



**HAL**  
open science

# Detection of Missing Roundabouts in Maps for Driving Assistance Systems

Clément Zinoune, Philippe Bonnifait, Javier Ibañez-Guzmán

► **To cite this version:**

Clément Zinoune, Philippe Bonnifait, Javier Ibañez-Guzmán. Detection of Missing Roundabouts in Maps for Driving Assistance Systems. IEEE Intelligent Vehicles Symposium (IV 2012), Jun 2012, Alcala de Henares, Spain. pp.123-128, 10.1109/IVS.2012.6232245 . hal-00709919

**HAL Id: hal-00709919**

**<https://hal.science/hal-00709919>**

Submitted on 19 Jun 2014

**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.

# Detection of Missing Roundabouts in Maps for Driving Assistance Systems

Clément Zinoune<sup>1,2</sup>, Philippe Bonnifait<sup>1</sup>, Javier Ibañez-Guzmán<sup>2</sup>

**Abstract**—Passenger vehicles are evolving into sensor-based computer controlled platforms with different levels of autonomy. Digital maps representing road networks are being used as an *a priori* source of information to provide context and to anticipate oncoming situation. On top of it, world models are built for machine understanding. However, these can have local errors, affecting location based functions. A common one is due to the rapid deployment of roundabouts, as they are introduced widely, with navigation maps ignoring their presence. This work introduces a novel approach to make the vehicle able to detect the presence of a roundabout while it is driven. It is then possible to update the map by the vehicle itself. The approach is based on graphical pattern recognition methods using a Bayesian classifier. The approach has been demonstrated experimentally using data acquired in real-traffic conditions.

## I. INTRODUCTION

Continuous progress in information and communication technologies has meant that modern vehicles are being transformed into sensor-based platforms with often advanced wireless connectivity capabilities. Most vehicle manufacturers are today working on the deployment of Advanced Driving Assistance Systems (ADAS). The rationale is to facilitate the comfort of driving tasks, reduce the number of accidents as well as to improve traffic flow whilst including costs as a major constraint.

Statistics have shown that the introduction of ADAS functions improves safety. For example, the introduction of an “Automatic Intelligent Speed Assistant” could reduce up to 15% of road fatalities [1]. However, ADAS require more and more detailed information about the driving context the vehicle is going to be facing in the near future. This is provided in some part by sensors mounted on the vehicle like radar or camera especially for the close vehicle’s surroundings because they are necessarily limited by their fields of view and the occlusions due to obstacles in the scene (other vehicles for instance). The high cost of these sensors is also a drawback for their use in standard vehicles. Digital maps are gradually used to get a wider scale context information. For example, as they store road geometry information, they may be used to help to control the vehicle speed and bend the front lights when a curve is approaching [2]. Moreover, a large amount of road attributes are kept in the digital maps which makes possible to inform the driver of the current speed limit [3], to change the perception

field of view in accordance to the kind of environment the vehicle is evolving in (rural or urban) [4], or to turn off the cruise control system as soon as the vehicle is approaching a crossing or a toll gate. This is obviously in addition to the standard use of maps and global navigation satellite systems (GNSS) for guidance purposes.

However, digital maps include multiple errors originating in the collected data or in changes that have occurred since the compilation of the maps and their distribution to the vehicles. This has resulted in serious integrity issues that not only are a common client complain of users of navigation devices but also reduce the integrity of safety related applications. Within this context, it is very important to be able to determine when the maps are erroneous and, if possible, to correct them using data which is collected by the subject vehicle. Confidence in the detected errors and in the corrections to make should increase as the trajectory is repeated by the vehicle and shared with other users.

Map errors can be classified in 3 categories: geometrical, topological and attributes. Geometrical errors make that the map doesn’t match with reality. Topological errors refer to connection errors between nodes of the map. Incorrect attributes mean that the semantic information extracted from the map (like speed limit) doesn’t correspond to reality. We consider in the following only geometrical errors.

In practice, map engines are in charge of providing the pertinent map information to the ADAS functions through an Electronic Horizon [5] for instance. In order to monitor the map quality, a sub-system is introduced between the map engine and the ADAS (Fig. 1). It collects the GPS and proprioceptive information to make its own representation of the travelled trajectory which is then used to detect errors in the map. Once an error has been detected, the faulty link ID is then stored in a look-aside database (LADB). This “don’t use the map” information will be exploited by the ADAS next time the vehicle will travel in the same area.

In this paper, we focus on one particular kind of geometrical errors, the missing “roundabouts”. Roundabouts (i.e. traffic circles) are particularly interesting elements of the road networks that affect the content of a map. They are introduced extensively instead of crossroad intersections because they divide by four the number of potential collision points, they reduce the vehicle speed and improve traffic flow [6]. Further, they may not be represented in digital navigation maps. These also have significant consequences on ADAS. For example, a cruise control system must be turned off when the vehicle approaches a roundabout which is possible only if it has been registered in the map.

<sup>1</sup> Heudiasyc UMR CNRS 7253, Université de Technologie de Compiègne, France. philippe.bonnifait@hds.utc.fr, clement.zinoune@utc.fr

<sup>2</sup> Renault S.A.S, France. javier.ibanez-guzman@renault.com

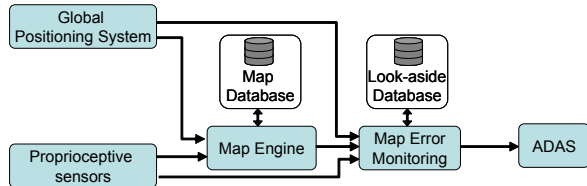


Fig. 1. Map monitoring architecture. The system informs the map-aided ADAS whether the map is wrong thanks to data previously collected and stored in a look-aside database.

On the one hand, road changes including new roundabouts may be detected by using image processing methods like active contour [7] or gradient [8] on aerial images. On the other hand, the vehicle itself can be used to create maps by converting the vehicle’s trajectory into road segments [9]. This generally assumes a very accurate (and so often expensive) localisation system like bi-frequency GPS combined with inertial measurements units or lidars.

Methods based on usual vehicles equipments can also be found in the literature. In [10], a camera mounted underneath the vehicle’s mirror is used to extract and classify road singularities using a neural network. [11] uses also a neural network architecture to classify roads travelled by several vehicles into classes like *road*, *local street*, *roundabout* or *traffic light stop* according to a set of parameters (speed, rate of acceleration, curvature or change in travel direction). [12] proposes algorithms to infer the presence of road or roundabouts based on offline and centralized GPS trace mining.

In this paper, a method for detecting the presence of roundabouts is studied, using a ego-localization systems which exploits the vehicle embedded sensors. The idea is to recognize roundabouts in the passed vehicle’s path. The immediate trace of the position estimates is first converted in an adequately sampled window. Then, an efficient descriptor is proposed and exploited in a Bayesian classifier that has been chosen to take profit of its low computational complexity and make embedded application possible. Once a roundabout has been detected, this information is compared to the digital map content. If the roundabout is not already recorded in the map, an error is detected and recorded in the LADB.

The remainder of the paper is organized as follows: Section II formulates the proposed framework including the localization algorithm, the preprocessing necessary to feature recognition, the descriptors, the defined classes as well as the estimation of the roundabout characteristics. Section III, includes the experimental content, it validates the proposed approach using data acquired from a test vehicle operating in real traffic conditions. It includes the experimental apparatus, and an analysis of the algorithm performances. Finally, Section IV concludes the paper.

## II. FRAMEWORK

The algorithm presented here uses as input information coming from standard sensors mounted in the vehicle and

a standard global positioning system (GPS). The odometer, the yaw rate, the speed and the speed difference between the rear wheels are used here. They have the advantage to be broadcast on one single bus in the vehicle (CAN-bus) so to be easily read and recorded. Proprioceptive sensors are very efficient to provide a precise estimate of the travelled path on a short horizon of time. GPS provides the global position of the vehicle which makes possible to georeference the roundabouts as soon as they are detected. Merging the two modalities allows localizing precisely and globally.

### A. Localization

The vehicle localisation task is done by a loosely coupled Extended Kalman Filter (EKF). It uses GPS fixes when available to correct the proprioceptive estimation [13]. The state representation is shown in (1).

$$\begin{cases} X_{k+1} = f(X_k, v_k) + \alpha_k \\ Y_k = H.X_k + \beta_k \end{cases} \quad (1)$$

$$\begin{cases} X_k = [x_k \ y_k \ \omega_k \ b_k \ \psi_k]^T \\ Y_k = [x_{gps \ k} \ y_{gps \ k} \ \omega_{g \ k} \ r_k]^T \end{cases} \quad (2)$$

The state, observation and input vectors are detailed in (2), where index  $k$  denotes the  $k^{th}$  time step. In the state vector  $X$ ,  $(x \ y)$  are the vehicle’s coordinates in a East-North-Up (ENU) reference frame.  $\psi$  and  $\omega$  stand for the vehicle’s heading (clockwise from the North direction) and yaw rate (rotation rate around the  $Up$  axis) respectively. Finally,  $b$  is the vehicle’s gyroscope bias as defined in (4). The observation vector  $Y$  contains the vehicle’s position according to the GPS observation  $(x_{gps} \ y_{gps})$ , the yaw rate measured by the vehicle’s gyroscope  $\omega_g$  and the speed difference between the rear vehicle’s wheels  $r$ . The measured input is the vehicle’s speed  $v$ .

The model and observation noises  $\alpha$  and  $\beta$  respectively are supposed to be white and zero mean.

The state transition function  $f$  depends on the input  $v$  and on the state itself. It is defined in (3) where  $T$  is the sampling period and  $l$  is the vehicle’s rear track. The observation model  $H$  used here is given in (6).

$$f(X_k, v_k) = \begin{bmatrix} x_k + T v_k \sin(\psi_k) \\ y_k + T v_k \cos(\psi_k) \\ \omega_k \\ b_k \\ \psi_k - T \omega_k \end{bmatrix} \quad (3)$$

$$\omega_g = \omega + b \quad (4)$$

$$r = \frac{2}{l} \omega \quad (5)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2/l & 0 & 0 \end{bmatrix} \quad (6)$$

For a moving vehicle, the observability of each state variable can be easily demonstrated.

The vehicle's state is computed in real time and its position over the last  $L$  meter is kept in a sliding window. This track is used in the following to ascertain the vehicle has crossed a roundabout.

### B. Pre-processing

The first step of feature recognition is pre-processing. The area of interest is isolated from the whole vehicle track using the concept of *sliding window*, e.g. only the last  $L$  meters of the vehicle's trajectory are used to search for features.

The vehicle's odometer is used to spot the time since which the vehicle covered  $L$  meters. The vehicle positions are stored in a buffer memory and dismissed as soon as it goes out of the sliding window. The track present in the buffer memory at any time is called *buffer track*.

Pre-processing consists in changing the reference frame of the *buffer track* to reduce the diversity of the possible roundabout shapes. The shapes and roundabout orientations are then made independent. For this purpose, we propose to use an egocentric approach. The *buffer track* is transformed into the current vehicle frame using the rotation matrix  $Z$  defined in (7). As  $\psi$  denotes the vehicle's heading with respect to the North (vertical axis), this rotation matrix is slightly different from the usual one.

$$Z_k = \begin{bmatrix} \sin(\psi_k) & -\cos(\psi_k) \\ \cos(\psi_k) & \sin(\psi_k) \end{bmatrix} \quad (7)$$

### C. Descriptor

In classification and recognition tasks, the choice of a descriptor is crucial. In this paper, we focus on pieces of vehicle's trajectory as graphical 2D objects. The descriptor should then illustrate these two dimensions independently.

Our choice of a descriptor was motivated by the similarities between our problem and the handwriting recognition domain [14], [15]. Let the descriptor be composed of two halves: left and down profiles. The left (resp. down) vector stores the distances between the left (resp. down) side of the interval and the first point met on this line (resp. column) as shown in Fig. 2. The profiles contain  $n$  equally spaced elements.

One issue when one deals with features such as roundabouts is the diversity of the elements called roundabouts. This comes, on the one hand from the number of exits of the roundabout and the one chosen by the driver, and on the other hand the roundabout radius. The first difficulty will be solved by defining a suitable number of classes, as developed later on. Secondly, to prevent scale effects, the descriptor is normalized using (8). The descriptor of one *buffer track* is therefore a vector  $D$  of  $2n$  elements.

$$D = \left[ \begin{array}{cccc|cccc} c_1 & c_2 & \cdots & c_n & l_1 & l_2 & \cdots & l_n \\ \hline & \max_{i:1 \rightarrow n} (c_i) & & & \max_{i:1 \rightarrow n} (l_i) & & & \end{array} \right] \quad (8)$$

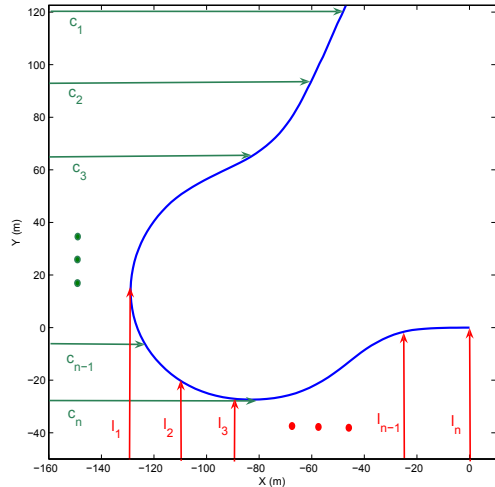


Fig. 2. An example of *buffer track* (in blue) with its associated descriptor. The first half is left profile (in green), the second one is down profile (in red).

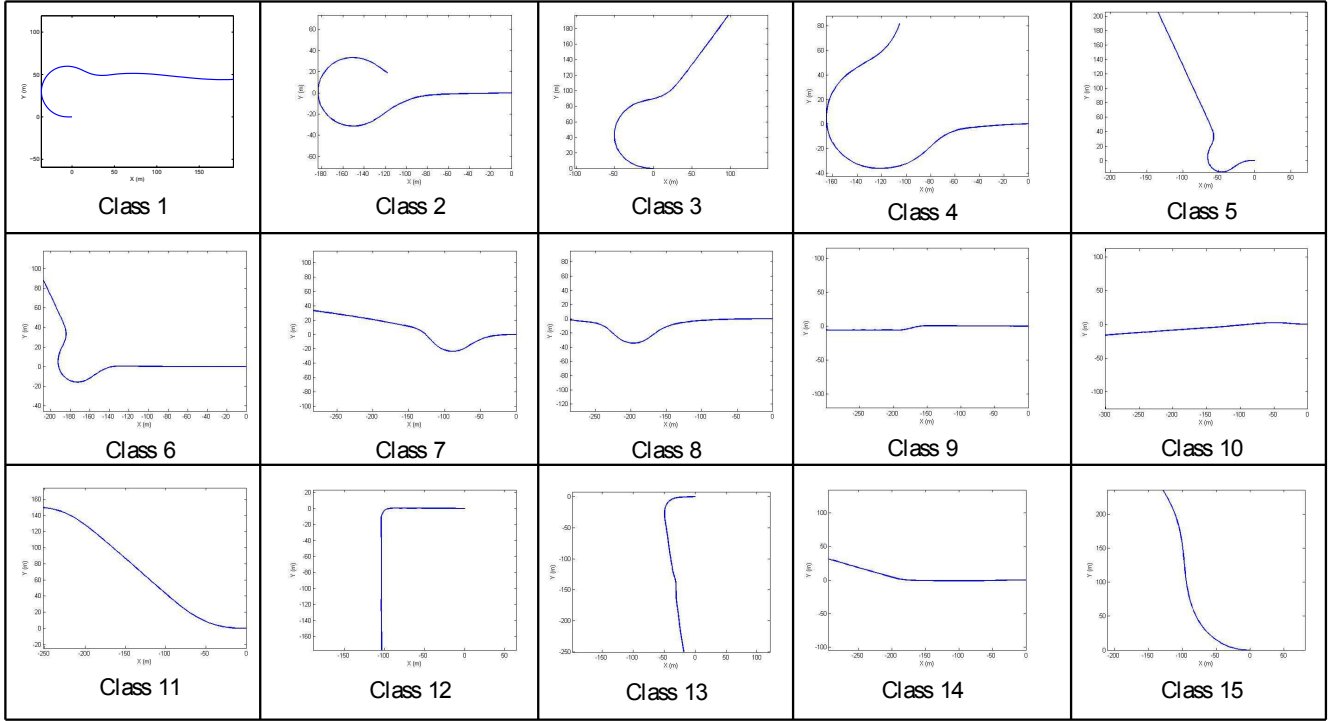
As soon as the vehicle goes straight, the projection of the *buffer track* along the  $Y$  axis becomes very small. The left profile sampling shown in Fig. 2 is made inconvenient and may introduce errors in the classifier. Moreover, in this situation, the vehicle cannot have crossed any roundabout, thus, *buffer tracks* that have a vertical range less than threshold  $\Delta Y_{min}$  are neglected.

### D. Classes

Consistent classes must be defined to make the recognition process efficient. It is not suitable to create only two main classes: one containing every *buffer track* that contains a roundabout and the other with anything else. Indeed, a too wide variety of shapes inside the same class (due to the large diversity of roundabout shapes as said previously) makes them inconsistent and the learning and recognition steps very tough. The solution found here has been to split the “roundabout” class into sub-classes. Each one being associated to one kind of *buffer track* containing a roundabout (one for the vehicle went straight, one for the vehicle took the third exit, ...). Likewise, several classes associated to non-roundabout *buffer tracks* have been designed.

Table I illustrates the typical *buffer tracks* of every class. These have been manually designed to represent the variety of the *buffer tracks* as well as possible. Thus, roundabout at which the vehicle went straight, turned left and made a U-turn are represented. Also, a class is set for each one of them when the roundabout is at the beginning and at the end of the *buffer track* to improve the robustness of the classifier. This makes the first eight classes. The same way to proceed is applied to non-roundabout classes: Nearly straight trajectories are in classes nine and ten, left turns are in classes eleven, fourteen and fifteen and right turns are in classes twelve and thirteen.

TABLE I  
CLASSES USED FOR ROUNDABOUT INFERENCE



Once these representative samples have been manually isolated, their corresponding descriptors are computed as developed in the previous paragraph.

### E. Recognition

Let us consider a closed frame of discernment  $\Omega$  and a set of classes  $\{w_i\}$  ( $i = 1 \rightarrow k$ ) being a partition of  $\Omega$ . According to total probability law and to the Bayes' theorem, the conditional probability of a sample  $D$  to belong to the  $i^{th}$  class is given by (9). In this equation, it is assumed that every class has the same prior probability.

$$P(w_i|D) = \frac{P(D|w_i)}{\sum_{j=1}^k P(D|w_j)} \quad (9)$$

We consider sample  $D$  as a vector containing  $2n$  random variables. Let us assume that, within a given class  $w_i$ , samples are normally distributed around the representative sample  $D_i$ . Thus, if  $D$  belongs to  $w_i$  then  $D \sim \mathcal{N}(D_i, P) \vee i \in \{1, \dots, k\}$  where  $P$  is the covariance matrix.

The posterior probability of  $w_i$  to be the correct class associated to the sample  $D$  is then given by (10).

$$P(w_i|D) = \frac{\exp\left(-\frac{(D-D_i)^T \cdot P^{-1} \cdot (D-D_i)}{2}\right)}{\sum_{j=1}^k \exp\left(-\frac{(D-D_j)^T \cdot P^{-1} \cdot (D-D_j)}{2}\right)} \quad (10)$$

The sample  $D$  is then associated to the maximum posterior probability class. This method minimises error probability.

### F. Roundabout placement

As soon as a roundabout has been detected in a *buffer track*, the final step is to find its centre and its radius. This allows to convert the discovery of a roundabout back into absolute coordinates and to identify it clearly into the LADB. This step is based on the clustering of the successive rotation centres. It is easy to prove that if three points are not aligned, a unique circle passing by them exists. The centre and radius of every points trio is computed along the whole *buffer track* as presented by blue crosses in Fig. 3. Considering the fact that the curvature radius remains almost constant while the vehicle is in the roundabout, the true centre belongs to the densest cluster. As shown by the red square and circle, the selected centre is the barycentre of this cluster. This method is a good compromise between simplicity and robustness.

The centre coordinates are converted back into absolute ENU using the reciprocals of (7) and then into global latitude/longitude coordinates.

Usually, the *roundabout* information is actually one of possible value of the attribute *road type* linked to a road segment (as *motorway*, *parking*, *tunnel*,...). If a roundabout has been detected for the first time by the vehicle and is missing in the digital map, its location is added to the digital map and a "don't use the digital map here" flag is raised for

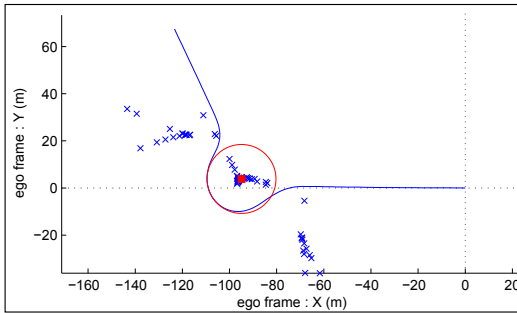


Fig. 3. Determination of the roundabout centre and radius in a *buffer track*. Blue line represents the *buffer track* itself and blue crosses are its successive rotation centres. Red square denotes the centre retained associated to its circle (red line).

ADAS. Then, next time the vehicle goes through this place, ADAS will be able to reject the map information.

### III. RESULTS

#### A. Experimental setup

The method has been tested in real conditions thanks to a *Renault Espace* passenger vehicle driven in various conditions, urban, suburban and rural environments. The localisation system on board was a standard single frequency Ublox 4T receiver. A Ixsea LandIns Inertial Navigation System (INS) tightly coupled to a Novatel GPS receiver has been used as position ground truth.

We also had access to more than 300 vehicle's internal states variables of the vehicle via the CAN-bus. More precisely, we have used the odometer, the speed, the rear wheel speed difference and the yaw rate.

The few parameters of the algorithm have been tuned as follows. First the length  $L$  of the *buffer track* must be larger than the perimeter of the biggest roundabout the vehicle may cross. We have set  $L = 250m$  which corresponds to a roundabout radius of 40 meters which is also the value chosen for the maximum detectable radius  $R_{Max}$  in the filter described before. Then, the minimum  $Y$  range defined in this filter is set so that  $\Delta Y_{min} = 10m$  which corresponds to minimum detectable radius of 5 meters for U-turns and 10 meters for straight crossings. Finally, as the covariance matrix  $P$  of the classifier is not sensitive, the identity matrix has been used.

#### B. Method performance evaluation

The most relevant way to assess the method is to use it on real data. A large amount of data representing more than 150 km have been recorded in various environments. Rural, suburban and urban areas have been covered totalling up 37 different roundabouts. Some of them have been crossed several times by different ways and under different traffic conditions.

Firstly, the performance of the localization sub-system has been assessed thanks to the ground truth provided by

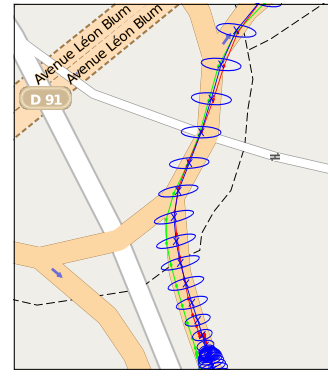


Fig. 4. Detail of vehicle's path. The position estimations are associated to their covariance ellipses (blue line, crosses and ellipses). GPS fixes used in the filter are denoted in red and ground truth is in green.

TABLE II  
METHOD PERFORMANCE

True positives: 100%	False negatives: 7.2%
False positives: 0%	True negatives: 92.8%

the INS. The maximum distance error is below 15 meters and equals 5 meters in average.

The consistency of the filter is highlighted in Fig. 4. It shows that the true position (in green) and the measured position (in red) are always inside the 3-sigma covariance ellipses (blue ellipses) which corresponds to a 99.7% probability. One can notice that when the vehicle is stopped (bottom side of the picture), these ellipses decrease due to a larger number of observations for the same position.

Secondly, the classifier performance has been deduced from more than 100 km of driving sequences. The classes assigned by the classifier have been manually compared to true classes. The results have been merged to provide the overall roundabout detection rates. These are summarized in Table II. One can notice that every *buffer track* classified as roundabout actually contains a roundabout, i.e. there are no false positives. Moreover, data used here contains ambiguous manoeuvres like U-turn and left turns with no roundabout. However some roundabouts are missed by the classifier. This is due, on the one hand, to the fact that the vehicle crossed them nearly straight which makes the roundabout almost invisible in the *buffer track*. On the other hand, the evaluation of a *buffer track* is independent from one time step to the other. A roundabout miss-detected when it was at the beginning of the track may be detected a few time steps later. This justifies the uses of two classes for one roundabout shape as denoted in II-D.

Fig. 5 represents a part of the outputs of the algorithm. One can notice that the roundabouts have been correctly detected and their centres located on the map. A direct application of the algorithm is shown in Fig. 6. The vehicle has been driven on a recently modified road. On road D926, a new roundabout has been built and another one is still under construction. We can see the roundabout centres (red stars) detected by the algorithm while the vehicle was driven



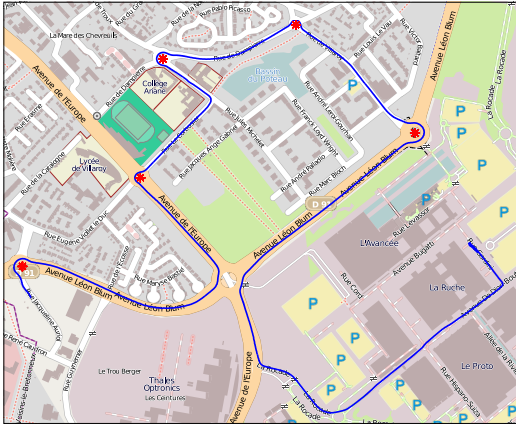


Fig. 5. The position estimates of the vehicle in urban area is represented by a blue line. The vehicle is travelling anticlockwise. The roundabout centres found by the program are shown by red stars.

along the blue path. It can also be noticed on this picture that the U-turn is successfully classified as a non-roundabout. The algorithm makes then possible to detect that the map is obsolete and to store this information for next passages in the LADB. We see in this situation that a cruise control system must be henceforth turned off on road D924.

The background map used in Fig. 5 and Fig. 6 to illustrate the results is taken from OpenStreetMap project [16], [17].

The method has been coded and tested in Matlab and runs more than 10 times faster than the real data. Then, even if it has not been implemented yet, it makes conceivable an embedded application.

#### IV. CONCLUSIONS AND FUTURE WORK

The presented approach is based on the notion of integrity monitoring of digital maps used in ADAS functions for passenger vehicles. To prevent malfunctions, the detection of one typical road geometrical error has been addressed, namely, missing representation of roundabouts in digital maps. The shape of a path followed by a vehicle when traversing a roundabout has a typical signature, thus it has been possible to apply a Bayesian classifier on data representing the traversed path. For this purpose, only data from existing vehicle sensors was used (i.e. odometer, gyroscope, speedometers and GPS). Further, the resulting algorithm has low computational complexity, so it can be easily incorporated into embedded applications. The proposed solution has been extensively tested on data acquired in real-traffic conditions showing promising results by detecting the roundabouts that the vehicle traversed whatever their diameter, shape or the driver trajectory within the roundabout. It was possible to detect recently built roundabouts and to add them onto a look-aside database as these were not yet present in commercially available digital maps. The on-line correction of digital road maps is applicable in particular to commuting trajectories, even after the first run. Thus the performance of map-based ADAS functions is made more reliable. Further to this work, other aspects of map integrity will be assessed drawing also advantage from perception system based on camera which appears widely in passenger vehicles.



Fig. 6. Example of application of the algorithm in rural environment. Here the road network has changed recently, one roundabout has been created and another one is being built. These are not registered in the digital map (red lines) yet. The vehicle track is in blue and the centres of the roundabouts found by our algorithm are in red stars.

#### REFERENCES

- [1] "Traffic Accident Causation in Europe Project (TRACE)," <http://www.trace-project.org>, 2008, accessed 30 January 2012.
- [2] M. Chowdhary, "Driver assistance applications based on automotive navigation system infrastructure," in *Int. Conf. on Consumer Electronics, Digest of Technical Papers*, 2002, pp. 38 – 39.
- [3] H. Janssen and W. Niehsen, "Vehicle surround sensing based on information fusion of monocular video and digital map," in *Intelligent Vehicles Symposium, IEEE*, June 2004, pp. 244 – 249.
- [4] A. Unterholzner and H.-J. Wunsche, "Adaptive state space control of a camera platform for an autonomous ground vehicle," in *Intelligent Vehicles Symposium, IEEE*, June 2009, pp. 591 – 596.
- [5] S. Durekovic and N. Smith, "Architectures of map-supported ADAS," in *IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 207 – 211.
- [6] R. Ewing and E. Dumbaugh, "The built environment and traffic safety: A review of empirical evidence," *Journal of Planning Literature*, vol. 23, no. 4, pp. 347–367, May 2009.
- [7] M. Ravanbakhsh and C. S. Fraser, "Road roundabout extraction from very high resolution aerial imagery," *International Archives of Photogrammetry and Remote Sensing*, 2009.
- [8] M. de Gunst and J. den Hartog, "Knowledge-based updating of maps by interpretation of aerial images," in *Pattern Recognition, Computer Vision Image Processing., Proceedings of the 12th IAPR*, 1994.
- [9] A. Chen, A. Ramanandan, and J. A. Farrell, "High-precision lane-level road map building for vehicle navigation," in *Position Location and Navigation Symposium (PLANS), IEEE/ION*, 2010.
- [10] A. P. Leitão, S. Tilie, M. Mangeas, J. p. Tarel, V. Vigneron, and S. Lel, "Road singularities detection and classification," in *European Symposium on Artificial Neural Networks*, 2003.
- [11] F. Ekpenyong, D. Palmer-brown, and A. Brimicombe, "Updating of road network databases: Spatio-temporal trajectory grouping using snap-drift neural network," in *Int. Conf. on Engineering Applications of Neural Networks (EANN07)*, vol. 284, 2007, pp. 237–246.
- [12] G. Agamennoni, J. Nieto, and E. Nebot, "Robust and accurate road map inference," in *Robotics and Automation (ICRA)*, 2010, pp. 3946 – 3953.
- [13] P. Bonnfait, P. Bouron, P. Crubille, and D. Meizel, "Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles," in *IEEE International Conference on Robotics and Automation, ICRA*, vol. 2, 2001, pp. 1597–1602 vol.2.
- [14] K. bin Abd and S. Hashim, "Handwriting identification: a direction review," in *Signal and Image Processing Applications (ICSIPA)*, Nov. 2009, pp. 459 – 463.
- [15] R. Plamondon and S. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE*, vol. 22, no. 1, pp. 63 – 84, Jan 2000.
- [16] M. Hentschel and B. Wagner, "Autonomous robot navigation based on openstreetmap geodata," in *Intelligent Transportation Systems (ITSC)*, Sept. 2010, pp. 1645 – 1650.
- [17] (2012, January) Openstreetmap project homepage. [Online]. Available: <http://www.openstreetmap.org>