Ego-Lane Estimation by Modeling Lanes and Sensor Failures

A. L. Ballardini[*], D. Cattaneo[*], R. Izquierdo[†], I. Parra[†], M. A. Sotelo [†], D. G. Sorrenti[*]

[*]Dept Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, Milano, Italy

[†]Dept of Computer Engineering, University of Alcalá, Alcalá de Henares, Spain

Abstract— In this paper we present a probabilistic lane-localization algorithm for highway-like scenarios designed to increase the accuracy of the vehicle localization estimate. The contribution relies on a Hidden Markov Model (HMM) with a transient failure model. The idea behind the proposed approach is to exploit the availability of OpenStreetMap road properties in order to reduce the localization uncertainties that would result from relying only on a noisy line detector, by leveraging consecutive, possibly incomplete, observations. The algorithm effectiveness is proven by employing a line detection algorithm and showing we could achieve a much more usable, *i.e.,* stable and reliable, lane-localization over more than 100Km of highway scenarios, recorded both in Italy and Spain. Moreover, as we could not find a suitable dataset for a quantitative comparison of our results with other approaches, we collected datasets and manually annotated the Ground Truth about the vehicle ego-lane. Such datasets are made publicly available for usage from the scientific community.

I. INTRODUCTION

Autonomous systems require an accurate understanding of the surrounding environment in order to safely plan their actions. For intelligent road vehicles one such fundamental perception task concerns the localization of the vehicle. Autonomous vehicles cannot always rely on global positioning systems based on Global Navigation Satellite System (GNSS) signals (*e.g.,* GPS) because they use to undergo multi-paths and physical barriers, leading sporadically to very poor position accuracy or even to no estimate at all. Therefore navigation modules usually couple the GNSS systems with cartographic maps and methods that leverage the road graph network as well as other common features [1]–[5], *e.g.,* buildings or roundabouts, which are retrieved from well-established cartographic services like OpenStreetMap. The maps represent an important piece of information that can be exploited as prior in the localization context. Even though these methodologies, usually known as *lock-on-road* procedures, see, *e.g.,* [6], [7], led to remarkable increases in the localization accuracy, they still do not use to achieve lane-level localization, *i.e.,* accuracies in the order of 0.1m [8]. One of the main disadvantages of today's mapping services is their coarse accuracy with respect to the road segments, *i.e.,* the alignment between the road graph and the satellite imagery is not usually reliable. Moreover, due to the collaborative nature of some of such services, together with a lack of automatic testing and validation procedures of the contributions, the accuracy is not consistent within the database.

Interesting approaches, originally proposed in the photogrammetry research field, try to solve this problem by

Fig. 1: Two frames from the proposed annotated dataset, with the overlaid detector output. In the first image, the vehicle was traveling in the A4 highway, Italy, performing a lane change. The second image depicts a frame from the A-2 highway, Spain.

means of satellite imagery parsing. In the robotics and computer vision fields, former attempts addressed only the extraction of the road areas [9], although the most interesting works also exploit mapping services. The authors in [10] propose to segment road regions leveraging aerial images and supervising the process using publicly available road vector data, but their approach is not aimed at updating the map service database. The authors in [11] explicitly propose to enhance the OpenStreetMap road graph by including information about road width and road segments center-lines. These enhancements are extremely valuable in the context of vehicle localization, since errors in road center-lines represent the most common source of inaccuracy when using features from map services as the main clues to perform localization. Recently, the same authors extended their work including both aerial and ground-level imagery [12], introducing a fine-grained road semantics that includes lanes, sidewalks and parking lots. Pursuing lane-level localization, the authors in [13] propose to exploit the objects present in the surrounding of the vehicle and to describe the probabilistic dependencies between the object measurements by means of a factor graph model. A similar proposal came from the authors of [14], where Histogram of Oriented Gradients are used to align the images acquired from a front facing camera to the road lane markings, to improve the

vehicle localization. Other works, aimed at enriching the maps with additional high-level features like *lanelets*, have been recently presented, *e.g.,* [15]. Here the authors introduce a novel specification for autonomous driving maps, which allows them to include also traffic regulation rules, known as tactical information.

In this paper, we present a probabilistic method aimed at enhancing the ego-lane estimation obtained from a simple line tracker. The tackled problem, also known as host-lane estimation, consists of the identification of the current lane occupied by a vehicle, given the number of lanes of the road, retrieved from a service like OpenStreetMap, and a GNSS prior.

Differently from other works available in the literature, here we present a modular, hence reusable, algorithm aimed at improving the lane-level localization that can be obtained from a generic line detector. The system relies on a Hidden Markov Model (HMM) with a transient failure model, which allows us to accommodate inaccurate or missing road marking detections.

The paper is organized as follows. Section II provides an overview of the existing ego-lane estimation literature, Section III describes the proposed algorithm and Section IV discusses the experimental configuration. Finally, Section V critically presents the experimental results of the system, followed by our concluding remarks.

II. RELATED WORK

Ego-lane estimation for autonomous driving has been extensively investigated in the last decades. The first achievements were obtained by the group of Prof. Dickmanns [16], exploiting a road representation model by means of clothoids, updated using Kalman filters. Starting from these remarkable results, an active research has been conducted in the successive years [17]–[20]. Heterogeneous modeling techniques including parabolas, clothoids, poly-lines or b-splines were proposed, typically computed from images, after some preprocessing phases designed to remove clutter and irrelevant areas.

One of the major challenges for these algorithms is the detection of the road surface. Achieving a good detection of the road surface is crucial since it is the basis for more complex tasks, but this detection is usually adversely affected by the large amount of clutter usually found on real roads. While faded road markings, unusual or specific weather conditions, or even light variations might severely affect the road surface detection, the visibility of the road surface is quite frequently hampered by the presence of other vehicles, thus requiring further considerations to solve the problem.

Most of the current Advanced driver-assistance systems (ADAS), like Lane Departure Warning (LDW) or Adaptive Cruise Control (ACC), require just a partial understanding of the whole observed scene, like vehicle's lane lines or lane crossing points in highway-like scenarios [21], [22].

For what sensing concerns, even though LIDAR-based algorithms sport the advantage of active lightening, vision-based algorithms represent today the most frequently used

approach for line detection and ego-lane estimation, since road marking are designed to be human-visible in mostly all driving conditions [21]. Many authors propose to increase the performance of ego-lane estimation algorithms with additional road information gathered by map services and with information provided by GNSS.

An interesting approach is presented in [23], where the authors tackled the ego-lane estimation as a scene-classification problem. They infer the lane number in a holistic fashion, leveraging both spatial information and objects around the vehicle, and finally training the best classifier with different learning algorithms. In [24] the author presented a robust lane-detection-and-tracking algorithm combining a particle filtering technique for lane tracking and RANSAC for the detection of lane boundaries. The work detects left and right lane boundaries separately, without exploiting fixed width lane models, and combining lane detection and tracking within a common probabilistic framework.

To deal with ego-lane estimation the authors in [25], [26], respectively in highway and urban scenarios, propose to exploit boosting classifiers and particle filtering approaches. A similar research was performed by [27], where multiple evidence from a visual processing pipeline was combined within a Bayesian Network approach.

Close to our proposal are the works in [28]–[30], where the authors specifically address the multiple-lane detection problem. In [28] multiple lane detections are performed after a first processing phase, in which the authors identify the ego-lane geometry. Then, adjacent lanes are first hypothesized and then tested, assuming same curvature and width for all lanes, a fair assumption for most of multi-lane roads, including highways. Similarly, the work proposed in [29] also considers highway scenarios and parallel lane markings, with respect to the detected ego-lane. More recently, the authors in [30] proposed a multi-lane detection algorithm based again on a hypothesis generation and verification scheme, ensuring an accurate geometric estimation by means of a robust line fitting pipeline and vanishing point estimation.

Differently from the other contributions, where the authors propose new detection pipelines for the ego-lane estimation problem, here we introduce a generic scheme for lane filtering, aimed at improving the ego-lane estimation capabilities of potentially every line detector. Also the output of a lane detection algorithm could be fed into our algorithm, to increase its performance in ego-lane estimation. Our aim is to enhance the localization capabilities of the scene understanding framework proposed in [7], introducing a lane-awareness module capable of reducing localization errors in highway-like environments. As a by-product, we can easily compare the localization results obtained with and without the new proposed algorithm.

III. PROPOSED ALGORITHM

Starting from a rough global localization as well as the detections of the road markings, the goal of the proposed algorithm is to estimate the vehicle ego-lane, to achieve in-lane localization accuracy in highway scenarios. The algo-
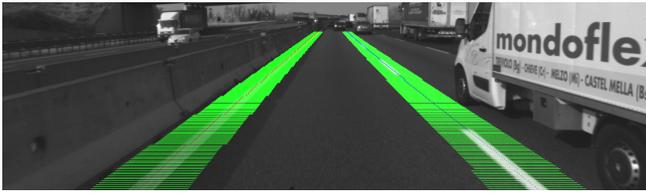
Fig. 2: An example of a moderately congested condition on the A4 (Turin - Milan - Venice - Trieste) highway, Italy. Even at this moderate level of congestion most road markings are hidden by traffic.

Fig. 3: Left: Only one of four lines are detected. Right: The highlighted lanes (two and three) have a higher probability of being the vehicle ego-lane, as the evaluation of the relative distance *w.r.t.* detected line implies.

rithm is designed to tolerate occasional temporary failures of a basic line detector as well as its noisy measurements. A line detector is a software component that detects and tracks the relative position of both dashed and continuous road lines, with respect to the vehicle.

On the one hand, the estimation of the vehicle ego-lane can be considered as a by-product of the line detection procedure. In fact, the position of all the lines within the road, relative to the vehicle, might allow us to evaluate the current lane using simple geometric considerations, on a per-frame basis. Unfortunately, line detections are usually not reliable enough, being hampered by faded road markings, cluttering elements from the nearby traffic or weather conditions, see *e.g.,* Figure 2 and 3a and 5. On the other hand, consider the situation depicted in Figure 3a, surely a critical situation for ego-lane estimation. Even though the exact lane cannot be estimated from the detected line, a distance measured from the detected line would allow us to limit the uncertainty only to the compatible lanes, as depicted in Figure 3b.

Our proposal is to tackle the ego-lane estimation with a probabilistic approach, in order to allow the system to infer the ego-lane by leveraging consecutive, yet incomplete, observations over time. We propose an HMM approach with n-lane states, corresponding to the number of lanes retrieved from an OpenStreetMap-like service.

A. Simple Line Detector and Tracker

In this section, we shortly describe the basic line detection and tracking algorithm used in this work. The pipeline leverages the images from an on-board stereo rig, with known calibration *w.r.t.* the vehicle reference frame. The algorithm consists of the following steps:

- The contours of the road markings are extracted from the Bird Eye / Inverse Perspective view (BEV / IPV) of
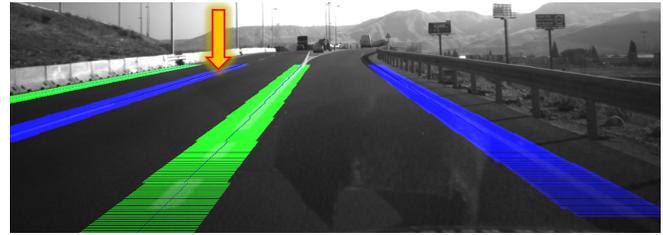

Fig. 4: Considering only the line indicated with the arrow, we can estimate the probability of being in Lane$\{1|2|3\}$ to be $\{0, 0.5, 0.5\}$. The procedure, repeated for all the detected lines as well as other detector reliability insights, allows us to tackle the in-lane localization problem. Here the green and blue lines visually suggest the reliability of the detected lines (green means higher).

Box 1: The line tracker output for the image in Figure 4. The isValid flag is set to TRUE when RI=10, using a hysteresis counting procedure. A negative offset indicates a line on the left of the vehicle.

Line1: isValid = 1; continuous=1; RI: 10; offset: -9.15m
Line2: isValid = 0; continuous=0; RI: 09; offset: -6.47m
Line3: isValid = 1; continuous=0; RI: 07; offset: -2.15m
Line4: isValid = 0; continuous=1; RI: 00; offset: +0.99m

the right camera image and discarded if their extension is below a threshold.

- Considering the detected contours, the algorithm tries to fit a fixed number of clothoids that include as many contour areas as possible, exploiting the stereo images to exclude points not being on the ground plane.
- The parameters of each clothoid model are then updated by means of a Kalman Filter.

With respect to the last 10 frames, a hysteresis counting procedure is used to track each line reliability. We refer to this information as *Reliability Index* (RI) in the following sections, and it allows us to set the isValid flag once the counter reaches its maximum value.

The basic line detector and tracker achieves good performances only under optimal illumination conditions and, as depicted in Figure 4 and shown in the corresponding results in Box 1, dashed lines and shadows are not always handled correctly.

However, the algorithm allows us to evaluate our contribution, which is designed to enhance the vehicle ego-lane estimation by exploiting a noisy sensor as well as the *road lane* properties of OpenStreetMap.

B. HMM with Transient Failure Model

To tackle the unavoidable problem of sensor failures, we applied a filtering algorithm based on the HMM proposal introduced in [31]. The proposed scheme allows us to take
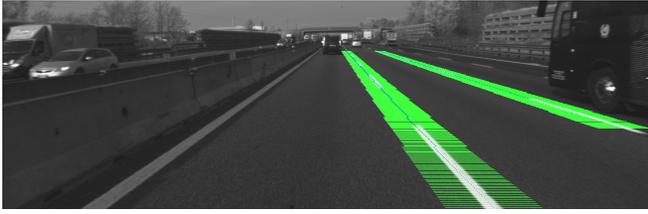
Fig. 5: In this figure, two out of five lines are correctly tracked. The shadow created by the Jersey barrier prevents the correct detection of the leftmost line. Comparable issues arise also with dashed lines if the space between two consecutive detected dashes is increased by the presence of other vehicles.

TABLE I: Basic Transition Matrix (BTM)

	$Lane_1$	$Lane_2$	$\cdots$	$Lane_n$
$Lane_1$	$\mathcal{N}(1,1,\sigma)$	$\mathcal{N}(2,1,\sigma)$	$\cdots$	$\mathcal{N}(n,1,\sigma)$
$Lane_2$	$\mathcal{N}(1,2,\sigma)$			
$\cdots$	$\cdots$			
$Lane_n$	$\mathcal{N}(1,n,\sigma)$			$\mathcal{N}(n,n,\sigma)$

$$STM_{2n \times 2n} = \left(\begin{array}{c|c} A_{n \times n} & B_{n \times n} \\ \hline C_{n \times n} & D_{n \times n} \end{array} \right) \qquad (7)$$

C. Counting Scheme

To evaluate the line tracker measurements, we derived an ad-hoc sensor model which exploits both the spatial and the RI information generated by the line tracker. The pipeline is composed as follows. First, we sort the lines in ascending order, considering their lateral offset *w.r.t.* the vehicle. Then a vector of counters, called *tentative*, of dimension n is created. The elements are populated by iterating the following considerations over all the valid lines, taking into consideration both dashed and continuous lines:

- we add 1 to all the *tentative* vector positions which are in accordance with the measurement, *i.e.,* compatible with the line,
- if the line has the continuous flag enabled, we add an additional *Bonus Value* (BV) associated to the *tentative* vector position (based on the distance *w.r.t.* the line).

As an example, considering the line indicated with an arrow in Figure 4, the resulting *tentative* vector corresponds to $[0; 1; 1]$ after the evaluation of the highlighted line. During the iteration phase, we also accumulate all the line-RI counters, which allow us to obtain an overall measure of the current detector reliability that we call $SensorOk$ (and its opposite $SensorBad = 1 - SensorOk$), according to Equation 8. The sensor matrix Z, designed to incorporate the new detector measure into our belief, consists of two separate parts called S_1 and S_2, which characterize the sensor model depending respectively on $SensorOk$ and $SensorBad$. In fact, while a reliable sensor measurement should be quickly integrated into the state, a failure of the sensor should not compromise the model estimate. Here the transient failure model is applied. S_1 is then calculated according to the *tentative* vector only, while S_2 introduces a certain amount of inertia, parameterized by the parameter w.

$$SensorOk = \frac{\sum_1^n isValid_i \cdot RI_i}{10 \cdot n} \qquad (8)$$

$$SensorBadState = tentative \cdot w + X_t \cdot (1 - w) \qquad (9)$$

$S_1 \rightarrow SensorOk \cdot tentative$
$S_2 \rightarrow SensorBad \cdot Sensor\_Bad\_State$

Finally, the sensor matrix Z consists in a $2n$ matrix composed as follows:

$$Z = \left(\begin{array}{c|c} S_1 & S_2 \end{array} \right) \qquad (10)$$

advantage of incomplete road line observations in a probabilistic fashion, exploiting the line RI to better estimate the current ego-lane. The HMM model implements a filtering procedure over a single discrete random variable, where each filtering iteration depends on the parameterization in Equation 1, see below for an explanation of the parameters.

$$HMM(n, \sigma_1, \sigma_2, P_1, P_2, BV, w) \qquad (1)$$

The vehicle state space X_t is defined using road lanes retrieved from the OpenStreetMap service. The belief over such state space represents the probability of being in one of the retrieved lanes, considering both the case of having a properly operating (Ok) or a faulty (Bad) line detector. These multiple cases are combined into a single "megavariable" [31], whose values are all the possible tuples of the individual state variables. The X_t state variable, see Equation 6, is created using the number n of lanes retrieved from the OpenStreetMap road that is currently being driven. Regarding the *State Transition Matrix* (STM), we used an approximated Gaussian transition model in which the appropriate probabilities are generated from the Basic Transition Matrix (BTM) shown in Table I, which needs to be extended in order to combine all the four cases of the "megavariable". The STM is shown in Equation 7, where parts A–D correspond respectively to:

A $\rightarrow$ lane transitions in SensorOK state,
B $\rightarrow$ lane transitions from SensorOk to SensorBad state,
C $\rightarrow$ lane transitions from SensorBad to SensorOk state,
D $\rightarrow$ lane transitions in SensorOK state.

Each part is instantiated using Equations 2–5, where P_1 and P_2 represent how likely the HMM will stay in each of the two sensor states (Ok or Bad) and $\sigma_{1,2}$ represent two different values used to generate the probability density function (PDF). Finally, the combination of the state X_t multiplied by the STM matrix gives the prediction state $\overline{X}_t$.

$$A = BTM(\sigma_1) \cdot (P_1) \qquad (2)$$
$$B = BTM(\sigma_1) \cdot (1 - P_1) \qquad (3)$$
$$C = BTM(\sigma_2) \cdot (1 - P_2) \qquad (4)$$
$$D = BTM(\sigma_2) \cdot (P_2) \qquad (5)$$

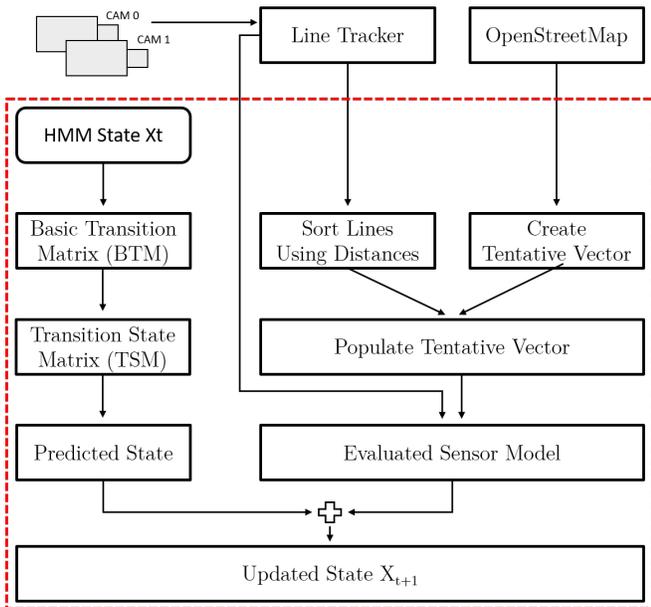$$X_t = \{Lane_{1..n}SensorOk; Lane_{1..n}SensorBad\} \qquad (6)$$

Fig. 6: The figure depicts the overall pipeline involved in the estimation of the vehicle's current lane. The blocks inside the dashed box represent actions of the proposed model.

The new lane positioning estimates are calculated combining the aforementioned prediction $\overline{X}_t$ multiplied by the Z matrix described so far into the new state vector X_{t+1}. Finally, to identify the ego-lane, we select the highest value from the X_{t+1} vector. The overall algorithm is depicted in Figure 6.

IV. EXPERIMENTAL CONFIGURATION

To effectively verify the improvements offered by our model, we collected two datasets in real driving conditions. The first dataset was recorded in the A4 highway, Italy, from Bergamo to Milan. The second dataset is from the A-2 highway area of Alcalá de Henares, Spain. Both the datasets were recorded at 10 fps and have a resolution of 1312x540 and 1392x400 pixels respectively. Differently from standard datasets like KITTI, in which the highway sequences only contain few lanes, we drove our vehicles on wider highways with 3 and 4 lanes (Spain and Italy respectively), including more than 100 lane changes in the A4 highway sequences. We manually created a ground truth (GT) by annotating the correct lane number on more than 20K frames, considering as $Lane_1$ the leftmost lane as in Figure 3. For each frame, we also included a flag that indicates whether the vehicle is actively crossing a lane, *i.e.,* moving to an adjacent lane (see Figure 1). For Equation 1, the parameter values used during the experimental section performed in Italy are as follows: $n = 4$, $\sigma_1 = \sigma_2 = 0.72$, $P_1 = 0.9$, $P_2 = 0.2$, $BV = 2$, $w = 0.6$. This parameterization was empirically defined after an initial optimization phase, aimed at identifying the best parameter configuration with respect to the GT. Believing that further research is required in the context, and to allow future researchers to compare their work with respect to ours, we published our datasets and the associated GT values

TABLE II: Ego-Lane Detection Dispersion

	Detectory Only	Our Model
Correct Lane	5276	6978
Offset 1	3744	2762
Offset 2	779	212
Offset 3	153	0

The table refers to the A4 Dataset, Seq #1, and reports the number of frames in which the lane algorithms correctly identified the vehicle lane position (first row). The three remaining rows show how the algorithms spread the misclassifications over the adjacent lanes. The Brief score associated to the detector is 0.293 while our model achieves 0.198. As the reader may notice, the metrics shows the better performances of our approach.

TABLE III: Line Detector Only

	1	2	3	4	Support	Recall
GT Lane 1	2230	320	21	3	2574	0.866
GT Lane 2	904	2005	275	16	3200	0.627
GT Lane 3	373	1666	927	5	2971	0.312
GT Lane 4	150	369	574	114	1207	0.094
Total	3657	4360	1797	138		
Precision	0.61	0.46	0.516	0.826		
F1 Score	0.7158	0.53	0.389	0.17		

Confusion Matrix, A4 Highway, using the line detector only.

online[1].

V. RESULTS

We evaluated the performances measuring the ego-lane estimates of the both algorithms (*i.e.,* the detector and the proposed model) in a per-frame basis, reporting whether correct lane classifications were achieved.

Figure 7 shows a short area of the A4 highway together with qualitative results of the algorithm performances, while in Table II we report the dispersion over the ego-lane detection, taking into account the full sequence length. As

[1]The dataset and the annotations are available on our lab's website: http://www.ira.disco.unimib.it/ego-lane-estimation-by-modeling-lanes-and-sensor-failures

TABLE IV: Line Detector + Our Model

	1	2	3	4	Support	Recall
GT Lane 1	2080	432	62	0	2574	0.808
GT Lane 2	246	2477	476	1	3200	0.774
GT Lane 3	13	871	2082	5	2971	0.701
GT Lane 4	0	136	732	339	1207	0.281
Total	2339	3916	3352	345		
Precision	0.889	0.633	0.621	0.983		
F1 Score	0.847	0.696	0.659	0.437		

Confusion Matrix, A4 Highway, using the proposed model.

TABLE V: Line Detector Only

	1	2	3	4	Support	Recall
GT Lane 1	2091	69	21	3	2574	0.957
GT Lane 2	704	1630	101	16	3200	0.665
GT Lane 3	280	1264	704	4	2971	0.313
GT Lane 4	80	267	424	113	1207	0.128
Total	3155	3230	1250	136		
Precision	0.663	0.505	0.563	0.831		
F1 Score	0.783	0.574	0.402	0.222		

Confusion Matrix, A4 Highway, using the line detector only, on frames not involving lane transitions.
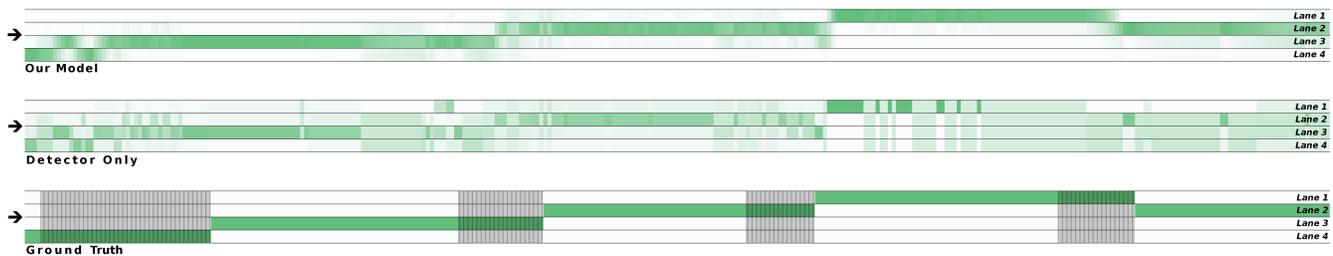
Fig. 7: A short section of the 4-lanes A4 highway in Italy. More saturated colors correspond to a higher probability of being in the specified lane. The figure depicts a comparison between our localization model (top) *w.r.t.* the results achieved using the detector only (middle). The bottom part of the graph corresponds to the ground truth, where the grayed part corresponds to the transition areas between lanes. As it can be seen, the approach yields good improvements with more stable detections over the detector's results.

TABLE VI: Line Detector + Our Model

	1	2	3	4	Support	Recall
GT Lane 1	1913	209	62	0	2574	0.876
GT Lane 2	199	2009	242	1	3200	0.82
GT Lane 3	0	491	1757	4	2971	0.78
GT Lane 4	0	113	470	301	1207	0.34
Total	2112	2822	2531	306		
Precision	0.906	0.712	0.694	0.984		
F1 Score	0.891	0.762	0.735	0.506		

Confusion Matrix, A4 Highway, using the proposed model, on frames not involving lane transitions.

TABLE VII: Line Detector Only - Spain dataset

	1	2	3	Support	Recall
GT Lane 1	2051	353	74	2478	0.828
GT Lane 2	550	3701	317	4568	0.81
GT Lane 3	93	1383	1230	2706	0.455
Total	2694	5437	1621		
Precision	0.761	0.681	0.759		
F1 Score	0.793	0.74	0.569		

Confusion Matrix, Spain dataset, using the line detector only.

TABLE VIII: Line Detector + Our Model - Spain dataset

	1	2	3	Support	Recall
GT Lane 1	2255	223	0	2478	0.91
GT Lane 2	345	3873	350	4568	0.848
GT Lane 3	43	306	2357	2706	0.871
Total	2643	4402	2707		
Precision	0.853	0.88	0.871		
F1 Score	0.881	0.864	0.871		

Confusion Matrix, Spain dataset, using the proposed model.

TABLE IX: Line Detector Only - Spain dataset

	1	2	3	Support	Recall
GT Lane 1	1740	162	52	2478	0.89
GT Lane 2	320	3213	63	4568	0.784
GT Lane 3	48	1225	1022	2706	0.596
Total	2108	4600	1137		
Precision	0.825	0.698	0.899		
F1 Score	0.857	0.784	0.596		

Confusion Matrix, Spain dataset, using the line detector only, on frames not involving lane transitions.

TABLE X: Line Detector + Our Model - Spain dataset

	1	2	3	Support	Recall
GT Lane 1	1860	94	0	2478	0.952
GT Lane 2	200	3323	73	4568	0.924
GT Lane 3	0	254	2041	2706	0.889
Total	2060	3671	2114		
Precision	0.903	0.905	0.965		
F1 Score	0.927	0.915	0.926		

Confusion Matrix, Spain dataset, using the proposed model, on frames not involving lane transitions.

readers may observe, here our naive line detector shows its limitations. On the one hand, and not surprisingly, the results show that the line detector alone is unable to correctly detect the correct ego-lane, mostly because of missing detections due to clutter or illumination issues. As depicted in the middle part of Figure 7 (*i.e.,* detector only segment), the detector results are extremely noisy, resulting in unreliable ego-lane detections. For instance, the detector is completely missing the final transition from $Lane_1$ to $Lane_2$, leaving the vehicle without almost any in-lane localization clue. On the other hand, the filtering effect of the HMM model is clearly shown in the upper part of the same image. Here the proposed model correctly identified the lane transitions even without a complete set of line measurements, and promising results are summarized in Figure 8. From a technical perspective, our model outperformed the basic detector in all our tests. In the confusion matrices Tables III to X, we report the most relevant information we used to assess our algorithm performances. It is worth noting that, regarding the dataset recorded in Spain, both the algorithms achieve better performances. This is most likely related to the better view of the whole road in front of the vehicle, which contains 3 lanes instead of 4. Finally, with respect to the experimental activity and the results, it is clear that with a slightly better line detector would result in a great improvement.

CONCLUSIONS AND FUTURE WORKS

We presented an ego-lane estimation algorithm aimed at enhancing the accuracy of the vehicle localization in highway scenarios. Differently from other works, we have also proposed a reusable optimization designed to cooperate
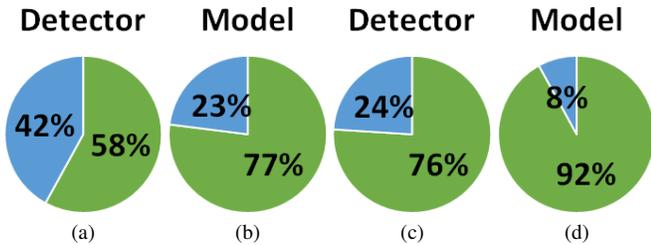
Fig. 8: Comparison graphs between the localization accuracies using our proposal *w.r.t.* using the detector only, in both Italy (4-lanes highway, Figures 8a and 8b) and Spain (3-lanes highway, Figures 8c and 8d). In all figures, green represents correct detections and blue wrong detections.

with existing line detectors. With respect to the existing ego-lane estimation literature, our algorithm achieves good localization even when fed with noisy and/or occasionally missing data, *i.e.,* the typical output of a real and therefore faulty line detector. We exploited an HMM-based scheme to take advantage of real road line observations in a probabilistic fashion. The proposed algorithm allows us to improve the localization robustness in treacherous conditions, where lane markings are missing or are hidden by traffic clutter and/or lightening issues. As part of our future works, we are currently working on introducing a seamless transition system between different lane configuration scenarios, by leveraging our previous contributions in the context of vehicle localization. This integration would allow us to tackle the possibility of errors due to GNSS to OpenStreetMap mismatches, as well as incorrect map information. Finally, we are currently analyzing the results and limitations of the proposed algorithm in urban-like scenarios.

REFERENCES

[1] P. Ruchti, B. Steder, M. Ruhnke, and W. Burgard, "Localization on OpenStreetMap data using a 3D laser scanner," in *2015 IEEE Int. Conf. Robot. Autom.* IEEE, may 2015, pp. 5260–5265.

[2] M. Raaijmakers and M. E. Bouzouraa, "In-vehicle Roundabout Perception Supported by A Priori Map Data," in *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2015-Octob, 2015, pp. 437–443.

[3] A. Jose M., L. Antonio M., G. Theo, and L. Felipe, "Combining priors, appearance, and context for road detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1168–1178, 2014.

[4] A. L. Ballardini, D. Cattaneo, S. Fontana, and D. G. Sorrenti, "Leveraging the OSM building data to enhance the localization of an urban vehicle," in *2016 IEEE 19th Int. Conf. Intell. Transp. Syst.* IEEE, nov 2016, pp. 622–628.

[5] A. L. Ballardini, D. Cattaneo, S. Fontana, and D. G. Sorrenti, "An online probabilistic road intersection detector," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 239–246.

[6] I. P. Alonso, D. F. Llorca, M. Gavilan, S. Á. Pardo, M. Á. Garcia-Garrido, L. Vlacic, and M. Á. Sotelo, "Accurate global localization using visual odometry and digital maps on urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1535–1545, 2012.

[7] A. L. Ballardini, S. Fontana, A. Furlan, D. Limongi, and D. G. Sorrenti, "A Framework for Outdoor Urban Environment Estimation," in *2015 IEEE 18th Int. Conf. Intell. Transp. Syst.* IEEE, sep 2015, pp. 2721–2727.

[8] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intell. Veh. Symp.* Ieee, jun 2011, pp. 163–168.

[9] F.-m. YE, L. SU, and J.-l. TANG, "Automatic Road Extraction Using Particle Filters from High Resolution Images," *J. China Univ. Min. Technol.*, vol. 16, no. 4, pp. 490–493, dec 2006.

[10] J. Yuan and A. M. Cheriyadat, "Road Segmentation in Aerial Images by Exploiting Road Vector Data," in *2013 Fourth Int. Conf. Comput. Geospatial Res. Appl.* IEEE, jul 2013, pp. 16–23.

[11] G. Mattyus, S. Wang, S. Fidler, and R. Urtasun, "Enhancing Road Maps by Parsing Aerial Images Around the World," in *2015 IEEE Int. Conf. Comput. Vis.*, vol. 11-18-Dece. IEEE, dec 2015, pp. 1689–1697.

[12] S. Wang, S. Fidler, and R. Urtasun, "HD Maps : Fine-grained Road Segmentation by Parsing Ground and Aerial Images," in *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3611–3619.

[13] F. Kuhnt, J. M. Zöllner, F. Kuhnt, S. Orf, S. Klemm, and J. Z. Marius, "Lane-precise localization of intelligent vehicles using the surrounding object constellation," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. November, 2016.

[14] G. Cao, F. Damerow, B. Flade, M. Helmling, and J. Eggert, "Camera to map alignment for accurate low-cost lane-level scene interpretation," in *Proc. IEEE Intell. Transp. Syst. Conf.*, no. November, 2016.

[15] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *IEEE Intell. Veh. Symp. Proc.*, no. Iv, 2014, pp. 420–425.

[16] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-D Road and Relative Ego-State Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, 1992.

[17] D. Pomerleau, "RALPH: rapidly adapting lateral position handler," in *Proc. Intell. Veh. '95 Symp.*, 1995, pp. 506–511.

[18] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 62–81, 1998.

[19] C. J. Taylor, J. Malik, and J. Weber, "A real-time approach to stereopsis and lane-finding," in *IEEE Intell. Veh. Symp. Proc.*, 1996, pp. 207–212.

[20] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-Snake," *Image Vis. Comput.*, vol. 22, no. 4, pp. 269–280, 2004.

[21] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, apr 2014.

[22] J. McCall and M. Trivedi, "Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, mar 2006.

[23] T. Gao and H. Aghajan, "Self lane assignment using egocentric smart mobile camera for intelligent GPS navigation," in *2009 IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2009*, 2009, pp. 57–62.

[24] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, 2008.

[25] T. Kuhnl, F. Kummert, and J. Fritsch, "Visual ego-vehicle lane assignment using Spatial Ray features," in *IEEE Intell. Veh. Symp.*, no. Iv, 2013, pp. 1101–1106.

[26] J. Rabe, M. Necker, and C. Stiller, "Ego-lane estimation for lane-level navigation in urban scenarios," in *2016 IEEE Intell. Veh. Symp.*, vol. 2016-Augus, no. Iv. IEEE, jun 2016, pp. 896–901.

[27] S. Lee, S. W. Kim, and S. W. Seo, "Accurate ego-lane recognition utilizing multiple road characteristics in a Bayesian network framework," in *IEEE Intell. Veh. Symp.*, vol. 2015-Augus, no. Iv, 2015, pp. 543–548.

[28] M. Nieto, L. Salgado, F. Jaureguizar, and J. Arróspide, "Robust multiple lane road modeling based on perspective analysis," in *Int. Conf. Image Process. ICIP*, 2008, pp. 2396–2399.

[29] Y. Jiang, F. Gao, and G. Xu, "Computer vision-based multiple-lane detection on straight road and in a curve," in *Int. Conf. Image Anal. Signal Process.*, vol. 2, 2010, pp. 114–117.

[30] S.-N. Kang, S. Lee, J. Hur, and S.-W. Seo, "Multi-lane detection based on accurate geometric lane estimation in highway scenarios," in *IEEE Intell. Veh. Symp.*, no. Iv. IEEE, jun 2014, pp. 221–226.

[31] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, sect. 15.5*, 2nd ed. Pearson Education, 2003.