# Lane marking aided vehicle localization

Zui Tao, Philippe Bonnifait, Vincent Fremont, Javier Ibañez-Guzman

# Lane marking aided vehicle localization

Z. Tao[1,2], Ph. Bonnifait[1,2], V. Frémont[1,2], J. Ibañez-Guzman[3]

***Abstract*— A localization system that exploits L1-GPS estimates, vehicle data, and features from a video camera as well as lane markings embedded in digital navigation maps is presented. A sensitivity analysis of the detected lane markings is proposed in order to quantify both the lateral and longitudinal errors caused by 2D-world hypothesis violation. From this, a camera observation model for vehicle localization is proposed. The paper presents also a method to build a map of the lane markings in a first stage. The solver is based on dynamical Kalman filtering with a two-stage map-matching process which is described in details. This is a software-based solution using existing automotive components. Experimental results in urban conditions demonstrate an significant increase in the positioning quality.**

## I. Introduction

Currently, there is much interest on vision-based and map-aided vehicle localization due to the availability of detailed digital navigation maps. Vehicle location tasks can be classified into three task level scales according to the required accuracy: Macroscale, Microscale and Mesoscale [1]. Different driving functions need different scales. Macroscale needs an accuracy of 10 m and interests resides on tasks that can be made using digital road network data. At the mesoscale, the required resolution accuracy is at the decimeter level; this is equivalent to lane-level accuracy for a digital map [1]. Different research endeavours exist on building lane-level maps [2], [3]. Having accurate maps allows their use for contextualization and localization for intelligent vehicle applications.

From an autonomous navigation perspective, microscale tasks are concerned with lane-keeping or parking functions, as well as detecting and avoiding obstacles. For this, most applications use computer vision and lidars and do not require map objects to be geo-referenced. To position in real-time a vehicle with decimeter-level accuracy for a microscale task is a major challenge. The sole use of differential GNSS receivers (e.g. GPS) cannot guarantee the required accuracy during the whole task, particularly if GNSS outages exist. Several sensor fusion approaches have been explored. These rely often on the use of GPS, Inertial Measurement Unit (IMU), vehicle odometry and digital maps [4]-[5].

Lane-level digital maps represent a lane by its centre-line [2], [3]. These are built using a vehicle with a high accuracy localization, typically a RTK type GPS receiver plus a high end IMU. Each lane is surveyed by driving as close as possible to its centre, resulting in some errors. Another approach is to map features on the road with a mobile mapping vehicle equipped with a high accuracy localization system and calibrated vision sensors [6].

All roads have different types of lane markings. These are painted in a symmetrical manner and following national standards [7]. Therefore, the vehicle does not need to follow the centre-line of the road during mobile mapping. Lane markings have similar properties with road centre-lines both are clothoids. They can be simplified to polylines having a limit number of shape points. In this paper, we exploited these characteristics to embed lane level information into digital maps and their extraction from video images [8], [9]. Both are applied to improve the localization function.

The remainder of the paper is organized as follows: Section II introduces the system model for lane marking measurements including a camera model that takes into account for vertical motion of the camera used to extract lane markings. Section III describes the mobile mapping process for building a digital map of the lane markings that is to be used as source of lateral constraints in the localization estimates. Section IV describes the map aided localization solver. Section V includes results from a series of road experiments using a passenger vehicle. Finally, Section VI concludes the paper.

## II. Lane marking measurements

### A. Frames

The system exploits GPS information expressed in global geographic coordinates. All estimations are made in Cartesian coordinates with respect to a local navigation frame $R_O$ tangent to the Earth surface. Defined with its x-axis pointing to the East, y-axis to the North and z-axis oriented upwards with respect to the WGS84 ellipsoid. Its origin can be any GPS position fix close to the navigation area. As this point can be chosen arbitrarily close to the navigation area, the trajectory can be considered as planar in 2D. Two more frames as shown in Fig. 1 are needed. $R_M$ denotes the vehicle reference frame ($x_M$ is the longitudinal axis pointing forward and $y_M$ is oriented such that $z_M$ points upwards). Point C represents the origin of the camera frame $R_C$ located at the front of the vehicle. To stay consistent with the vision system conventions, $y_C$ is right-hand oriented. Even if the camera is located behind the windscreen with a position offset $(C_x; C_y)$, every detected lane marking is expressed w.r.t. $R_C$.

Let denote ${}^{M}T_{C}$ the homogeneous transformation matrix mapping $R_C$ in $R_M$ and ${}^{O}T_{M}$ the one mapping $R_M$ in $R_O$.

The authors are with [1]*Université de Technologie de Compiègne (UTC)*, [2]*CNRS Heudiasyc UMR* 7253, [3]*Renault S.A.S, France.*

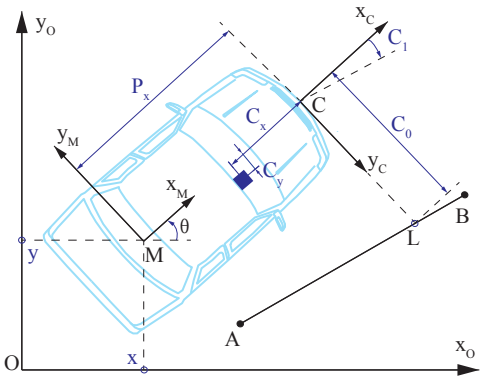


Fig. 1. Reference frames. [A; B] is a detected lane marking segment.

### B. Intrinsic and extrinsic parameters sensitivity analysis

In practice, the camera is attached to the body of the vehicle which is itself a suspended mass. So, the 2D-world hypothesis can be easily violated. However, when a camera is correctly installed behind the windshield, the camera roll and yaw angles are very small and can be easily compensated by calibration and image warping. But when the vehicle crosses a speed bumper, both the altitude and the tilt angle of the camera frame undergo changes (see Fig. 2). The same issue appears when the slope of the road changes or when the vehicle accelerates or breaks. Moreover, the height of the camera also depends on the load of the vehicle. Finally, some calibration errors related to the camera intrinsic parameters may occur.

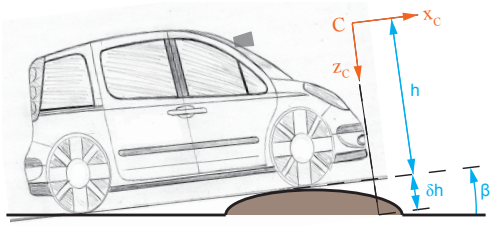


Fig. 2. Parameter variation while climbing a speed bumper

So, it is crucial to evaluate the sensitivity of the parameters involved in the development of lane marking measures particularly the tilt angle and the height w.r.t. the road.

*1) Nominal model with the camera parallel to the road:* In the camera frame $R_C$, the equation of a detected lane marking can be written as (by considering a first order Taylor's expansion of a clothoid in the camera frame [8]):

$$\begin{cases} y = C_1 \cdot x + C_0 \\ z = h \end{cases} \tag{1}$$

Where $C_0$ and $C_1$ are respectively the lateral distance and the heading of the detected lane marking and $h$ is the height of the camera w.r.t. the road.

In the image plane, using perspective projection, we have:

$$\begin{cases} u = f \cdot \frac{y}{x} + u_0 \\ v = f \cdot \frac{z}{x} + v_0 \end{cases} \tag{2}$$

with $f$ the focal length, and $[u_0, v_0]^T$ the coordinates of the principal point.

By plugging Eq. 1 into Eq. 2, we can get:

$$\frac{u - u_0}{f} = C_1 + \frac{C_0}{x} \tag{3}$$

and

$$x = \frac{h \cdot f}{v - v_0} \tag{4}$$

By plugging Eq. 3 into Eq. 4, we get:

$$u - \frac{C_0}{h} \cdot v + \frac{C_0 \cdot v_0}{h} - u_0 - C_1 \cdot f = 0 \tag{5}$$

The equation of the detected lane marking [8], [9], expressed in image plane is:

$$a_p \cdot u + b_p \cdot v + c_p = 0 \tag{6}$$

where $a_p$, $b_p$ and $c_p$ are the line coefficients. By comparing Eq. 5 and Eq. 6, we can get:

$$\begin{cases} C_0 = -\frac{h \cdot b_p}{a_p} \\ C_1 = -\frac{a_p \cdot u_0 + b_p \cdot v_0 + c_p}{a_p \cdot f} \end{cases} \tag{7}$$

*2) The camera is tilted downward:* Let suppose that the camera is tilted downward (or upward) with an angle $\beta$.

In the camera frame $R_C$, the equation of a detected lane marking can be written as:

$$\begin{cases} y = C_1 \cdot (x \cdot cos\beta - z \cdot sin\beta) + C_0 \\ x \cdot sin\beta + z \cdot cos\beta = h \end{cases} \tag{8}$$

We can get:

$$\begin{cases} C_0 = -\frac{h \cdot b_p \cdot f + (a_p \cdot u_0 + b_p \cdot v_0 + c_p - f \cdot b_p \cdot tan\beta) \cdot h \cdot sin\beta}{a_p \cdot f \cdot cos\beta} \\ C_1 = -\frac{a_p \cdot u_0 + b_p \cdot v_0 + c_p - f \cdot b_p \cdot tan\beta}{a_p \cdot f} \end{cases} \tag{9}$$

*3) Sensitivity analysis:* In this subsection, $\beta$, $f$ , $h$ , $C_0$ and $C_1$ stand for the nominal values of these five parameters, and $\beta + \delta\beta$, $f + \delta f$ , $h + \delta h$ , $C_0 + \delta C_0$ and $C_1 + \delta C_1$ are their true values. We suppose that the range of $C_1$ is $[-0.3, 0.3]$ rad and the range of $C_0$ is $[-127, 128]$ meters. Let study the sensitivity to $\beta$, $f$ and $h$.

- Sensitivity to $\beta$

$$\begin{cases} \frac{\delta C_0}{C_0} = \frac{a_p \cdot C_1 \cdot [tan(\beta + \delta\beta) - tan\beta] - b_p \cdot [sec(\beta + \delta\beta) - sec(\beta)]}{a_p \cdot C_1 \cdot tan\beta - b_p \cdot sec\beta} \\ \frac{\delta C_1}{C_1} = \frac{b_p \cdot [tan(\beta + \delta\beta) - tan\beta]}{a_p C_1} \end{cases} \tag{10}$$

Let suppose that a straight line in the image plane meets: $\frac{b_p}{a_p} = -1$. In order to see the sensitivity clearly, an example is given in Table I.

| $\frac{\delta\beta}{\beta}$ | 0.4 | 1 | 2 |
|---|---|---|---|
| $\frac{\delta C_0}{C_0}$ | 0.014 | 0.037 | 0.083 |
| $\frac{\delta C_1}{C_1}$ | 0.118 | 0.296 | 0.602 |

TABLE I

SENSITIVITY ANALYSIS WITH $h = 1.5\,m$, $C_1 = 0.3\,rad$, $\beta = 5\,degree$

As shown by Table I, $C_1$ is very sensitive to an error in $\beta$, but $C_0$ is not.

- Sensitivity to $f$

$$\begin{cases} \frac{\delta C_0}{C_0} = \frac{b_p \cdot tan\beta \cdot sin\beta - C_1 \cdot a_p}{b_p - C_1 \cdot a_p \cdot sin\beta} \cdot \frac{\delta f}{f+\delta f} \\ \frac{\delta C_1}{C_1} = \frac{b_p \cdot tan\beta - a_p \cdot C_1}{a_p \cdot C_1} \cdot \frac{\delta f}{f+\delta f} \end{cases} \quad (11)$$

Taking the relationship $\frac{b_p}{a_p} = -1$, and the ranges of $C_0$ and $\beta$ into consideration, one can easily find that $C_0$ is not sensitive to an error in $f$, but $C_1$ could be very sensitive to it.

- Sensitivity to $h$

$$\begin{cases} \frac{\delta C_0}{C_0} = -\frac{\delta h}{h} \\ \frac{\delta C_1}{C_1} = 0 \end{cases} \quad (12)$$

In this case, the error on parameter $C_0$ is directly related to the height variation.

As a conclusion:

- a $C_0$ error is proportional to an error in $h$ but $\delta h$ has a variation limited to few centimeters in normal driving conditions;
- $C_1$ is sensitive to errors on $f$ and $\beta$. The influence of the intrinsic parameters of the camera is limited but the tilt angle is a crucial parameter even in normal driving conditions.

So, we believe that $C_0$ is an accurate measure while $C_1$ should be used with caution. A way to exploit it accurately could be to use an IMU providing an estimate of $\beta$ or to estimate vanishing points using vision [10].

### C. Camera measurement

As mentioned previously, the measurement used for vehicle localization is the lateral distance $C_0$ (see Fig. 1).

In Fig. 1, [A; B] is the detected lane marking segment which has been extracted from a digital map by a map matching process. In $R_O$, the equation of segment [A; B] is:

$$y = a \cdot x + b$$

In $R_C$, the homogeneous coordinates of point $L$ (see Fig. 1) are ${}^C L = [0, C_0, 1]^T$. So, ${}^O L = {}^O T_M \, {}^M T_C \cdot {}^C L$ are the homogeneous coordinates of point $L$ in frame $R_O$:

$${}^O L = \begin{bmatrix} P_x \cdot \cos\theta + C_0 \cdot \sin\theta + x \\ P_x \cdot \sin\theta - C_0 \cdot \cos\theta + y \\ 1 \end{bmatrix} \quad (13)$$

where $P_x$ is a translation from point $M$ to the bumper.

Because point $L$ is on the segment [A; B], ${}^O L$ verifies the equation of the line $y = a \cdot x + b$ which gives:

$$C_0 = \frac{P_x \cdot \sin\theta - a \cdot P_x \cdot \cos\theta - a \cdot x + y - b}{a \cdot \sin\theta + \cos\theta} \quad (14)$$

The singularity due to a division by zero means that the detected line is perpendicular to the vehicle which can never occur in practice.

## III. Building a map of lane markings

The localization solver can be assisted by providing a lateral constraint; this can be made using an accurate digital map that includes geo-localized lane markings. For real-time purposes, a limited number of parameters must represent the map line segments.

By definition, mobile mapping is the process of collecting geospatial data from a vehicle, typically fitted with a range of video cameras, radars, lasers or any number of remote sensing systems. All sensors data are acquired and time-stamped, the results are fused and road features are extracted. These are localized accurately mainly using a high-precision localization system (i.e. RTK-GPS, IMU, vehicle odometry). In this paper, we collect lane features using a driving assistance series video camera that provides attributes of the clothoides detected when collecting data.

In the following, we describe the main steps that are involved in the map building stage. Please note that every lane of a carriageway can have two lane markings.

### A. Referencing detected lane markings

As the mobile mapping is done in post-processing, it is quite easy to synchronize the positioning data with the camera measurements by interpolating the trajectory after having compensated the clock drift of the acquisition system with respect to the GPS time. This allows geo-referencing every detected lane marking point $L$ in $R_O$ by using Eq.13.

### B. Clustering points by lanes

In this paper, we propose to fit the lane marking points into polylines instead of higher order curves (which could fit lane markings better) since:

- the camera observation model described in section II-C is based on line segments;
- polylines need less parameters than high-order curves, so it is more efficient.

Points are graphically regrouped by polylines. Keeping in the map every mapped point can provide a huge amount of data which will not be efficient for real-time navigation particularly in the map-matching stage. Therefore, the obtained clusters of marking points are simplified. In the following, we describe the key stages to do this simplification.

### C. Polylines segmentation

We need to find the shape points of the simplified polylines. In other words, we need to identify the segments. Between every two adjacent shape points, there is a segment, with a certain bearing, which stands for a length of lane marking. The Douglas–Peucker's algorithm [11] is a well known algorithm for reducing the number of points in a curve that is approximated by a series of points. It's used here to find the shape points which divide the lane marking into parts with different headings. In this stage, the accuracy is controlled by choosing suitable tolerance which is the maximal euclidean distance allowed between the simplified segments and the origin points.

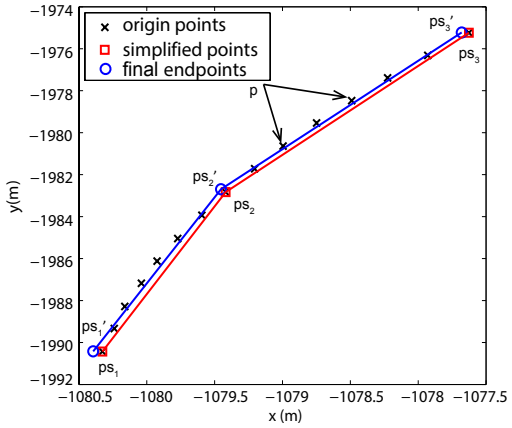


Fig. 3. Illustration of the 2 stages lane marking fitting. The tolerance is 20 centimeters.

Let consider the example described in Fig. 3. $p$ is a set of lane marking points. By using Douglas–Peucker's algorithm, one can obtain shaping points $ps_1$, $ps_2$ and $ps_3$ which are three points chosen from $p$. For every point between $ps_1$ and $ps_2$, their Euclidean distances to segment $[ps_1; ps_2]$ are smaller than the tolerance, the same stands for the points between $ps_2$ and $ps_3$.

If we choose a very small tolerance, the segments in a polyline can be too short, and there can be too many shape points in the polyline. For this reason, we propose to choose a tolerance of decimeter-level.

### D. Improving map accuracy

In order to further improve the accuracy of the polylines by reducing the errors effects due to the segmentation process, a least-squares algorithm is performed with every point between two adjacent shape points of the segmented polyline. For instance, in Fig. 3, one can notice that every point between $ps_1$ and $ps_2$ is above segment $[ps_1; ps_2]$.

Let consider a polyline with $n$ points. After $n-1$ least-squares stages, $n-1$ new straight lines are formed (blue lines in Fig. 3). It is then necessary to retrieve the nodes (endpoints) and shape points of the new polyline. There are two cases. The intersection of two successive lines define a new shape point (e.g. $ps_2^{'}$). By convention, the new nodes are defined as the orthogonal projection of the previous node on the new line (e.g $ps_1^{'}$). To come back to the example, the new polyline is $\left[ps_1^{'}; ps_2^{'}; ps_3^{'}\right]$ at the end of the process. It is the result that is finally stored in the digital map.

The algorithm 1 resumes this process.

### E. Modifying the clustering stage

In order to perform the Douglas–Peucker's algorithm [12], it should be noticed that the set of points has to be linked by a function (i.e. each abscissa value must have only one image). It means that the points in the same cluster have to be arranged in a way that the x-coordinates are monotonically increasing or decreasing. Therefore, lane markings sets have to be divided into subsets when this condition is not verified. The Fig. 4 gives an example.

As a conclusion, the clustering described in subsection III-B is modified as follows:

**Algorithm 1** Polyline fitting algorithm

Input: set of points $p$ in the working frame $R_0$
Output: shape points $ps^{'}$ of the polyline

1) $\left(ps_1, \ \ldots \ , ps_n\right) = dpsimplify(p, tolerance)$[12]
2) **For** i = 2 to n
3) $\quad line_{i-1}$ ←Least_squares_fitting(points between $ps_{i-1}$ and $ps_i$)
4) $\quad$ **If** i≠n
5) $\quad\quad ps_i^{'}$ ←intersection of $line_{i-1}$ and $line_i$
6) $\quad$ **End if**
7) **End for**
8) $ps_1^{'}$ ←orthogonal projection of $ps_1$ on $line_1$
9) $ps_n^{'}$ ←orthogonal projection of $ps_n$ on $line_{n-1}$
10) **Return** $\left(ps_1^{'}, \ \ldots \ , ps_n^{'}\right)$

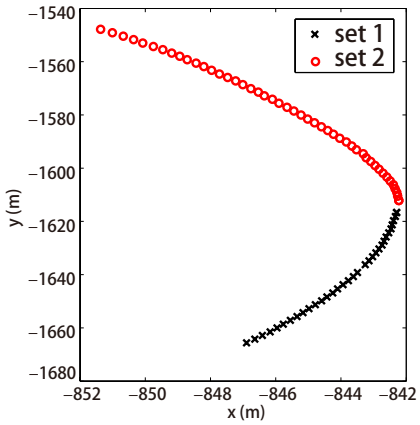


Fig. 4. Example where consecutive points of the same lane marking are divided into two clusters

- the points in one set have to physically belong to the same lane marking;
- the points in one set should meet the requirement of a function because of the Douglas–Peucker's algorithm;
- the points in one set should be close to each other. Indeed, when doing the collection of lane marking points, the camera can miss some lane markings, especially when the vehicle is making a turn. So, the distance between two following points in one set can be high and the resulting segment between them can be far from the real lane marking.

In practice, in order to do a good quality mapping, the lane marking points are divided into suitable different clusters under human supervision.

Please note that some close lane marking detections may belong to two sides of the same physical marking. In this case, we fit both edges of the lane marking. This can occur when the marking is located between two adjacent lanes (See Fig. 5). As an example: when the vehicle is traveling on lane 1, it detects segment 2; when traveling on lane 2, it detects segment 3; these two segments are two edges of the same lane marking located between lane 1 and lane 2.

### F. Normal angle of lane marking segments

Each lane can have two border lines, both of them mapped into two polylines. Fig. 5 gives an example. The four blue

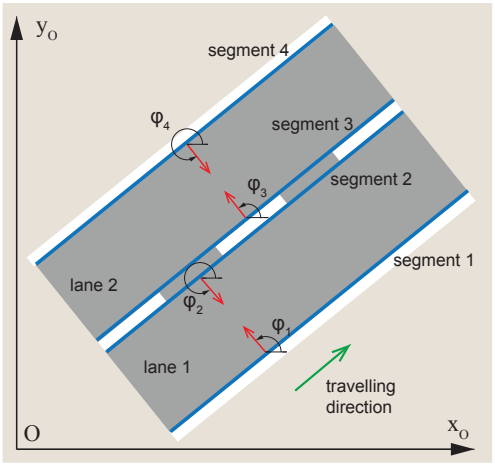


Fig. 5. Lane marking map of a two-lanes roadway

straight-line segments are part of the lane marking map. Please remind that both sides of the central lane marking are mapped. Segment 1 and 2 are two border lines of lane 1; segment 3 and 4 are two border lines of lane 2.

In order to improve the map-matching process when the vehicle is traveling, a normal vector is defined for every segment. By convention, it is perpendicular to the segment and it points to the lane center-line (See red arrows in Fig. 5). In practice, this vector can be managed by an angle denoted $\varphi$ which is by convention defined w.r.t. the x-axis of frame $R_O$ with a range $[0, 2\pi]$.

## IV. Map-aided Localization

In this section, we present the key stages of the localization system that uses L1-GPS loosely coupled with proprioceptive sensors and lane marking detections map-matching with a previously built map.

### A. Localization solver

Fig. 6 shows the structure of the localization solver. $\widehat{X}$ is the estimated state, and $P$ is the estimated covariance matrix. The inputs are:

- proprioceptive data, i.e. linear and angular velocities provided by the ABS wheel speed sensors and the ESP yaw rate gyro made accessible through a CAN bus gateway,
- L1-GPS (mono-frequency receiver with C/A pseudo-ranges) with PPS for time synchronization and latency compensation,
- camera observations $C_0$,
- map of the lane markings obtained by using the method described in section III.

The data fusion is realized by using an Extended Kalman Filter (EKF) with innovation gates, a well-known technique that consists to reject observations that are away from the prediction from a statistical point of view.

A classical unicycle model is used to predict the vehicle pose. GPS errors (biases) are also estimated by the localization solver. Let $[x_{gps}, y_{gps}]^T$ denote the measurements

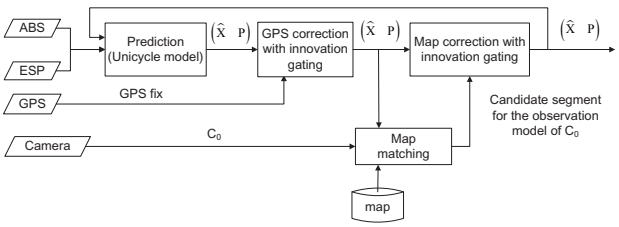


Fig. 6. Diagram of the localization solver

provided by the GPS receiver. The used observation model is

$$\begin{cases} x_{gps} = x + \varepsilon_x \\ y_{gps} = y + \varepsilon_y \end{cases} \tag{15}$$

where $\varepsilon_x$ and $\varepsilon_y$ are the colored errors of GPS fixes that are added to the Kalman state vector $X = [x, y, \theta, \varepsilon_x, \varepsilon_y]^T$.

The camera observation model has been described in section II-C. Hereafter, we detail the map-matching process and an internal integrity test used to reject miss-matches if they occur.

### B. Map-matching

Generally, map-matching algorithms integrate positioning data with spatial road network data (roadway centre-lines) to identify the correct link on which a vehicle is traveling and to determine the location of a vehicle on a link [13].

When the vehicle is traveling on a lane with lane markings, the camera is supposed to be able to detect one or both border lines of the lane. Map-matching here is to determine which are the segments in the built lane marking map that the camera have detected. It works by sequentially applying each stage described in the following.

*1) Lane side selection:* This is the first step of the process. The idea is to select only the polylines that are on the correct side. Conceptually, it works as if every candidate solution is located inside the lane in order to consider only angles. Let take Fig. 5 as an example. The vehicle is traveling following the green arrow direction but we don't know in which lane.

With the normal angle of the lane marking segment defined in subsection III-F, one can calculate on which side of vehicle the segment is located:

$$side = \begin{cases} left, & -\pi \leq \theta - \varphi \leq 0 \\ & or \\ right, & \pi \leq \theta - \varphi \leq 2\pi \\ & otherwise \end{cases} \tag{16}$$

where the range of $\theta$ (vehicle heading angle) is $[0, 2\pi]$.

The detected lane markings are easily localized on left or right:

$$side = \begin{cases} left, & C_0 \leq 0 \\ right, & C_0 > 0 \end{cases} \tag{17}$$

Only when the calculated side by Eq. 16 is the same as the lane side provided by $C_0$, the segment can be chosen as a candidate.

Let suppose that the camera gives a measurement $C_0 < 0$, which means a lane marking detection on the left side. Only


Fig. 7. Experimental vehicle

segment 2 and 4 can be chosen as the candidate segments. This guarantees that the vehicle always lies in the region of the road after a correction of camera observation.

*2) Matching degree of candidate segments:* To do the selection among the candidates, for each candidate segment, a matching degree $D = D\left(C_0, \widehat{C_0}\right)$ is defined based on the Mahalanobis metric using only the lateral distance (since the heading is sensitive to errors as shown before):

$$D = \left(C_0 - \widehat{C_0}\right)^T \left(HPH^T + R\right)^{-1} \left(C_0 - \widehat{C_0}\right) \quad (18)$$

where:

- $\widehat{C_0}$ is the predicted camera measurement using the camera observation model and $\widehat{X}$ after the GPS correction;
- $R$ is the covariance of the observation noise;
- $H$ is the Jacobian matrix of the camera observation model at $\widehat{X}$ after the GPS correction.

The most probable segment is selected by the minimization of $D$.

*3) Innovation gating:* The Mahalanobis metric defined in subsection IV-B.2 is also used for integrity check. Only when the Mahalanobis distance is less than a threshold, camera measurements can be regarded as a good lane marking detection, and the observation $C_0$ can be used to correct the estimated state by GPS and IMU. This integrity test is useful to exclude wrong lane marking detections and mismatches.

## V. Results

Outdoor experiments have been carried out in the city of Compiègne (France). Fig. 7 shows the experimental vehicle. It should be noticed that the weather was cloudy.

### A. Experimental set-up

The experimental vehicle was equipped with a NovAtel RTK-GPS receiver coupled with a SPAN-CPT IMU running at 10Hz. The system received RTCM 3.0 corrections through a 3G connection from a GPS base station Septentrio PolaRx2e@ equipped with a Zephyr Geodetic antenna. This high accuracy system (few tens of centimeter) was used during the mobile mapping process. It also provided ground truth data for the localization method. The station was located at the research center in Compiègne. It was the origin $O$ of the local coordinate system (latitude $49.4°$, longitude $2.796°$, ellipsoidal height 83 m). A CAN-bus gateway was used to access to wheel speed sensors (WSS) and to a yaw rate gyro. A Mobileye camera was used to detect the lane markings, on the basis of lane detection function of Mobileye's Lane Keeping and Guidance Assistance System (LKA). The output is described in Mobileye's LKA common CAN protocol. The low-cost GPS was a U-blox 4T with a patch antenna with no EGNOS correction.

### B. Mobile mapping results

We performed two paths on the same road (one-way double lane). The length of the path is 3736 m. The vehicle passed three crossroads and a fork road. During the first path, lane marking points have been collected to make the digital map. The second path is used to test the localization solver. Fig. 8 shows the obtained lane marking map (blue lines).

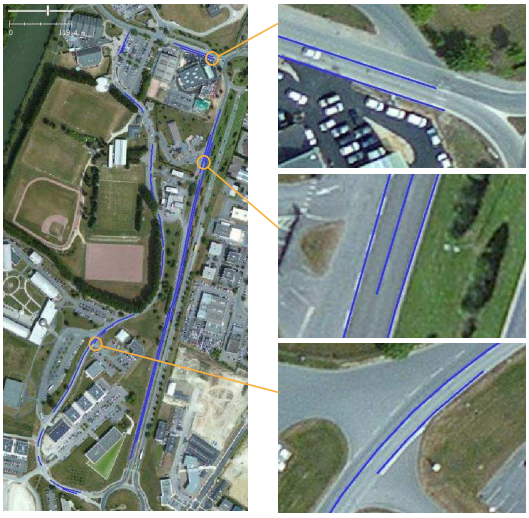

Fig. 8. Map of a test site superimposed on an Open Street Maps image

One can notice that the blue lines are quite consistent with the lane markings on the road. Though not all the lane markings are mapped (only one path is used for mobile mapping and the camera also missed some lane markings, especially at the crossroads), this map is enough to test the performance of the localization system proposed in section IV.

### C. Localization results

The localization solver described in section IV has been tested by using data replay.

Fig. 9 shows changes of localization errors on $x$ and $y$ over time, plotted with $\pm 3\sigma$ bounds (red lines) to show how the filter is well tuned.

Fig. 10 shows lateral and longitudinal positioning errors. Three methods are compared: L1-GPS (green lines), IMU coupled with L1-GPS (red lines) and map-aided localization (black lines). The blue/green points indicate there are good lane marking detection on left/right side at the moment. One can see that, first, the data fusion of GPS with CAN

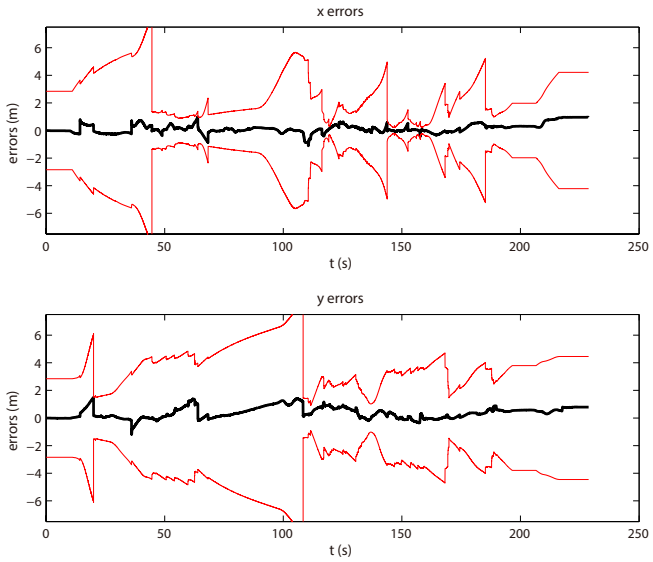


Fig. 9. Positioning errors on $x$ and $y$

sensor enhances the accuracy. Second, the use of the camera with the mapped lane marking increases significantly the performance.

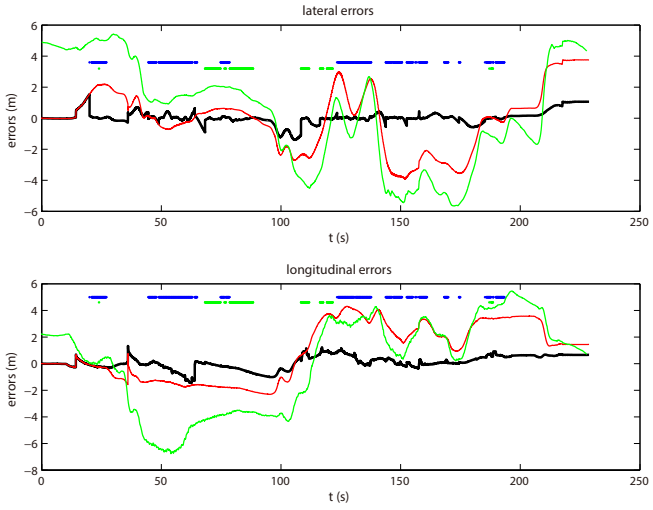


Fig. 10. Lateral and longitudinal positioning errors

Table II gives performance metrics of the localization solver. Logically, lateral accuracy is highly improved since the camera observation $C_0$ is relative to the lateral distance between the vehicle and the detected lane marking. One can notice that longitudinal accuracy is also improved by the camera system due to heading variations of the path. Camera observations are available 45.9% of the time. 95% horizontal positioning error is less than 1.25 m.

## VI. Conclusion

A localization solver based on the use of road lane markings has been presented. It is based on information stored on road lanes in the form of polylines in an a priori map. The uniqueness of our solution is that it exploits information from detected lane markings from a driving assistance series

| | Horizontal PE (m) | | Lateral PE (m) | | Longitudinal PE (m) | |
|---|---|---|---|---|---|---|
| | I | II | I | II | I | II |
| mean | 2.74 | 0.54 | 1.49 | 0.26 | 1.96 | 0.39 |
| std. dev. | 1.23 | 0.39 | 1.26 | 0.34 | 1.16 | 0.39 |
| max | 4.85 | 1.56 | 3.92 | 1.56 | 4.31 | 1.46 |
| median | 3.23 | 0.53 | 1.09 | 0.11 | 1.74 | 0.36 |
| 95th percentile | 4.23 | 1.25 | 3.73 | 1.06 | 3.76 | 0.94 |

TABLE II

Error statistics. (PE: positioning error; I: results without map; II: results with map)

video camera. This information is matched to the information stored in the navigation maps and then used to assist the estimations of a L1-GPS receiver together with vehicle proprioceptive data. The importance of height variations of the video camera due to road profiles has been demonstrated and so measured camera relative heading angles have to be considered carefully. So, we recommend only the use of lateral distance. Results in real traffic conditions have shown the high potential of the proposed approach. Future work resides on enhancing results when approaching tight curves when road markings are difficult to extract. Envisaged applications are on actuating advanced driving assistance and autonomous driving.

## References


[1] J. Du, J. Masters, and M. Barth, "Lane-level positioning for in-vehicle navigation and automated vehicle location (avl) systems," *Proc IEEE Int. Transp. Systems*, pp. 35–40, 2004.

[2] D. Betaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *IEEE Transactions on Int. Transp. Systems*, vol. 11, pp. 786–798, 2010.

[3] A. Chen, A. Ramanandan, and J. A. Farrell, "High-precision lane-level road map building for vehicle navigation," *Position Location and Navigation Symp.*, pp. 1035–1042, 2010.

[4] R. Toledo-Moreo, D. Betaille, F. Peyret, and J. Laneurit, "Fusing gnss, dead-reckoning, and enhanced maps for road vehicle lane-level navigation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 798–809, 2009.

[5] M. Hentschel and B. Wagner, "Autonomous robot navigation based on openstreetmap geodata," Madeira Island, Portugal, 2010, pp. 1645–1650.

[6] Mattern, N.Schubert, R.Wanielik, and G.Fac, "High-accurate vehicle localization using digital maps and coherency images," *Int. Vehicles Symp.*, 2010.

[7] (2012) Maunual on uniform traffic control devices. [Online]. Available: http://mutcd.fhwa.dot.gov/pdfs/2009r1r2/mutcd2009r1r2edition.pdf

[8] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," *Int. Vehicles Symp.*, pp. 109–114, 1994.

[9] K. Chris, L. Sridhar, and K. Karl, "A driver warning system based on the lois lane detection algorithm," *Int. Vehicles Symp.*, vol. 1, pp. 17–22, 1998.

[10] J.-C. Bazin and M. Pollefeys, "3-line ransac for orthogonal vanishing point detection," *IEEE Conf. on Int. Robots and Systems*, pp. 4282–4287, 2012.

[11] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[12] W. Schwanghart. (2010) Recursive douglas-peucker polyline simplification. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/21132-line-simplification/content/dpsimplify.m

[13] M. A. Quddus, W. Y. O. b, and R. B. N. b, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, pp. 312–328, 2007.