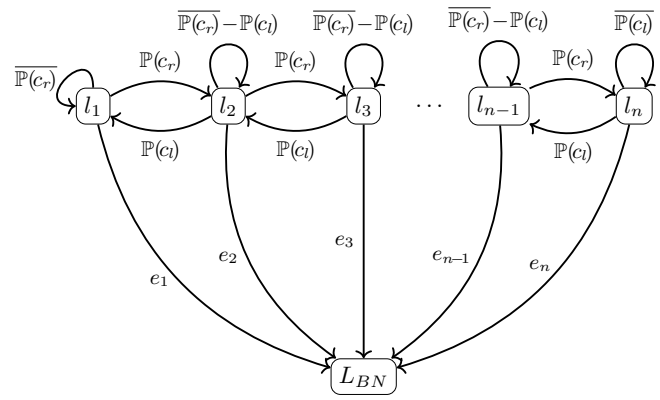

Fig. 12: Hidden Markov Model for n lanes case, with $L = \{l_1, l_2, ..., l_n\}$ being the set of hidden states and $O = \{e_1, e_2, ..., e_n\}$ being the set of observations resulting from the BN.

we can estimate the current ego-lane state given the observation obtained from the BN, but before we have to compute the lane change probability.

3.3.4 *Transition probability (Lane change probability)*

To calculate the lane change probability, we model the lateral position x_0 estimated in the ego-lane level localization as a normal distribution with mean μ_{x_0} and variance σ_{x_0}:

$$x_0 \sim \mathcal{N}(\mu_{x_0}, \sigma^2_{x_0}) \tag{19}$$

The value μ_{x_0} and variance σ_{x_0} are obtained from the estimated probabilistic model (x_v, C_{x_v}). Therefore, we predict the lateral position x_0 at time t_{k+1} as shown on Fig 13. Accordingly, the lane-change probability is calculated as

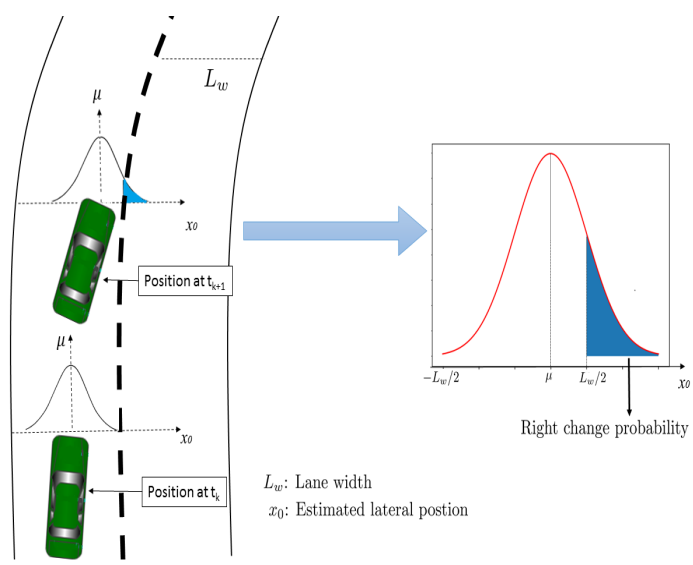

Fig. 13: Lane change probability, with the blue area representing the right change probability. The coordinate system for x_0 is centered on the vehicle

follows:

$$P(C_r) = \int_{L_w/2}^{+\infty} \frac{1}{\sigma_{x_0}\sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu_{x_0})^2}{2\sigma_{x_0}^2}} dx \quad (20)$$

$$P(C_l) = \int_{-\infty}^{-L_w/2} \frac{1}{\sigma_{x_0}\sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu_{x_0})^2}{2\sigma_{x_0}^2}} dx \quad (21)$$

with $P(C_r)$ the probability to right change lane and $P(C_l)$ the probability to left change lane.
Now that we have designed the HMM, we will introduce the real-world experimental results in the following section.

4 REAL-WORLD EXPERIMENTAL RESULTS

The different parts of the localization algorithm have been introduced in the previous section. In order to prove the effectiveness of the presented algorithms, real-world experiments have been carried out.
To do so, we tested our algorithm on real driving datasets. The first one was collected in the region of Clermont-Ferrand in France, where we drove our acquisition vehicle on two lanes and three lanes road in a national highway. In order to collect these datasets, the acquisition vehicle was equipped with a front camera, with the addition of an IMU and a classic GPS receiver. The data have been stored on a bag using using ROS. In addition to that, the ego-vehicle provides information about its own odometery. Finally, these datasets will be made available for researchers[1]. In the following, they will be referred to as "2-lanes" for two lanes road and "3-lanes" for three lanes road (1000 images frames).
Furthermore, we wanted to compare the performances of our algorithm to the literature on more challenging scenarios. Naturally, we turned our attention to the KITTI datasets [47]. Although these dataset contain few lane-changing scenarios, we will however use some part of those in order to show the effectiveness of our RLL module. In addition, we tested our algorithm on some datasets referred to in [19]. Unfortunately, all the data were not ready yet. We were able to test on some of them, noted as the A4-Highway Italy, for a total of 9528 frames. The collected datasets were manually annotated in a per-frame basis in order to determine the correct ego-lane classification. In the following, we will refer to these datasets as "4-lanes".
Each part of the ego-localization: RLL, ELL and LLL will be tested on the presented datasets, the results will be discussed in terms of accuracy. In the end, the overall algorithm will be discussed in terms of computation time.
In order to assess the presented RLL algorithm, the correct Way has been manually annotated for each GPS frame. The resulting ground truth was compared with HMM-MM. The results are summarized on Table 2. As highlighted, the Map-Matching with the addition of the HMM improves the overall accuracy precision. This can be explained by the inclusion of the topology of the road network in the HMM. Indeed, the transition states are governed by the connectivity between links. Furthermore, we test the reliability of our algorithm in more challenging scenarios. Thus, we used the well-known

1. shorturl.at/lAR28

	2-lanes	3-lanes	KIIT-1	KIIT-2
Number of frames	1136	205	107	76
Correct MM Without HMM	99.64 %	99.51 %	100 %	100 %
Correct MM with HMM	100 %	100 %	100 %	100 %

TABLE 2: Map Matching results on the entire datasets without and with the HMM, KITTI-1 and KITTI-2 refer to KITTI sequences:'_2011_09_26_0001' and '_2011_09_26_0002'

KITTI [47] database in urban scenarios. The used sequences are the following: "sequence_2001_09_26_0001" and "sequence_2001_09_26_0001". Even if the two sequences were taking in urban areas, the correct map-matching obtained shows the effectiveness and the robustness of our proposed RLL algorithm in varying scenarios. Finally, it can be noticed that for the "4-lanes", the GPS datasets have not been mentioned. Indeed, practically the entirety of the datasets is on the same Highway. Thus, the results on MM would not be relevant.
As discussed in [48] [13], in most of the work presented in the literature, the aim is to find the white strips in the image without seeking for consecutive elements of lane marking. Cnversely, in our work, we are more interested in the curvature of the lane marking than the white strips. However, by doing so, the characterization of the algorithm is considerably more difficult. Some examples are presented in Figure 14 showing the robustness of the presented ego-lane marking detector on different lighting conditions. Once the ego-lane localization completed, the Lane level localization is performed. So as to point out the increment of each added part, we first, determined the ego-lane using solely the BN with adjacent lanes. Within the second instance, we introduced adjacent vehicle detection in the BN. In all instances, we filtered the outcome of the BN with the HMM. All the results obtained are summarized in Table 3.
Considering the results, the increment provided by each module is clearly illustrated. Indeed, altogether cases the

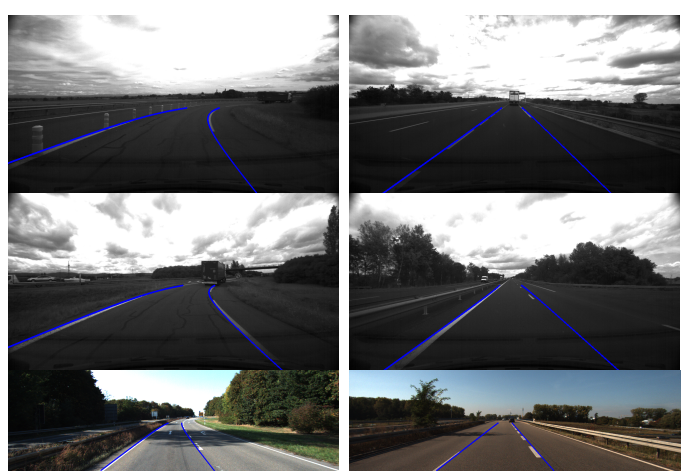

Fig. 14: Some examples of correct ego-lane marking recognition (in Blue). As highlighted, the ego-lane marking are correctly detected in varying imaging conditions: low light level for the top images and high brightness for the bottom images.

	$BN+ALD$	$BN+ALD+HMM$	$BN+ALD+AVD$	$BN+ALD+AVD+HMM$
2-lanes	91.89%	99.00%	93.60%	99.00%
3-lanes	78.8%	86.8%	81.30%	90.90%
4-lanes	67.36%	78.36%	74.67%	85.35%

TABLE 3: Classification accuracy for ego-lane determination. BN+ALD refers to the BN feed with the adjacent lanes detection, BN+ALD+HMM indicates the HMM with the corresponding BN, BN+ALD+AVD refers to the BN feed with adjacent lanes and vehicles detection and BN+ALD+AVD+HMM refers to the HMM with the corresponding BN.

BN+ALD+AVD provides more accurate classification than the BN+ALD, which suggests that the addition of another information source also will improve the obtained accuracy. After investigation on the inaccurate results, it appears that the false classifications obtained using the BN+ALD are due to two main reasons: either the lanes marking are not detected or the lanes marking are wrongly detected. For the primary case, this can be explained if the lanes marking are missing or hidden by an object. For the second case, it shows the limitation of the used lane marking detector. Moreover, even if the introduction of the adjacent vehicle detection shows excellent results, there are some cases where the vehicles detection are not relevant, for instance, if the detected vehicle is on the nearby road. To overcome this issue, we would have to determine the localization of the detected vehicle relative to the ego-vehicle, which is not the case since we use solely images as input.

Finally, despite that the authors[2] in [19] did not take into account the lane change scenarios and designed empirically the HMM. Nevertheless, we manage to outperform their results on the same datasets. Indeed, they achieved 77% correct classifications, where we were able to reach 85.35% on 9528 frames. Some ELL in different images are presented in Figure 15. The presented results have been carried using Python 3.7 under a Dell G3 3579 Core i7 8th generation equipped with an NVIDIA GeForce GTX 1050 Ti. The computation time results presented in Figure 16 shows that even if the entire algorithm was coded in Python, real-time implementation is possible. Indeed, the sum of all parts of the ego-localization algorithm is under $500\,\mathrm{ms}$ ($446.95\,\mathrm{ms}$ on average). In addition, if we glance in-depth at the time consumed in the RLL, we found that on average, it takes $238.16\,\mathrm{ms}$ to query the local server containing the OSM data. Furthermore, from the $143.67\,\mathrm{ms}$ dedicated to the LLL, $65.60\,\mathrm{ms}$ are spent on the YOLO detector. These results lead us to assume that implementation on C/C++ will divide the calculation time by 10.

5 Conclusion

In this paper, we presented an end-to-end ego-vehicle localization. Starting from Road-level localization (RLL) using OSM datasets and a classic GPS receiver, to the Ego-lane level localization (ELL) using a recognition lane marking in an information-driven fashion and finally, a Lane-level localization (LLL) using the well-known YOLO detector in a probabilistic framework composed of a Bayesian Network and Hidden Markov Model to correctly determine the ego-lane in the road. The distinction from other work lies in the overall aspect of the ego-localization that has been tackled in stock in the paper. In addition, this work is distinguished by the information-driven approach.

For our future work, we are currently working on adding detectors from different sensors, i.e., lidar, radar to enhance the proposed framework. On the other hand, we are still working on the OSM datasets in order to update this database when it provides false information about the environment.

2. The authors would like to acknowledge the authors of [19] for their help with their datasets.

6 Acknowledgment

This work has been sponsored by Sherpa Engineering and ANRT (Conventions Industrielles de Formation par la Recherche). This work has also been sponsored by the French government research program Investissements d'avenir through the RobotEx Equipment of Excellence (ANR-10-EQPX-44) and the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01), by the European Union through the program Regional competitiveness and employment 2014- 2020 (FEDER - AURA region) and by the AURA region.

References

[1] D. Iberraken, L. Adouane, and D. Denis, "Multi-level bayesian decision-making for safe and flexible autonomous navigation in highway environment," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3984–3990.

[2] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[3] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.

[4] G. P. Team, "Global positioning system (gps) standard positioning service (sps) performance analysis report," 2014.

[5] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation research part c: Emerging technologies*, vol. 15, no. 5, pp. 312–328, 2007.

[6] C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation research part c: emerging technologies*, vol. 8, no. 1-6, pp. 91–108, 2000.

[7] M. E. El Najjar and P. Bonnifait, "A road-matching method for precise vehicle localization using belief theory and kalman filtering," *Autonomous Robots*, vol. 19, no. 2, pp. 173–191, 2005.

[8] M. Bierlaire, J. Chen, and J. Newman, "A probabilistic map matching method for smartphone gps data," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 78–98, 2013.

[9] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 336–343.

[10] C. Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, "Online map-matching based on hidden markov model for real-time traffic sensing applications," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 776–781.

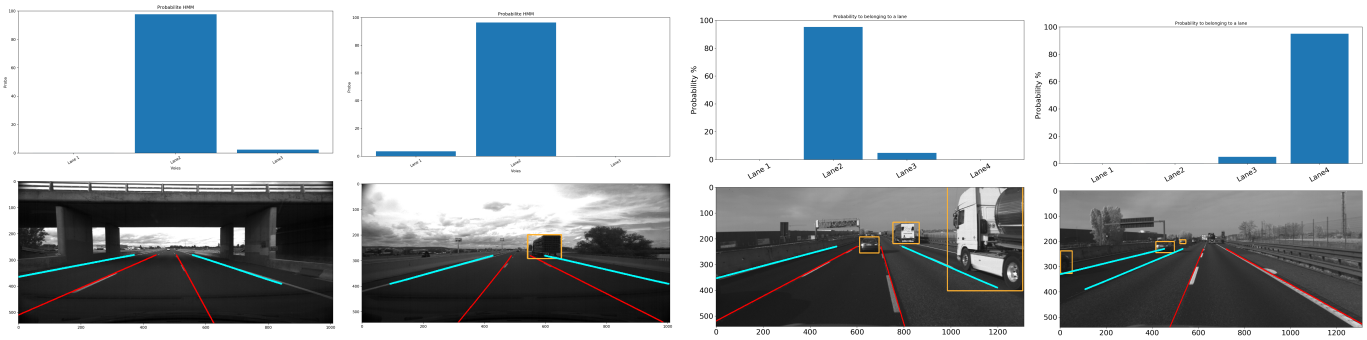

Fig. 15: Lane Level Localization on 3 lanes and 4 lanes scenarios. The orange boxes are the output of the Yolo detector. Cyan is used for adjacent lanes markings. Red is used for the ego-lane marking. A video is available at shorturl.at/lAR28.

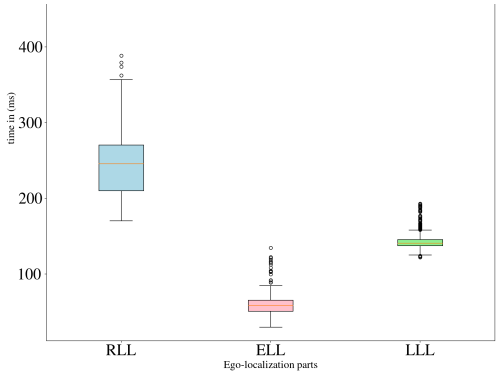

Fig. 16: Computation time in ms for each part of the presented algorithm: RLL in Light-blue, ELL in pink and LLL in light-green, with mean values of $244.53ms$ for the RLL, $58.74ms$ for the ELL and $143.68ms$ for the LLL. Since the computation time for the ELL and LLL depends on the observability conditions of the markings lanes in the image, the dispersion in time is much bigger than the RLL.

[11] M. M. Atia, A. R. Hilal, C. Stellings, E. Hartwell, J. Toonstra, W. B. Miners, and O. A. Basir, "A low-cost lane-determination system using gnss/imu fusion and hmm-based multistage map matching," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3027–3037, 2017.

[12] A. Kasmi, D. Denis, R. Aufrere, and R. Chapuis, "Map matching and lanes number estimation with openstreetmap," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2659–2664.

[13] J. Kim and C. Park, "End-to-end ego lane estimation based on sequential transfer learning for self-driving cars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 30–38.

[14] A. Borkar, M. Hayes, M. T. Smith, and S. Pankanti, "A layered approach to robust lane detection at night," in *2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*. IEEE, 2009, pp. 51–57.

[15] Y. Jiang, F. Gao, and G. Xu, "Computer vision-based multiple-lane detection on straight road and in a curve," in *2010 International Conference on Image Analysis and Signal Processing*. IEEE, 2010, pp. 114–117.

[16] M. Nieto, L. Salgado, F. Jaureguizar, and J. Arróspide, "Robust multiple lane road modeling based on perspective analysis," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 2396–2399.

[17] D. Svensson and J. Sörstedt, "Ego lane estimation using vehicle observations and map information," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 909–914.

[18] V. Popescu, R. Danescu, and S. Nedevschi, "On-road position estimation by probabilistic integration of visual cues," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 583–589.

[19] A. L. Ballardini, D. Cattaneo, R. Izquierdo, I. Parra, M. Sotelo, and D. G. Sorrenti, "Ego-lane estimation by modeling lanes and sensor failures," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–7.

[20] V. Popescu, M. Bace, and S. Nedevschi, "Lane identification and ego-vehicle accurate global positioning in intersections," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 870–875.

[21] S. Lee, S.-W. Kim, and S.-W. Seo, "Accurate ego-lane recognition utilizing multiple road characteristics in a bayesian network framework," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 543–548.

[22] A. Kasmi, D. Denis, R. Aufrere, and R. Chapuis, "Probabilistic framework for ego-lane determination," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1746–1752.

[23] F. Li, P. Bonnifait, J. Ibanez-Guzman, and C. Zinoune, "Lane-level map-matching with integrity on high-definition maps," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1176–1181.

[24] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue *et al.*, "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.

[25] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 690–703, 2016.

[26] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "Deeplanes: End-to-end lane position estimation using deep neural networksa," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 38–45.

[27] V. Milanés, D. F. Llorca, J. Villagrá, J. Pérez, C. Fernández, I. Parra, C. González, and M. A. Sotelo, "Intelligent automatic overtaking system using vision for vehicle detection," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3362–3373, 2012.

[28] H. Sawano and M. Okada, "A road extraction method by an active contour model with inertia and differential features," *IEICE transactions on information and systems*, vol. 89, no. 7, pp. 2257–2267, 2006.

[29] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using b-snake," *Image and Vision computing*, vol. 22, no. 4, pp. 269–280, 2004.

[30] D. Cáceres Hernández, L. Kurnianggoro, A. Filonenko, and K. Jo, "Real-time lane region detection using a combination of geometrical and image features," *Sensors*, vol. 16, no. 11, p. 1935, 2016.

[31] A. V. Nefian and G. R. Bradski, "Detection of drivable corridors for off-road autonomous navigation," in *2006 International Conference on Image Processing*. IEEE, 2006, pp. 3025–3028.

[32] M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 449–454.

[33] D. Gruyer, R. Belaroussi, and M. Revilloud, "Accurate lateral positioning from map data and road marking detection," *Expert Systems with Applications*, vol. 43, pp. 1–8, 2016.

[34] M. Revilloud, D. Gruyer, and M.-C. Rahal, "A new multi-agent approach for lane detection and tracking," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3147–3153.

[35] J. Al Hage, P. Xu, and P. Bonnifait, "Student's t information filter with adaptive degree of freedom for multi-sensor fusion," in *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.

[36] J. M. Alvarez, A. M. Lopez, T. Gevers, and F. Lumbreras, "Combining priors, appearance, and context for road detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1168–1178, 2014.

[37] F. Ramm, J. Topf, and S. Chilton, *OpenStreetMap: using and enhancing the free map of the world*. UIT Cambridge Cambridge, 2011.

[38] J. Kang, H. Kim, J. Park, and Y. Choi, "An enhanced map-matching algorithm for real-time position accuracy improvement with a low-cost gps receiver," *Sensors*, vol. 18, no. 11, p. 3836, 2018.

[39] A. Luo, S. Chen, and B. Xv, "Enhanced map-matching algorithm with a hidden markov model for mobile phone positioning," *ISPRS International Journal of Geo-Information*, vol. 6, no. 11, p. 327, 2017.

[40] A. Abramov, C. Bayer, C. Heller, and C. Loy, "A flexible modeling approach for robust multi-lane road estimation," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1386–1392.

[41] J. Xiao, L. Luo, Y. Yao, W. Zou, and R. Klette, "Lane detection based on road module and extended kalman filter," in *Pacific-Rim Symposium on Image and Video Technology*. Springer, 2017, pp. 382–395.

[42] R. Aufrere, R. Chapuis, and F. Chausse, "A model-driven approach for real-time road recognition," *Machine Vision and Applications*, vol. 13, no. 2, pp. 95–107, 2001.

[43] L. Delobel, R. Aufrère, C. Debain, R. Chapuis, and T. Chateau, "A real-time map refinement method using a multi-sensor localization framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1644–1658, 2018.

[44] C. Aynaud, C. Bernay-Angeletti, R. Chapuis, R. Aufrère, C. Debain, and N. Karam, "Real-time vehicle localization by using a top-down process," in *17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–6.

[45] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[46] S. J. Russell and P. Norvig, "Artificial intelligence-a modern approach (3rd internat. edn.)," 2010.

[47] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[48] M. Revilloud, D. Gruyer, and M.-C. Rahal, "A lane marker estimation method for improving lane detection," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 289–295.

Abderrahim Kasmi received the Master degree in Robotics from University of Montpellier in 2017, He integrates Sherpa Engeenering and he is currently pursuing a Ph.D degree at Institut Pascal, with a focus on vehicle localization fusion framework. His research interests involve autonomous vehicles, multi-sensor fusion, localization on digital map (OpenStreetMap), and vehicle perception.

Johann Laconte received the Master degree of Computer Sciences and Modelling as well as the Master degree in Robotics from Clermont Auvergne University, France, in 2018. He is currently a Ph.D student at Institut Pascal in Clermont-Ferrand, France, where his thesis aims to better quantify and manage the risk of robots displacements. His research interests include Lidar calibration, perception systems and risk modelling.

Romulad Aufrère received the Ph.D. degree in computer vision in 2001. In 2002, he carried out a post-doctoral training with the Robotics Institute, Carnegie Mellon University, USA. Since 2003, he has been teaching automatics, robotics, and signal processing with the ISIMA Engineering School of computer science. He is currently an Associate Professor with the Pascal Institute Laboratory, Blaise Pascal University, France. His research interests are in intelligent vehicles. He is involved in multi-sensor perception, multi-vehicle localization and active perception.

Dieumet DENIS received, in 2015, a PhD degree in Control of Mobile Robots from Institut Pascal – Université Blaise Pascal in Clermont-Ferrand, France. During his PhD, he largely studied the thematic of vehicles dynamics stability leading to a patent deposit for the design of an active safety and driving assistance device dedicated to off-road vehicles. Since 2016, after a post-doctoral position at CNRS/Irstea, Dr DENIS joined SHERPA Engineering as Project Manager and Research Engineer in Autonomous Vehicles and ADAS field. He is currently responsible for the SHERPA's R&D pole in Clermont-Ferrand.

Roland Chapuis received the PhD degree in 1991, Blaise Pascal Univ. France. From 1991 to 2002 he was an assistant professor, he became a full-Professor in 2002 in the Electrical department of Polytech Clermont-Ferrand engineering school of Blaise Pascal University in Clermont-Ferrand France. He was the head of "Perception Systems" team in Pascal Institute laboratory for several years. His research interests include intelligent vehicles, dynamic objects recognition and tracking, multisensorial perception systems, multimodal data driven fusion architectures, multimodal simultaneous localization and mapping.