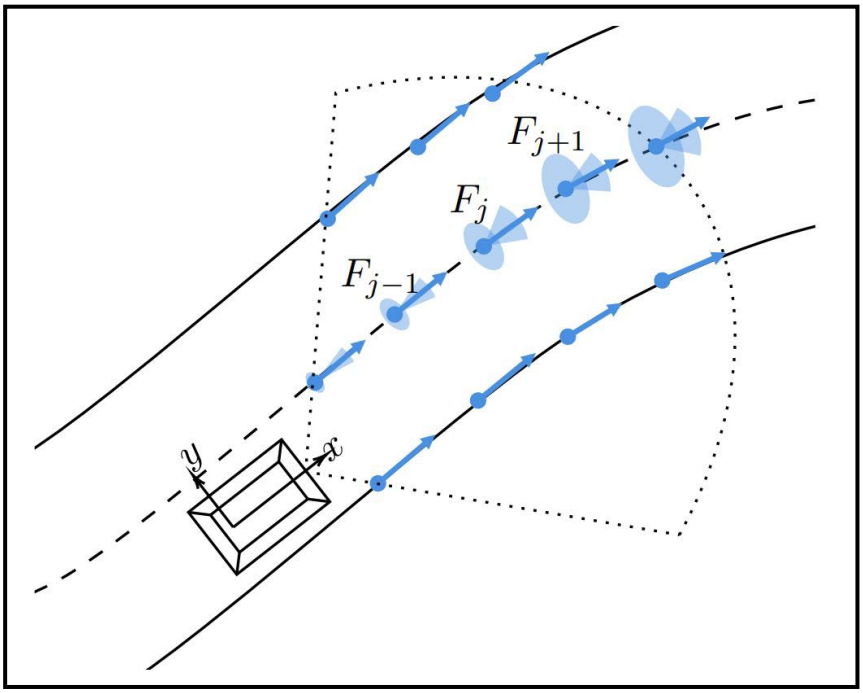
Multi-sensor data fusion for lane boundaries detection applied to autonomous vehicle

Federico Camarda

Par **Federico CAMARDA**

Fusion de données multi-capteurs pour la détection des bords de voie appliquée au véhicule autonome

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 14 janvier 2022
Spécialité : Automatique et Robotique : Unité de recherche Heudyasic (UMR-7253)
D2673

UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

THÈSE

Présentée pour l'obtention du grade de
Docteur de l'Université de Technologie de Compiègne

Spécialité : Automatique et Robotique

Unité de Recherche : Heudiasyc – UMR CNRS 7253

Par

Monsieur Federico CAMARDA

Fusion de données multi-capteurs pour la détection des bords de voie appliquée au véhicule autonome

Soutenue le 14 janvier 2022

Devant le jury composé de :

M. Philippe BONNIFAIT, professeur des universités, membre examinateur
Université de technologie de Compiègne, laboratoire Heudiasyc, Compiègne
M. Sergio RODRIGUEZ, maître de conférence, membre examinateur
Université Paris-Saclay, Cachan
M. Jean-Charles NOYER, professeur des universités, membre rapporteur
Université du Littoral Côte d'Opale, Calais
M. Roberto SACILE, professeur, membre rapporteur
Université de Gênes, Gênes, Italie
M^{me} Véronique CHERFAOUI, professeur des universités, directrice de thèse
Université de technologie de Compiègne, laboratoire Heudiasyc, Compiègne
M. Franck DAVOINE, chargé de recherches CNRS, directeur de thèse
Université de technologie de Compiègne, laboratoire Heudiasyc, Compiègne

Invité
M. Bruno DURANT, référent AD/ADAS,
Renault S.A.S., Guyancourt

Multi-sensor data fusion for lane boundaries detection applied to autonomous vehicle

Federico Camarda

Defended on January 14, 2022

Acknowledgments

It's been three tough years. Well, it would be such an understatement to say that these three specific years have just been *"tough"*. They have been discouraging, and harsh, and painful, and isolating, and heartbreaking. But it's over now, it's all over now. We did it. *"Haut les cœurs. On peut encore se parler, se toucher, se voir"*. Also, I did it. Mamma, ce l'ho fatta.

First, I'd like to thank my PhD advisors. Véronique Cherfaoui, Franck Davoine and Bruno Durand guided my choices throughout the entire journey with their wisdom, scientific knowledge and industrial experience. Thanks to all the jury members for their interest and participation in the evaluation process of my PhD work. I want to further express my gratitude to Philippe Bonnifait and Roberto Sacile, for establishing that UNIGE-UTC relationship that ultimately brought me to the achievement of this academic milestone.

Thanks to the colleagues and coworkers from both the Heudiasyc laboratory and the Renault Group offices. In-between these two workplaces I had the most interesting conversations, pauses café and motivational moments. I thank my fellow doctorants du labo Stefano, Anthony et Gabriel along with the outstanding thésards Renault Jean, Edouard et Louis. We went through this together and you have all been an important source of inspiration for me.

Thanks to all of my friends for supporting, with patience, often without clearly understanding what I was going through. But, still, supporting. Whether we met in Compiègne, in Paris, a Genova, during the thesis or before, I am grateful for having you around - it would have not been the same without you, not at all.

A few quick words thanking the memes, the songs, and the Twitch live streams that have been the soundtrack of this locked-down PhD thesis. *!watchtime : dottorzini ha passato 1 month 2 days 6 hours a guardare homyatol*. Looks like I've been spending enough time on that one already.

I conclude expressing the highest gratitude to my family, quoting myself from some previous thanking-list and witnessing that the following acknowledgement remains gloriously relevant over the years : *Grazie alla mia famiglia che da sempre crede in me, mi sostiene e rimane unita. Grazie Marti, Mamma e Papà.*

Abstract (en)

Perception and correct understanding of the road scene is crucial for any application of assisted and automated driving. In order to guarantee safety of the passenger and other road users, planning and navigation must be made on the basis of a reliable environment representation. Multi-sensor data and prior information is used to build this representation which incorporates identification of road users and road structure. For the latter, the focus is put on the drivable space and lane repartition. On highways, urban streets and generally all over the road network, the drivable space is organized in oriented corridors which enable safer and predictable navigation for everyone. In the development of intelligent vehicles, identifying the lane repartition and building an accurate road representation outlines the lane boundaries detection task. Depending on the specifics of the target automated system, car manufacturers integrate in currently commercialized vehicles ready-to-use lane detection solutions from Tier-1 suppliers generally featuring single and vision-based smart sensors. Such solutions may not be adequate in highly automated systems where the driver is allowed to divert their attention from the driving task and become passenger.

This thesis addresses the problem of lane boundaries identification relying on multi-sensor fusion of smart camera data (specifically, frontal and AVM cameras) and HD-maps. In the first part, an appropriate modeling for smart sensor measurements which is independent from the sensor nature is proposed. Uncertain detections of markings, barriers and other road elements contribute to the tracking of the surrounding lane boundaries using a novel clothoid-spline model. The second part focuses on the integration of prior information coming from digital maps. Similarly to the modeling of smart sensors, the involved uncertainties in the usage of map-providers have been taken into account to support the lane boundaries estimation. For the testing of the proposed approaches, a custom dataset of road data has been recorded using both off-the-shelf smart sensors and live streamed HD-maps. Validated and tuned tracking solutions are then integrated in close-loop experimentations on Renault prototype vehicle of SAE Level 3 of automation.

Resumé (fr)

La perception et la compréhension correcte de la scène routière sont cruciales pour toute application de conduite assistée et automatisée. Afin de garantir la sécurité du passager et des autres usagers de la route, la planification et la navigation doivent être effectuées sur la base d'une représentation fiable de l' environnement. Les données multi-capteurs et les informations préalables sont utilisées pour construire cette représentation qui intègre l'identification des usagers de la route et de la structure de la route. Pour cette dernière, l'accent est mis sur l'espace de conduite et la répartition en voies. Sur les autoroutes, les rues urbaines et plus généralement sur l'ensemble du réseau routier, l'espace carrossable est organisé en couloirs orientés qui permettent une navigation plus sûre et prévisible pour tous. Dans le cadre du développement des véhicules intelligents, l'identification de la répartition des voies et la construction d'une représentation précise de la route constituent la tâche de détection des limites des voies. En fonction des spécificités du système automatisé cible, les constructeurs automobiles intègrent dans les véhicules actuellement commercialisés des solutions de détection des voies prêtes à l'emploi provenant de fournisseurs Tier-1 et comprenant généralement des capteurs intelligents simples et basés sur la vision. Ces solutions peuvent ne pas être adéquates dans les systèmes hautement automatisés où le conducteur est autorisé à détourner son attention de la tâche de conduite pour devenir un passager.

Cette thèse aborde le problème de l'identification des limites de voies en s'appuyant sur la fusion multi-capteurs des données des caméras intelligentes (en particulier, les caméras frontales et AVM) et des cartes HD. Dans la première partie, une modélisation appropriée pour les mesures des capteurs intelligents, indépendante de la nature du capteur, est proposée. Les détections incertaines des marquages, des barrières et d'autres éléments de la route contribuent au suivi des limites des voies environnantes à l'aide d'un nouveau modèle de splines de clothoïdes. La deuxième partie se concentre sur l'intégration d'informations préalables provenant de cartes numériques. Comme pour la modélisation des capteurs intelligents, les incertitudes liées à l'utilisation des fournisseurs de cartes ont été prises en compte pour aider à l'estimation des limites de la voie. Pour tester les approches proposées, un ensemble sur mesure de données routières a été enregistré en utilisant à la fois des capteurs intelligents disponibles dans le commerce et des cartes HD live stream. Les solutions de suivi validées et tunées sont ensuite intégrées dans des expérimentations en boucle fermée sur un véhicule prototype Renault de niveau 3 d'automatisation SAE.

Abstract (it)

La percezione e la corretta comprensione della scena stradale è fondamentale per qualsiasi applicazione di guida assistita e automatizzata. Per garantire la sicurezza del passeggero e degli altri utenti della strada, la pianificazione e la navigazione devono essere effettuate sulla base di una rappresentazione affidabile dell'ambiente. Dati di origine multi-sensore e informazioni disponibili a priori sono utilizzati per costruire questa rappresentazione che incorpora l' identificazione degli utenti della strada e la struttura della strada stessa. Per quest'ultima, l' attenzione è posta sullo spazio percorribile e sulla ripartizione in corsie. Sulle autostrade, le strade urbane e in generale su tutta la rete stradale, lo spazio percorribile è organizzato in corridoi orientati che permettono una navigazione più sicura e prevedibile per tutti. Nello sviluppo di veicoli intelligenti, l' identificazione della ripartizione in corsie e la costruzione di una rappresentazione accurata della strada delinea il compito di rilevamento dei confini delle corsie o lane boundaries detection. A seconda delle specifiche del sistema automatizzato di destinazione, le case automobilistiche integrano nei veicoli attualmente commercializzati soluzioni di rilevamento di corsia pronte all'uso da fornitori Tier-1, generalmente composte di singoli sensori intelligenti e basate sulla visione computerizzata. Tali soluzioni potrebbero non essere adeguate in sistemi altamente automatizzati dove al guidatore è permesso di distogliere l'attenzione dal compito di guida e di diventare passeggero.

Questa tesi di dottorato affronta il problema dell'identificazione dei limiti di corsia basandosi sulla fusione multi-sensore di dati provenienti da telecamere intelligenti (in particolare, telecamere frontali e AVM) e mappe HD. Nella prima parte, viene proposta una modellazione appropriata per le misure dei sensori intelligenti che è indipendente dalla natura del sensore. I rilevamenti incerti di marcature, barriere e altri elementi stradali contribuiscono alla stima dei limiti delle corsie circostanti utilizzando un nuovo modello di spline di clotoidi. La seconda parte si concentra sull'integrazione di informazioni provenienti da mappe digitali. Analogamente alla modellazione dei sensori intelligenti, le incertezze coinvolte nell'uso di map-providers sono state prese in considerazione per supportare l'identificazione dei limiti di corsia. Per testare gli approcci proposti, è stato registrato un dataset personalizzato di dati stradali utilizzando sia sensori intelligenti off-the-shelf che mappe HD in live streaming. Le soluzioni di tracking convalidate e correttamente regolate sono poi integrate in sperimentazioni a circuito chiuso su un veicolo prototipo Renault di livello 3 di automazione SAE.

Contents

List of Figures

Chapter 1

General introduction

1.1 ADAS and AD

The automotive industry is experiencing a full-scale technological revolution. Advanced driving assistant systems (ADAS) and development of autonomous driving (AD) are currently part of the vision of most car manufacturer in the market. In fact, convenient assistance features for easier highway driving or essential safety capabilities are expected to be available on newly manufactured vehicles.

As customers get used to vehicles with on-board technology, the task of driving has become safer other than easier. Since its peak in 1972 [ONISR, 2020], road mortality has been continuously decreasing. Despite the ever-increasing spread of motorized vehicles, this trend has been supported by the evolution of the road network, the introduction of more adequate traffic laws and the refinement of driving licences. While the driving context keeps developing towards safety, recent studies point out that the main causes of road accidents can be directly attributed to driver. Even though many accidents are prevented every day by enhancing the driver awareness with the aid of informative ADAS, for a large portion of cases this approach would have not been effective. For example, in the case of tired driver or driver under influences, any amount of additional information about the surroundings may not be appropriately taken advantage of. This cases would be better handled by proactive ADAS, which act and control the vehicle on behalf of the driver. This approach taken to the extreme concertizes itself in the self-driving or autonomous vehicle (AV): the role of the driver is completely replaced and deployed by an automated system which is in charge of safely transporting the user to the destination.

Within this vision of automated transportation system, other forward-looking consequences or possible implementations become possible. Of course, safe and easily accessible transport for a larger public would be an immediate reality. New business models would be possible where fleets of vehicles are deployed for automatically addressing the need of transporting goods and passengers inbetween locations. This might even imply disruptive demographic changes where the technology proves to be scalable, cost-efficient, compatible with the public acceptance and evolution of the road regulations. Also the carbon footprint, which

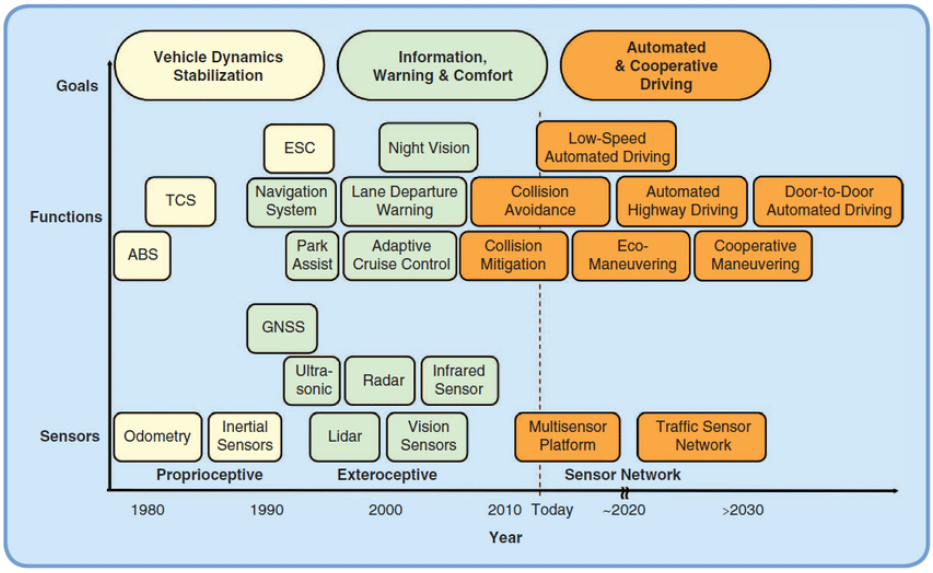


Figure 1.1: Past and potential future evolution toward Automated Driving [Bengler et al., 2014]. Early driving assistance systems relied on monitoring the ego-vehicle while more sophisticated solutions now observe the other road users and communicate with them.

challenge is currently of great interest and related to the deployment of renewable energies and electric vehicle, would be a key factor in the possible spread of AVs.

These considerations remain pure speculation on how would it be living in a world with AVs, however this technology is not available yet and appears long in coming. Fig.1.1 shows past and potential future evolution from assisted to automated driving.

1.1.1 Informative ADAS

Hereinafter, we present some example of informative ADAS. This kind of equipment is designed to assist the driver with goal of enhancing its awareness with some additional information that may be useful to the driving task. Depending on the seriousness of the information to be reported, different approaches of reporting it may be considered. For example, in case of immediate danger, strong alert sound may be preferred to simple notifications on the vehicle user interface (HMI). Most of commercialized vehicle nowadays may integrate:

- Lane Departure Warning (LDW): an alarm informs the driver when the vehicle dangerously approaches or crosses one of the lane boundaries of its lane
- Blind Spot Warning (BSW): indicator lights in the lateral rear-view mirrors inform the driver about the presence of others in the blind spots around the vehicle

- Parking Sensor: in the parking phase, the driver has better knowledge of close obstacles through changing frequency sound alert which helps avoiding damage to the vehicles itself or other's
- Driver Monitoring System (DMS): this time is the driver themself to be monitored. Warnings or encouragement to take a break are issued when tiredness or inattentive driving behaviour are detected

This approaches already contributed to the decrease of road mortality. However, they might be ineffective in many cases such as when the driver is unable to react to those warnings or when it is under the effect of substances. In fact, these cases still reflect a vast majority of the reported accidents [ONISR, 2020]. Informative ADAS can not prevent improper use of the vehicle, no matter how aggressive the alarm might be. It is not rare to find experienced drivers who prefer to deliberately ignore parking sensors alerts and, eventually, end up bumping into obstacles. Scratching car bumpers might not be an issue, yet other misuses can have safety consequences such as applying workarounds in order to disable the seat belt chime. It is quite clear that human decisions have a major impact on road causalities and that more active approaches of driving assistance are necessary to mitigate them.

1.1.2 Actuating ADAS

Driving assistance systems are also developed to be capable of actively take partial control of vehicle where certain conditions are met. Relevant examples are presented in the following. Some of them have been designed to be active extensions of the previously introduced informative ADAS. In fact, in many driving situations, after acknowledging relevant events around the vehicle, consecutive and automated actions can be operated in order to react and generally bring better safety. Most of commercialized vehicle nowadays may integrate:

- Adaptive Cruise Control (ACC): regular cruise control handles the throttling control by keeping the speed of the car as close as possible to a given value stated by the driver. Its "adaptive" extension adjust this value to the behaviour of the preceding vehicle while respecting the recommended inter-distance
- Lane Keeping Assistance (LKA): example of actuating system which extends the informative LDW. Controlled action of the steering wheel is enforced to maintain the car within the lane boundaries
- Lane Centering Assistance (LCA): further refinement of LKA, the vehicle is kept at the center of the lane. This systems aims at maintaining this trail the whole time after it is enabled by the driver.
- Automatic Emergency Braking (AEB): in case of sudden danger, the braking system is immediately activated. This is particularly precious in cases of unexpected pedestrians or animals on the roadway

- Traffic Jam Pilot (TJP): acceleration and braking are automatically handled in heavy traffic conditions
- Automatic Parking: completely automated routine for performing angle, perpendicular or parallel parking

All these features are generally implemented by acting separately on distinguished control dimensions of the vehicle. While steering wheel determines the car lateral position, action on throttle and breaking operates on the longitudinal position. Any measure of automated control of the vehicle must be acted in confirmed safety conditions. Besides, here we find first examples of actively actuating on the control of the vehicle as a way of easing the driving task, not only with the aim of reacting to dangerous situations.

1.1.3 Automated Driving

The development of more and more advanced ADAS suggests the real possibility of reducing to the minimum the required input from a human driver within the driving task. In the ideal case of automated driving (AD), as presented and studied by [Anderson et al., 2014], the driver would become a passenger and would simply indicate its preferred destination. The specific sequence of actuating control in order to safely transport the vehicle containing the passenger up to its destination would be completely handled by the driving system. The design of such system is expected to be incremental with respect to the currently available driving assistance systems and different degrees of automation have been defined. Firstly published in 2014, updated report [SAE International, 2021] from the Society of Automotive Engineers (SAE) declines six levels of automated driving as presented in Fig.1.2. As outlined in the table columns colouring, from level 0 to level 2 the human driver involvement is still required. From level 3 to level 5, the driving task is more and totally delegated to the system. Abbreviated notation is often adopted in the automotive industry associating labels from L0 (level 0) to L5 (level 5) to specific models of vehicles. Indeed, the prototype vehicle used in this research and presented in the following chapter is used at Renault for level 3 experiments, hence its naming *L3 prototype*. It is however important to note that it is not a specific system that identifies its level of automation, it is rather its utilization.

Let us illustrate few examples of common driving routines observing Fig. 1.3. From top to bottom, six examples of driving routines are given specifying what is the SAE AD Level that characterize each segment of the path from departure to arrival. The first example is trivial, there is no use of any driving assistance system activated by the human driver. In the third example, the vehicle is equipped with Automatic Parking and (possibly) no other ADAS. In this commute, the driver is completely involved in the driving task (level 0) expect for the very last maneuver (level 2). Differently, the vehicle deployed in the fourth example disposes instead of Automatic Parking and TJP: in the case of heavy traffic, the driver is allowed to delegate this sub-part of the trip to its vehicle. As this concerns more than a simple maneuver, phases of level 3/4 are identified in its

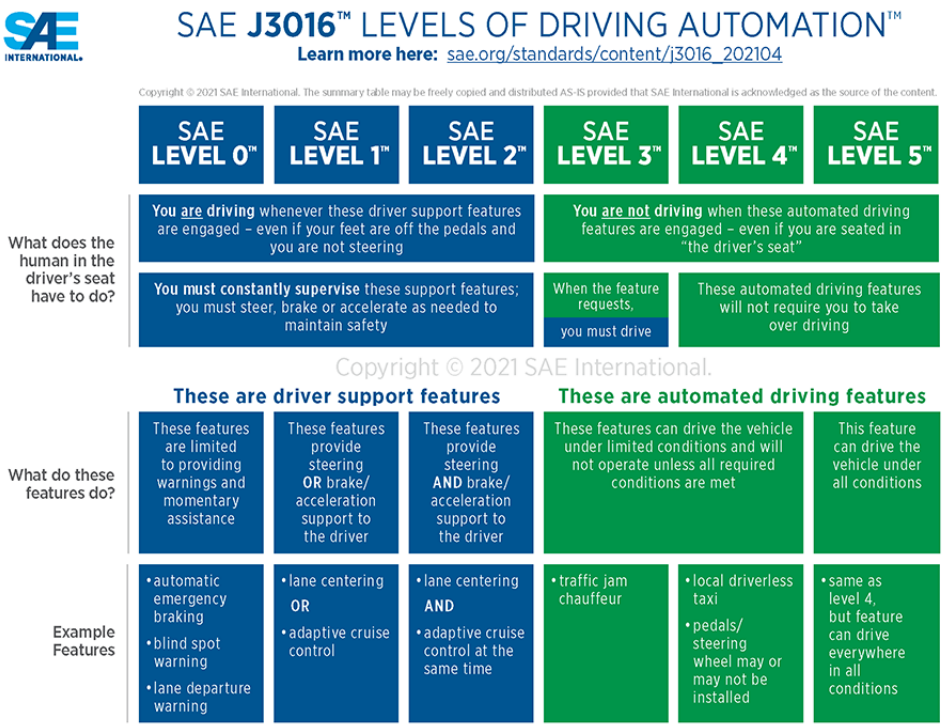


Figure 1.2: Levels of driving automation [SAE International, 2021]. From level 0 to level 2 the human driver involvement is still required. From level 3 to level 5, the driving task is more and totally delegated to the system

commute, depending on the specifics of its TJP. All the examples in Fig.1.3 drawn from [SAE International, 2021] entail the definition of an Operational Design Domain (ODD). The ODD, according to SAE, is the operating conditions under which a given driving automation system is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics.

1.2 Intelligent vehicles technology

We have presented so far the taxonomy of useful features that would make a vehicle "intelligent" and able to perform maneuvers of high level of autonomy. In the following, we are introducing how this automation is realized and what is the enabling technology deployed. At any degree of automation complexity, the system would have to be able to perceive an event, to elaborate the appropriate response and to put into practice its response. Consequently, onboard technology has to enable the vehicle to:

- Sensing - as much as possible about the driving scene and its users (ego-

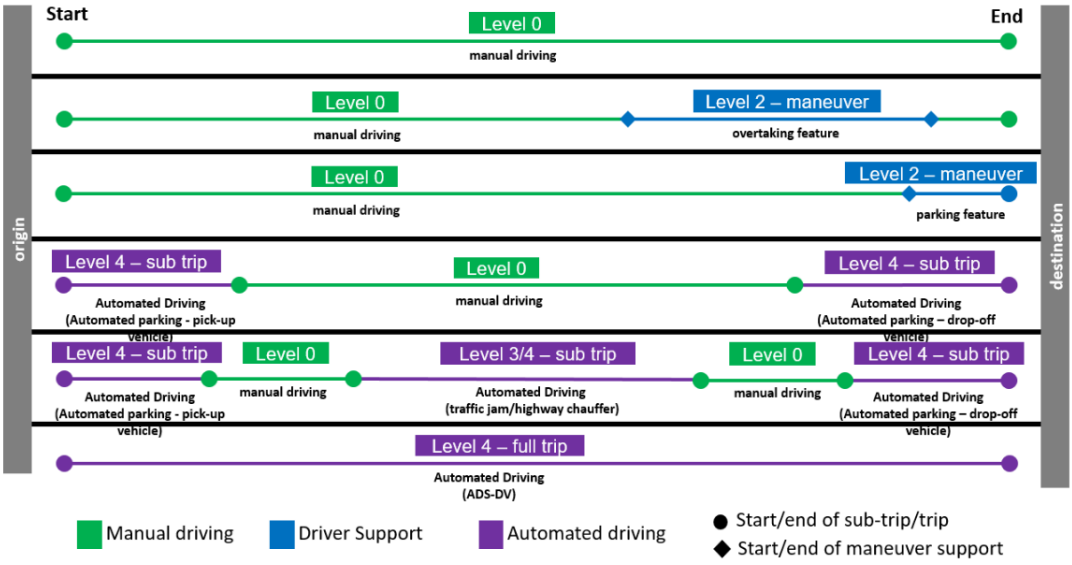


Figure 1.3: Six examples of driving routines [SAE International, 2021] are given specifying what is the SAE AD Level that characterize each segment of the path from departure to arrival.

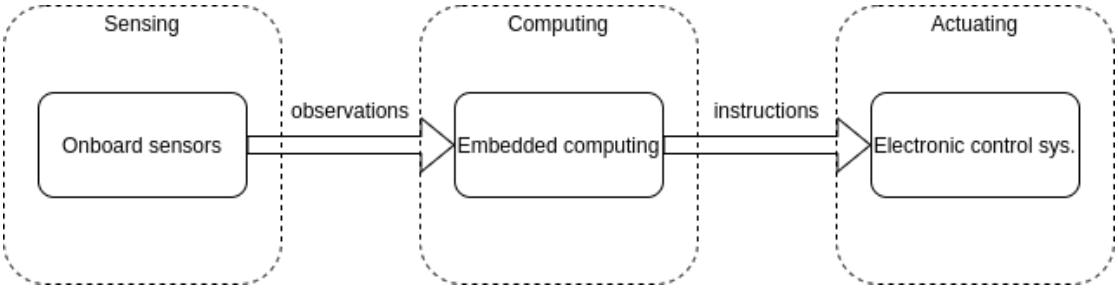


Figure 1.4: A schematic of the AD/ADAS workflow in its three main phases: sensing, computing and actuating.

vehicle included)

- Computing - the appropriate reaction to environmental events
- Actuating - and finally affecting the position of the ego-vehicle

These three main phases of the workflow in automated driving systems are illustrated in Fig.1.4 and enabled by specific technological

1.2.1 Onboard sensors

Information about the surrounding environment are given to the vehicle computing unit via streams of sensor data. Depending on the specific sensor, the device is installed on the vehicle body generally "facing" the area or elements that it aims at observing. For example, a tachometer would be observing the engine shaft in the car measuring its rotation speed while a dash cam would be

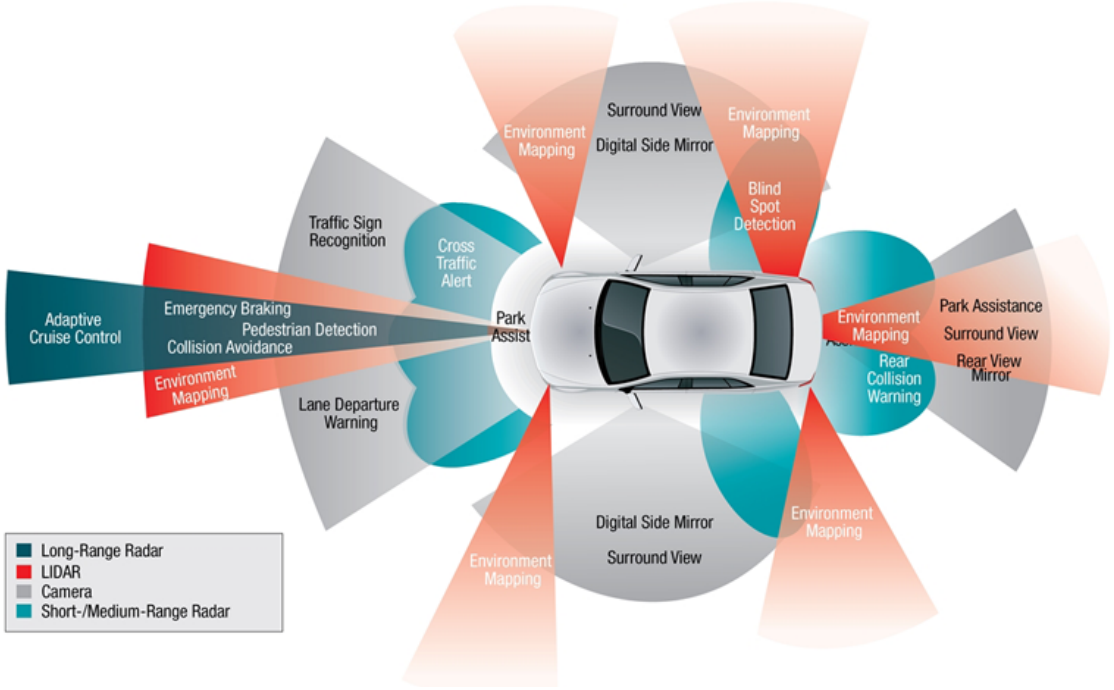


Figure 1.5: Exteroceptive sensors [Lengyel and Szalay, 2019] installed on a vehicle depting FoVs and ADAS features.

oriented and installed in order to have an optimal view of the driving scene. Fig. 1.5 shows an example of vehicle equipped of several exteroceptive sensors and presented in a bird-eye view schema. Each device identifies a surface around the vehicle representing its Field of View (FoV). In the illustration, a different colour is adopted for distinct technologies and a (possible) driving assistance feature is indicated for that sensor.

A Camera

Cameras are among the most popular perception sensors used in the industry. Installed nowadays on millions of vehicles, they are cheap and easy to be integrated in the car architecture. Images are 2D projections of portions (which depends on the field of view) of the surrounding 3D scene. At data-level, they are represented as matrices where each cell is pixel of a specific colour. Pixels may contain different information other than colour of the area they represent, such as heat-map for from thermal cameras. Usage of multiple cameras allows up to 360°field of view (as in Around Monitor View - AVM) or depth measuring (stereo vision camera). Several years of developments in computer vision and use of Artificial Intelligence (AI) make these sensors extremely powerful in shape recognition and object detection. Nevertheless, lighting condition have major impact on the operability of this technology, unlike others.

B Lidar

Lidar (acronym for Light Detection and Ranging) sensors enable time-of-flight distance measurements without physical contact using laser emitter and receiver. Mechanical lidars use rotating mirrors in order to produce multi-layer point clouds (set of range detections acquired in a unique scan) using a single emitter. Emerging solid-state lidars do not contain rotational elements, which make them preferable for easier maintenance and integration. On the one hand, this technology is robust to both dark and glaring lightning conditions. On the other hand, it suffers heavy weather, especially dense fog [Carballo et al., 2020], while each measurement return of intensity depends on the encountered material and colour.

C Radar

Most of currently commercialied ACC use radar technology in their implementation. Radio waves are exploited to measure the time of flight of the radar pulse, hence calculating distances from obstacles. Frequency modulation and use of Doppler measurements enhance these application enabling measurements of target velocity. This technology do not suffer from poor lighting condition and may allow for "see-through" obstacle detections in some cases. It is generally less accurate than others in distance measurements and may suffer ghost detections.

D Ultrasounds

Enabling for close obstacles detection, it finds application mostly in Parking Assist and Automatic Parking. Time of flight for ultrasonic waves is measured as in the case of radar technology. They are generally installed at multiple occurrences in front and back bumpers in order to monitor their immediate surroundings in any weather or lighting conditions.

E Proprioceptive sensors

This class includes all those sensors designed to provide information about the vehicle itself. Low cost and generally available on most vehicles (for implementation of Anti-lock braking system (ABS) and Electronic Stability Program (ESP)), they are used to monitor velocity, acceleration, wheel rotation speed, and yaw rate. Odometry information is essential for estimating ego-vehicle trajectory and computing relative velocities of other road users. Where greater accuracy is required (in research application, mapping or higher levels of automated driving) the use of Inertial Measurement Units (IMU) is preferred, which might entail increases in equipment costs.

F Sensors integration

Car manufacturers do not generally dispose the required expertise for sensor design and production. First of all, this know-how was not essential in car produc-

tion up a few years go. Also, similar to the manufacturing of other components such as headlights and windshield wipers, specialized companies have emerged which address the needs of more than a single car manufacturer. That is the case of tier-1 suppliers: serving has middle-men between car OEMs (original equipment manufacturer) and hardware suppliers, they provide adapted solutions for automotive use cases. In Fig.1.6, the supply chain for relevant industrial cases is shown. OEMs generally interface with tier-1 suppliers negotiating product specifics and unit pricing. This would impact the production vehicle pricing and the remaining charge of integration work to be completed from the car manufacturer. In fact, answering the "make-or-buy" dilemma is not trivial. For the implementation of a radar-based AEB, for example, OEM may simply buy and integrate a ready-made solution which would directly communicate with the braking system. They would not have to invest in any in-house workforce for data-processing development, they would have to handle the incorporation of the system within the vehicle architecture. However, because of this "total-buy" approach, they may lack of flexibility and be less competitive with respect to other ADAS offered in the market. "Total-make" would not imply the production of the sensor itself, rather the integration of a bare radar and in-house development from raw data understanding up to the braking conditions. Intermediate approaches also exist and often represent a good compromise for car assemblers. Tier-1 suppliers offer equipment including bare sensor and a first layer of data processing. So-called "smart sensors" (detailed in Section 2.2) deliver measurements at higher level of abstraction (object-level or lane-level) which are easier to be managed and demand lighter integration effort for the car manufacturer.

1.2.2 Embedded computing

Expanding the sensor set implies having to deal with multiple data sources and flows of data. Simple solutions features low-cost and low-consuming microprocessors, integrated in electronic control units (ECU). Several ECUs are found in commercialized vehicle, some of them merely implement state machines (e.g. handling passenger window up-down status). For larger system architectures, it is be necessary to dispose of substantial computational power. Especially in need of vision algorithms to be implemented and consistently executed, specific systems-on-a-chip (SOC) are designed to be as performing as workstation computers.

A Electronic control unit

Embedded systems that are used to control one or more electrical system in a car architecture are called electronic control units (ECU).

The aggregation of multiple ECUs (up to hundreads of them) compose the "computer's car". Currently commercialized vehicles have not, in fact, a single computational unit. Single ECUs have few computational resources and different denominations in function of the part they control. Some examples are Brake Control Module (BCM or EBCM), Suspension Control Module (SCM).

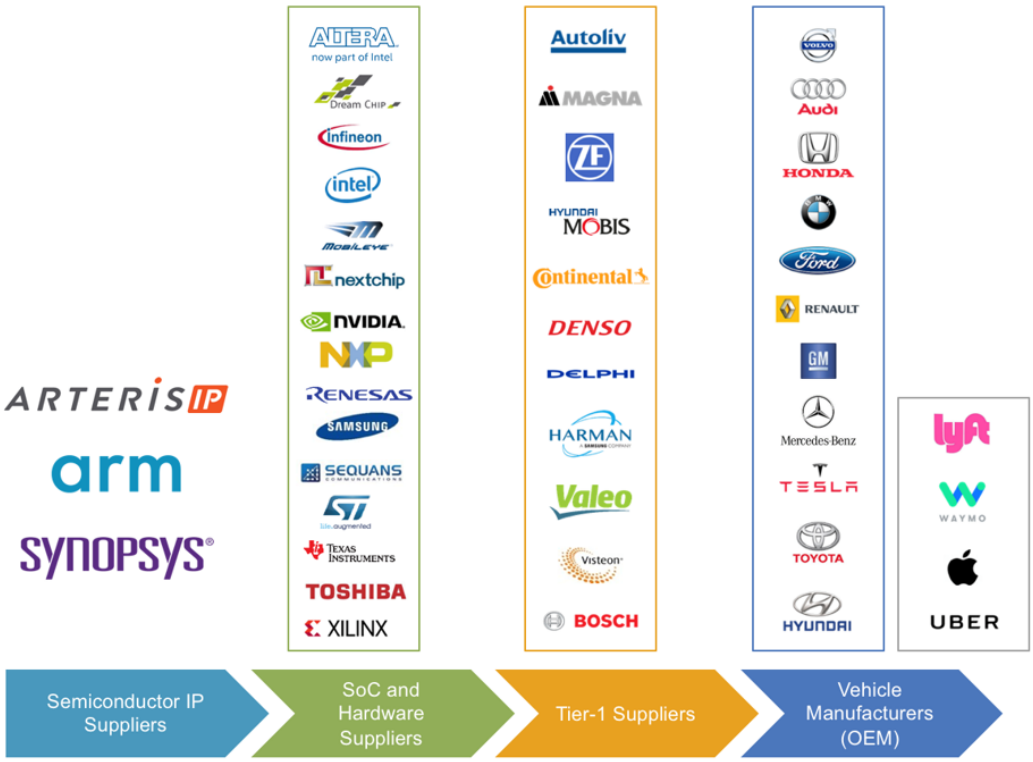


Figure 1.6: Automotive supply chain as illustrated by [Arteris, 2018] semiconductor supplier. This materials are used to build SoC and hardware which is used in Tier-1 solutions integrated in commercialized vehicles by car manufacturers.

B System-on-a-chip

As managing more and more ECUs with complex and sophisticated software is becoming impracticable, some of the most advanced OEMs are preferring a centralized and more poweful computing units. Systems-on-a-chip (SOCs), occasionally named autonomous driving chips, are specifcally designed to intake several data flows from different sensors such as multiple cameras and lidars. They typically integrate high-throughput interface for sensory inputs, neural processing units for efficient implementation of neutal netwroks, GPUs, computer vision processor and generally a large number of cores. Examples of commercial autonomous driving SOCs are Mobileye EyeQ series, Tesla FSD Chip, Nvidia DRIVE series.

1.2.3 Electronic control systems

Mechanical linkages are traditionally used for performing basic vehicle functions such as steering and braking. Electronic control systems, or Drive-by-wire solutions, can replace mechanical control systems using electromechanical actuators for both communicating with human-machine interfaces (pedals, steering feel emulators) or machine-machine interfaces, in the case of actuating ADAS or AD.

The embedded computing unit can, in fact, communicate with these actuators and implement automated control of the vehicle. Eliminating components such as pumps, belts, steering column and intermediate shafts can generally improve safety, ergonomics and reduce weight of the system.

A Throttle-by-wire

Vehicle propulsion is controlled with an electronic throttle which does not require any cable from the accelerator pedal. In the specific case of vehicles powered by electric engines, monitoring of the pedal transmits to the power inverter modules bringing sudden acceleration.

B Steer-by-wire

Electronic control units monitor the steering wheel input from the driver and control the direction of the wheels through electric motors, avoiding mechanical linkages in-between. Extreme implementations of steer-by-wire propose to replace the steering wheel with joystics or equivalent interfaces, yet these proposals currently had very limited commercial success. Electric Power Steering technology consists of a first step from mechanical steering which eases the process for the human driver.

C Brake-by-wire

Same as for Throttle-by-wire implementation, monitoring of the pedal with electronic components avoid the deployment of mechanicals. In this case, hydraulics and pressurized systems are replaced favoring responsiveness.

1.3 Thesis scope

The above introduction presented the context where our study is situated. This domain has been considerably active from both industrial and academic point of view. Incremental progress in each of the three areas mentioned in Section 1.2 is contributing to more advanced automation in the driving task and more easier accessibility to the technology. For example, the kind of on-board sensors named in Section 1.2.1 are just the major produced sensors in the industry. Currently underrated or scarcely known technologies are being developed and studied to potentially offer that specific environment understanding needed for the driving task. Similarly, in the so-called Computing area, technological progress may enable better data processing at lower costs.

The aim of this thesis is to propose a general solution for appropriately combining observations from exteroceptive sensors which is independent of the sensor technology. In the following, the interest of performing multi-sensor fusion in this context is outlined together with the framework where the thesis research is developed.

1.3.1 Multi-sensor fusion for lane detection

The above presented sensor technologies allow to on-board computer to build a representation of the environment which features both other road users and relevant road elements to the navigation. Supposing a fully operational sensor set, each device actively gives a distinct point of view on the state of the surrounding environment. Certainly due to the fact that each sensor is installed at a different position on the body of the vehicle, the main differences in those points of view are due to the specific perception technology.

In Fig.1.7, the table presents several usage of three exteroceptive sensors and whether they can perform well or not. As previously noted, cameras and lidars are very sensitive to weather conditions differently from radars. This technology may be completely ineffective for the detection of road markings, yet it is the only capable of perceiving through some occlusions. This complementarity among sensor technologies justifies and encourage the growing literature in the field of data fusion which has been lately included in commercialized solution for lane detection for automated driving.

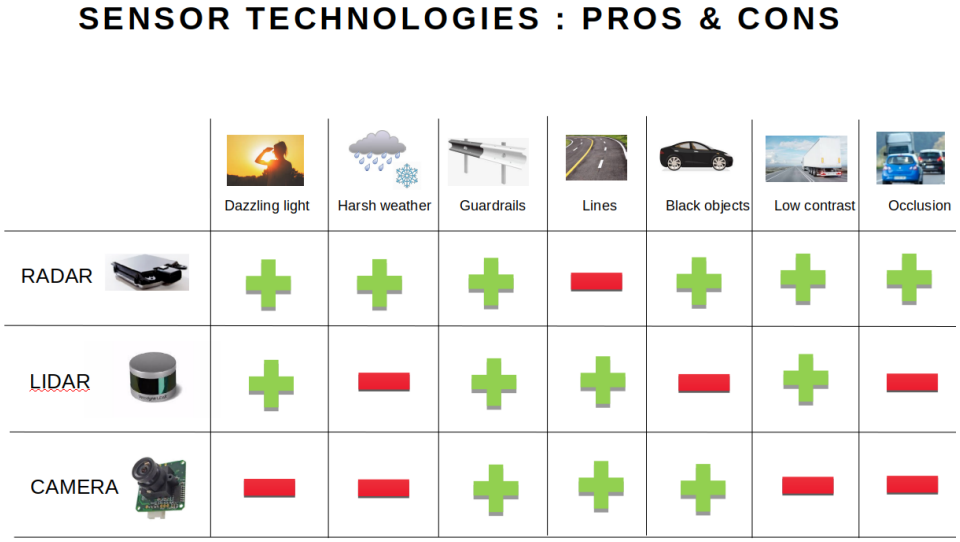


Figure 1.7: Comparison of sensor technologies [Derome, 2019]. Complementary advantages in using distinct technologies motivates research in sensor fusion.

1.3.2 Research framework

This PhD thesis is the result of a close collaboration between industrial and academic research in the field of autonomous driving. Under a Conventions industrielles de formation par la recherche (CIFRE) agreement granted by french Association Nationale Recherche Technologie (ANRT), Renault Group and Université de Technologie de Compiègne (UTC) co-managed this research project.

Heudiasyc Heudiasyc's research is in the field of information and digital technology (computer science, automatic control, robotics, and artificial intelligence).

From the start Heudiasyc has been closely allied to CNRS and is attached to CNRS's INS2I (Information Sciences) section. Heudiasyc disposes of a fleet of instrumented vehicles dedicated to experiments, developments and integrations of technologies for Advanced Driver Assistance Systems and Autonomous Driving.

SIVALab This work has been carried out within SIVALab, a shared laboratory between Renault and Heudiasyc (UTC/CNRS). SIVALab is a partnership between a research laboratory and a car manufacturer, with the intention of implementing a programme of research and innovation extending over at least four years.

Renault Group DEA-LEA1 This work is developed within the Renault's UET (Unité Elementaire de Travail) DEA-LEA1: Algorithmes Fusion et Véhicule Autonome. This team works across several projects typically associated to a vehicle model or to a class of models. Each projects defines the system architectures (e.g. available sensors, control units, computational power repartition and more) and a set of features offered available to the customer, such as AEB and ACC. These functions were used to be straightforwardly implemented through the integration of ADAS solutions provided by suppliers. For competitiveness reasons and because of the growing complexity of these functions, car manufacturers are now used to internally extend these packed solutions. This extensions consists here of a multi-sensor fusion layer which is ultimately demanded to identity the effectively meaningful targets to fulfill the user needs.

1.3.3 Structure of the manuscript

This thesis is organized as follows:

Chapter 2 formulates the problem, taking into account the industrial constraints and the target case study. At first, the existing kind of pipeline are presented before detailing the one chosen for this research. Thereafter, the state of the art of smart sensors for lane detection and map-providers is illustrated. The objectives of this thesis are then stated within a spotlight of the L3 industrial project which consists of the use case of this research.

Chapter 3 presents the onboard sensors solution for the estimation of lane boundaries. The deployed model for a generic smart sensor input is presented where probabilistc polynomials are used in the detection of road elements. The proposed tracking methodology is detailed in its main steps: initialization, prediction, association, update and adaptation to the clothoid spline road model. The solution is tested within the L3 experimental setup, both offline and on-vehicle. Qualitative and quantitative results are shown and discussed.

Chapter 4 presents solution with both onboard sensors and use of HD-maps. The adopted modelling of the map-provider input is presented together

with the tracking methodology. The proposed solution is tested within the L3 experimental setup. Qualitative and quantitative results are detailed and discussed.

Chapter 5 discusses general conclusions of the research work. Future perspectives are proposed, presenting preliminary results for those that have been investigated.

1.4 Contributions

The contributions of this thesis are listed and briefly introduced in the following. It is noted that the aforementioned thesis scoped entailed technical constraints on this work. This research have finally been oriented towards the addressing of the industrial use case with well-known information fusion methods, adopting elements of novelty in the modelling of the road elements and the involved uncertainties. End-to-end and deep-learning based solutions for lane detection have been investigated although not explored nor implemented. These choices of research are thoroughly discussed in Chapter 2 together with the state-of-the-art and the formulation of the problem. Hereinafter, the main contributions of this thesis:

- **Feature-tracking** (Section 3.3), used in Renault L3 experimentations, method for multi-sensor fusion of lane boundaries issued of smart sensors. This method models the single lane boundaries as a set of ordered road features, used to track position, orientantion and curvature at that point. This representation is updated when new smart sensor measurements are successfully associated. Each set of road features is then interpolated into a clothoid-spline, which is output to the next module in the pipeline.
- **Quantitative evaluation w.r.t. HD-map** (Section 3.4.2) of lane boundaries tracking methods in terms of lateral RMSE. In order to quantitatively measure the contribution of each sensor to the lane boundaries estimation, a comparison metric has been defined which takes into account the lateral distance.
- **Lane boundaries probabilistic association method** (Section 4.3) enabling measure-to-track pairing in our tracking proposals. Using road features projection and Mahalanobis distances, this method decides if two representations of lane boundaries refer to the same physical object.
- **Map-tracking** (Section 4.3), method for multi-sensor fusion of lane boundaries issued of smart sensors and map-providers.
- ***Precision* metric** (Section 4.4) enabling false-positives detection and multi-hypotheses localization scoring.

1.4.1 List of publications

F. Camarda, F. Davoine, V. Cherfaoui, B. Durand. *Multisensor Tracking of Lane Boundaries based on Smart Sensor Fusion*. IEEE Intelligent Vehicles Symposium (IV 2020), Oct 2020, Las Vegas, United States.

F. Camarda, B. Durand, F. Davoine, V. Cherfaoui. *Procédé de détection d'une limite d'une voie de circulation*. Renault/CNRS patent. Applied for European patenting at Institut National de la Propriété Industrielle (INPI) under the identifier n°2110938, Oct 2021.

Chapter 2

Problem formulation

This chapter presents the related field of research and finally details the problem we aim at solving. As mentioned, in the functioning of autonomous vehicles, a comprehensive knowledge of the surroundings is crucial. We are interested in the lane repartition of the roadway and we present at first where this knowledge is shaped within the "Computing" stage of an AV operation. Within this stage, a wide range of task breakdown alternatives is possible. We are presenting some relevant examples of pipeline together with the chosen one for our working case and the reasons that brought to this choice in our industrial working conditions. Selecting a pipeline defines the position of a Fusion module in the architecture. Hence, the inputs of this module are in-depth described presenting the state of the art for lane detecting smart sensors and map-providers. At this point where the context is completely defined and clarified to the reader, our mission and formulation of the lane boundaries estimation problem is outlined. Our objective of drawing the best of the multiple information sources about the environment also traduces in "client" (in term of pipeline modules) expectations. According to the intended usage of the final lane boundaries estimation, we tailor quantitative key indicators (such as driver comfort, estimated range extension or coherency) that would guide the definition of our data fusion solutions.

2.1 Autonomous driving pipeline

The developments toward autonomous driving technology have been heavily influenced by a same pace steady progress in artificial intelligence. Learned models has pushed the boundaries in the usage of sensor data but they also shaped valid alternatives in defining the system architecture of autonomous vehicles.

2.1.1 Learned architectures

Since [Pomerleau, 1989], proposals of training a neural network which directly mapped output of the sensors onto control signals of the vehicle have shown their potential. These solutions, so-called end-to-end, are relatively easy to implement and obtaining labeled data for their training as immediate as recording a naturally driving. A general schema is illustrated in Fig.2.1 and some examples

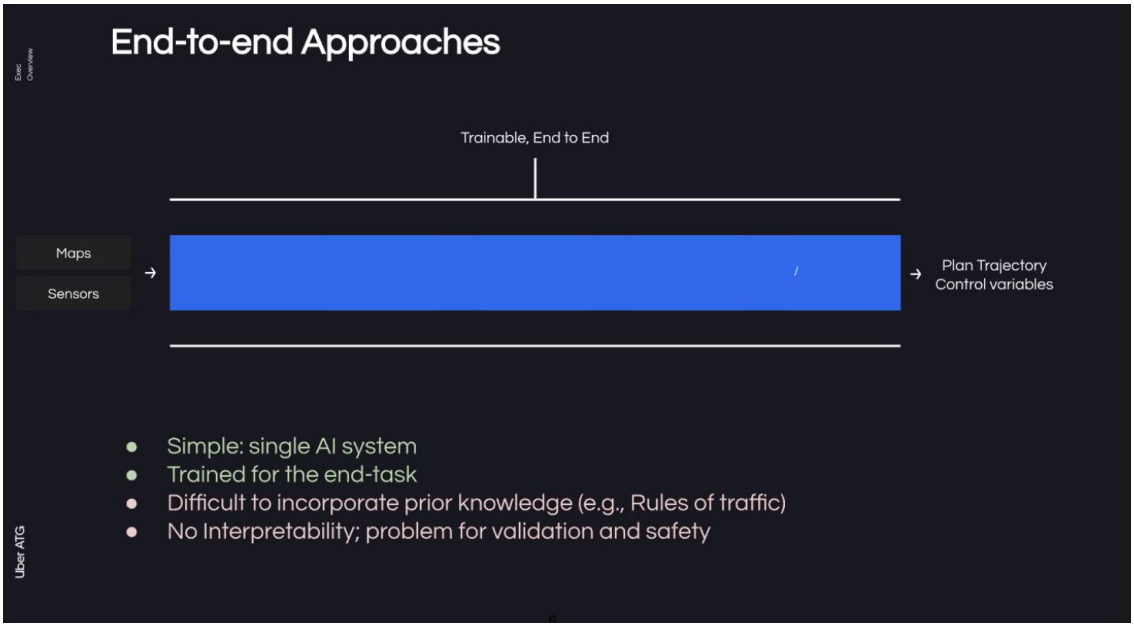


Figure 2.1: End-to-end [Urtasun, 2020] pipeline are completely learned architectures which are simple to be trained on the end task but hard to diagnose in case of (likely) issues in the general task.

are described in the followings. More recent years developments in graphic cards technology and further works from NVIDIA researchers [Bojarski et al., 2016] brought to working learned models of the road following task without necessarily decomposing it into lane detection, semantic abstraction, path planning and control. This simplification would be ideal and unburdening for the engineering of the AD problem, yet end-to-end deep learning-based approaches did not turn out to be the universal panacea for autonomous driving. In fact, the self-driving control of a vehicle generally consists a sequence of decisions and each step should be modelled in the automated system. Also, condensing the mapping between image data and steering/acceleration into a single neural network heavily struggles in the interpretability and diagnostic of faulty behaviours. These issues can not be fixed with larger training dataset or models, more advanced approaches propose architectural adjustments to face this.

Waymo's [Bansal et al., 2018] opted for mid-level learning where it has been taken better advantage of perception and control components. In this case, learning from both simulated and on-road recorded data is possible as the inputs of their recurrent neural network (RNN) are now abstract representations of the road agents rather than pure sensor data and directly output the control command.

Uber's [Zeng et al., 2019] sticks with the end-to-end deep learning-based architecture yet proposing interpretable intermediate representations such as 3D detections and explicit trajectories.

2.1.2 Traditional autonomy stack

It can be understood that vivid activity in the scientific community is pushing for learned architectures, which would be preferable for easier implementation and generally less burdensome developer cost. However, within the industrial context, a traditional engineering stack for the development and commercialization of vehicles enabled for (partial) automated driving seems to prevail. This configuration seemingly supports validation of the system's functional safety which is crucial for any commercialized product that involves human safety. The scope of functional safety for automotive applications has been defined by the International Organization for Standardization (ISO) which published the ISO 26262 standard [ISO, 2018]. Target systems are characterized in terms risk in Automotive Safety Integrity Levels (ASIL), which scale from A to D grading from the least through the most critical.

In the scope of this work, as developments took place within an industrial context, the system architecture of reference is traditional and linear. This approach encourages modular developments where engineering contributions and intervention in debugging and diagnostic. An example is given in Fig.2.2 and described in the followings. From raw sensor measurements to control instructions, this pipeline features the following key phases:

- Sensors/Maps: observations (Section 1.2.1) of the surroundings are taken as input, both regarding static and dynamic entities. Accordingly with the working conditions of the sensors, frame of reference transformations may be necessary before heading to the next step.

- Perception: raw observations may confirm consistent detections of road agents or elements. In this case, tracks are defined which identifies position and other relevant proprieties of these entities.

- Prediction: according to evolutional models of objects and their estimated proprieties (or state variables), track representation allows for a prediction of their behaviour in the future. On the one hand, mechanical models enable short-term motional predictions in the order of few seconds. On the other hand, more complex and behavioural models of the agents provides insights on the qualifying path alternatives that may be considered (e.g. likelihood for another vehicle of turning right or maintaining its route at an intersection).

- Planning: retrieved information about the driving scene and deducted models of the players within it are the foundation for defining a safe future trajectory which is complying with the specified destination. Motion consists of planning the next move of the ego-vehicle, this can be done at different levels of abstraction. At navigation level, the road network has to be considered for targeting the correct turns and macro-choices for finally reaching the target. These are the kinds of instructions we are currently used to follow, when manually driving with the aid of navigation systems. As we know, road design data is crucial for this process and real-time information

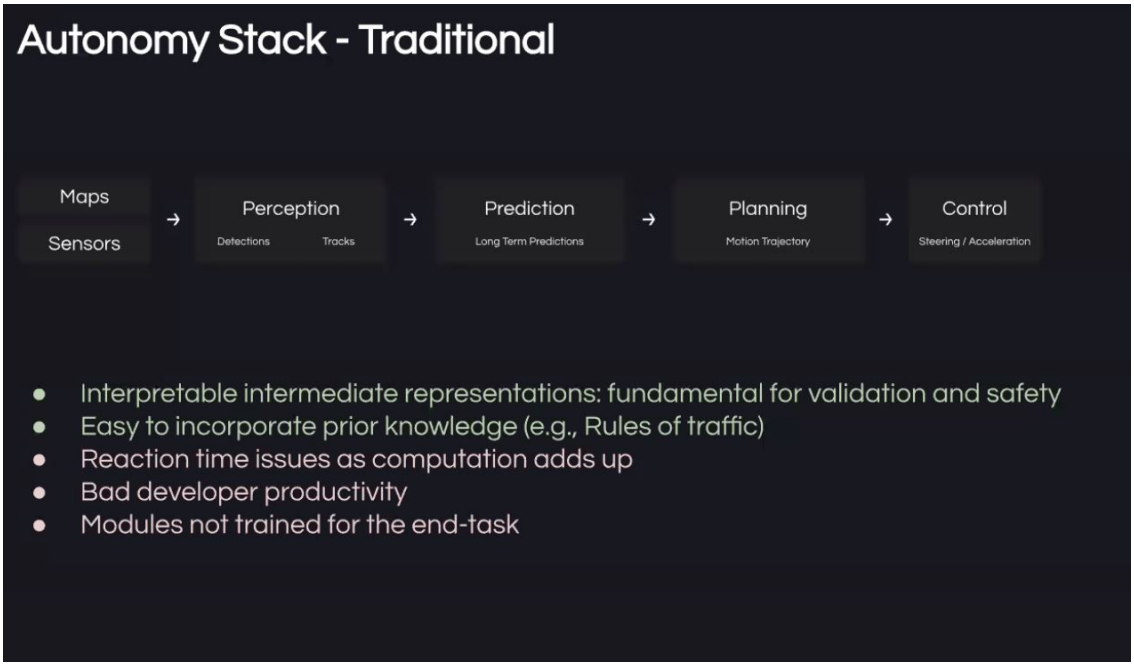


Figure 2.2: Traditional pipelines [Urtasun, 2020] require significant engineering effort but modularity eases prior knowledge incorporation and troubleshooting.

on traffic conditions may also affect our decisions at this level. At maneuver level, behaviour for close time horizon is planned. In the short-term, it might be convenient to overtake a slower vehicle or to perform a lane change. These lower-level-of-granularity kind of decision can be taken independently of the final destination. At last, for an even shorter (order of milliseconds) horizon of time, at immediate motional level, control of steering and acceleration in the next instants is sized to offer comfortable drive to the passenger.

- Control: depending on the specific technology of electronic control installed on the intelligent vehicle (Section 1.2.3), instructions at steering/acceleration level are executed with the lowest possible latency. This causes, evidently, to move the ego-vehicle at a new location. Together with the displacement of the other agents in the environment, this causes the sensors to perceive changes in the driving scene and to the pipeline to be execute again in the next iteration.

This kind of architecture shows noticeable advantages in diagnostics and malfunctioning fix. As example, it may occur that an issue in the pipeline is causing the brake control to instruct frequent decelerations of the vehicle. Probing the individual output of each of the key phases in the stack may unveil that the estimated velocity of all the other road agents is constantly being assigned a null value, which causes them to being perceived as static objects. This kind of tracking issue can be individually addressed in the Perception module, eventually avoiding incorrect inference in Prediction, Planning and unnecessary breaking. Reasoning of this sort in the development of automotive applications may not be

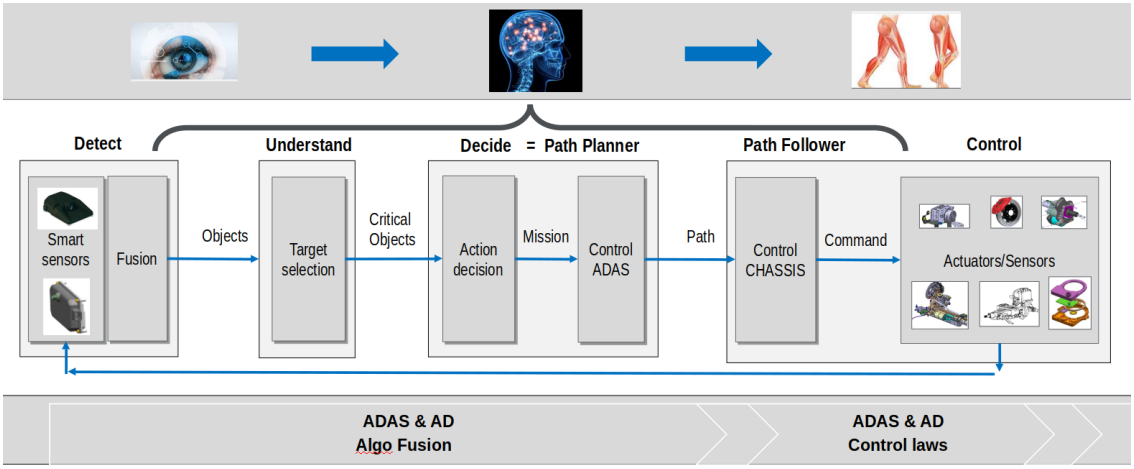


Figure 2.3: Adopted pipeline [Derome, 2019] with integration of smart sensors and a specific module for track-level fusion for both objects and road elements.

possible in the use of learned models, especially in hardly interpretable end-to-end deep learning-based architectures.

Chosen pipeline The above presented general autonomy stack sets the guidelines for the definition of the specific pipeline for our working case. Additional constraints are taken into account in this definition, which may vary from industrial regulations and relations with suppliers to practical work repartition in the company departments. The decision of adopting this same pipeline for this research work is motivated by several arguments. As first, the most practical, being conform to a previously defined architecture allows for the implementations to be carried on in a well-established environment. Conforming to the same development tools enable for a large set of ad-hoc utilities and recorded data to be directly available. Consequently, as we will be working within the first two module referred to in the architecture, the fact that existing versions of the following modules are being developed and compatible with our work enables debugging and analysis of our impact on the whole architecture.

Let us now detail, from raw sensor measurements to control instructions, the specific key phases of the adopted pipeline which is illustrated in Fig.2.3:

- Smart sensors: this is one of the major constraint applied to this research when adopting this scheme. As Section 1.2.1.F recalls, integration of ready-to-use solutions from Tier-1 suppliers is often easier for OEMs than dealing with specific raw sensor data. As this situation precisely corresponds to our working case, so-called smart sensors (and map-providers, for navigation information) will be the very first input of our pipeline. The main difference lies in the fact that with "pure" sensors we would be in control of the whole data processing. The use of smart sensors delegates handling of raw data in return of more abstract yet more practical abstraction of the elements in the driving scene. As example, in case our architecture is including a frontal camera for the detection of road markings, we will not

be dealing the pixel matrices issued of the camera observations. The smart camera, and from there the definition of "smart", will be performing image-processing-specific transformations in order to present its observations as a straightforward list of road elements, stepping up of a level of abstraction.

- Fusion: combining data from different point of view and sensors of different technology is key for a complete and reliable understanding of the environment. This specific module will be implementing the proposed solutions from this research. Information fusion can be performed at different level of abstraction. Because of the "smart sensor hypothesis", it will be best suited for applying of techniques of information fusion at track-level. Eventually, a comprehensive list of all objects and lane boundaries detected and tracked in the environment will be set up for the next module. Each element shall be presented in its state at a coherent and agreed time instant. Aware that a communication delay "Computing" and "Actuating" phase exists and may affect the motion planning, the future state of the scene elements is accordingly predicted through adequate motional models.

- Target selection : any of the objects and road elements delivered to this moduled has been confirmed to be detected in the scene with a certain amount of confidence. However, not all of these detections are critical in the immediate future, from the motional point of view of the ego-vehicle. Because of this, a sub-set of relevant targets is selected according to specific ADAS-related criteria. For example, in the simple case of AEB implementation, the closest ahead vehicle is eligible in the target selection if it is as close as its predicted trajectory collides with ego-vehicles's in the next few seconds. Critical target does not necessarily mean threatening, it refers to the need of correctly observing given elements in order to put into action a functionality. Left and right lane boundaries are generally key for any LKA and LCA implementation.

- Action decision : before defining and applying action on the vehicle actuators, the selected targets contribute as determining factors for deciding the ego-vehicle's next action. A mission to be followed is here implied always relying on specific criteria on the state of selected targets. At this stage, the calibration of individual thresholds may drastically affect the behaviour of the vehicle. As in case of series production it affects millions of vehicles in the road network, specific protocols are defined in the automotive industry in order to fairly test safety functionalities. At European level, these validation protocols are proposed by the European New Car Assessment Program (Euro NCAP).

- Control ADAS : the mission would generally imply active response in the vehicle system. Electronic systems may require to be solicited in some cases, such as turning signal or alerting in the Human Machine Interface (HMI), yet the most impactful action to be taken is determining the exact path for the vehicle to be followed. In order to facilitate comfortable driving for the passenger, the path for the successive few seconds is planned and delivered.

- Control chassis : planned path is rendered into specific actions and electronic commands delivered to the actuators, which may vary depending on the vehicle model, architecture or propulsion technology.
- Actuators : instructions at steering/acceleration level are executed which causes to move the ego-vehicle at a new location and to perceive changes in the driving scene and the pipeline to be execute again, as in the general definition of autonomy stack

Related state-of-the-art As it has been seen from the presentation of these pipelines, perception is an essential stage in the development of autonomous driving systems. It precedes and lays the basis for path planning and decision making. In this key phase, the goal is to pursue two main outcomes: an overview of other users in the driving scene and a solid representation of the surrounding environment. The latter is tackled in this work, specifically proposing information fusion approaches. For this reason, it is important to elucidate as much as possible on the functioning of the inputs to the fusion module. In the following, the state of the art of smart sensors for lane detection and map-providers is presented before ultimately providing the formulation of the problem targeted in this research.

2.2 Smart sensors: state-of-the-art

An extensive knowledge of the lanes repartition around a vehicle is necessary in order to enable it to perform autonomous navigation. In the lane detection problem, the aim is to identify the number of the surrounding lanes and their geometric shape. This has to be estimated with respect to the ego vehicle, which will base its navigation decisions on this representation. This approach to the problem is centered on the vehicle and its only aim is to perform at best with available priors and data from propriospective and exterospective sensors. Different approaches based on different levels of prior knowledge exists such as precise localization approaches. In this case, the work of [Poggenhans et al., 2018b] is a valid example, the lane detection problem is addressed by localizing precisely (with centimetre accuracy) in the frame of reference of a so-called high definition map, a lane-level representation of the environment. In this work and others [Li et al., 2017], the lane detection problem is solved accessing the geographic representation of the environment and querying this prior knowledge (usually stored in the form of a database).

In the usage of smart sensors, however, this prior is generally not available. The chosen approach therefore generally follows the structure proposed in [Huang and Teller, 2011]. Here, some main steps are defined to describe the synthesis of an accurate lane representation on the basis of raw sensor data. The choices made in these steps determine the final result, the working conditions and the scope of the possible application. An overview of the process, as proposed in [Huang and Teller, 2011] and its illustration (reported in Fig. 2.4), can be given as follows:

1. Lane model
2. Features extraction
3. Lane detection and tracking

These three main axes of the problem are taken as guidelines for the exploration of the literature, without forgetting the above presented industrial context. In fact, it is recalled that our solution would ideally exploit and combine existing commercial smart sensors. Also, the designed data-fusion architecture would be itself applicable, not too far in the future, to a marketed product or vehicle. Accordingly, this work will mostly focus on lane modelling, detection and tracking. Different feature extraction techniques are nevertheless examined in order to be optimally exploited, in the form of smart sensors (e.g. MobilEye Camera and other).

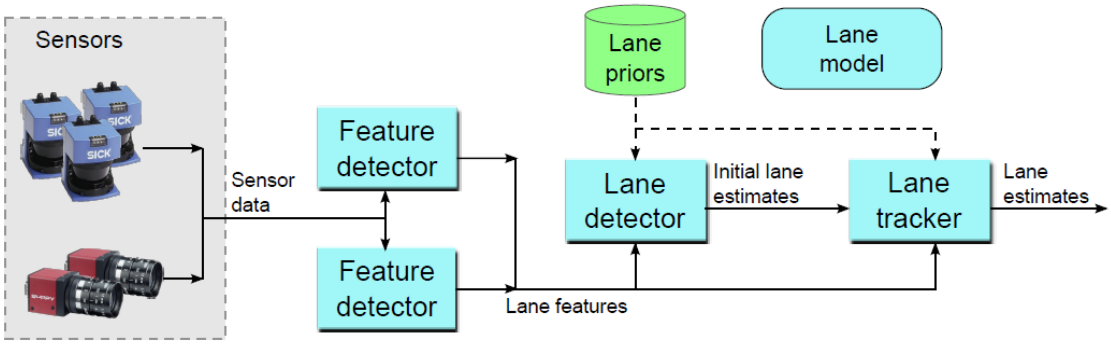


Figure 2.4: Lane detection pipeline [Huang and Teller, 2011].

2.2.1 Lane model

The choice of the lane model affects which types of lanes can be represented and the applicable estimation algorithms to fit it. Lane and road models can be categorized into three classes: parametric models, non-parametric models, and semi-parametric models.

1. **Parametric models**: a finite number of parameters maps into simple geometric curves. Common representations are straight lines, parabolic curves, circumferences arcs and hyperbolas. This first category has limited degrees of representation but it is suitable for efficient fitting algorithms such as RANSAC, Hough transform and vanishing point [Kong et al., 2009]. This enables simple applications but strong outlier and noise resistance, such as [Zou et al., 2019] which exploits RANSAC and Kalman Filter with a straight line model. On the other hand, if the intended application is anything more complex than an LKA (Lane Keeping Assistant), more flexibility is needed and strictly parametric are not a viable solution.
2. **Non-parametric models**: without an a priori specified structure the result is determined by the data. It may consist of a continuous representation

(e.g. continuous pixels), but it might not contemplate any smoothness or differentiability. This paradigma is preferred in all of those applications where features are issued of image matrix and extracted directly from a camera point of view, generally based on deep learning [Pan et al., 2018].

3. **Semi-parametric models**: in the presence of parametric and non-parametric components, these models do not assume any specific global geometry for the path. This category primarily includes different declinations of piecewise defined functions, so-called splines. The work presented in [Fatemi et al., 2014] has constraints quiet similar to those affecting our study. Relying on smart sensors, a clothoid-spline model is chosen guaranteeing the continuity of the curve's curvature all along the different segments. This model presents flexibility and interesting results, even thought the model proposed in [Fatemi et al., 2014] is based on strong and facilitating hypotheses. Moreover, the results from Bertolazzi [Bertolazzi and Frego, 2015] have been analyzed to evaluate if they could possibly benefit a clothoid-spline based lane tracking algorithm such as the latter. On the other hand, [Abramov et al., 2016] presents an even more generic lane model, which also implements constraints and priors between lanes (parallelism, minimal lane width, etc.) as form of edges in a Graph-SLAM approach.

2.2.2 Features extraction

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

For cameras and vision-based sensors, several existing works focus on extracting features based on colours, shapes and textures. The classic work [Bertozzi and Broggi, 1998] bases its lane detection module on the horizontal brightness variation of the input stereo-image, after the application of an Inverse Perspective Mapping (IPM) transformation. Lately instead, [Pan et al., 2018] represents the cutting-edge for lane detection, according to [Bai et al., 2019] and to the TuSimple Benchmark Lane Detection Challenge [TuSimple, 2021]. Its deep learning approach takes into account wide spatial relationships and results effective on extended objects with few appearance clues, i.e. traffic lanes, poles, and walls. [Lee et al., 2021] achieves state-of-the-art performance in CULane and BDD100K and distinct improvement on TuSimple dataset. [Xu et al., 2020] proposes a novel lane-sensitive architecture search framework named CurveLane-NAS to automatically capture both long-ranged coherent and accurate short-range curve information.

Other proprieties are considered with sensors of different nature, such as reflectivity of the material for lidars [Kammel and Pitzer, 2008] and echoes coming from road barriers for radars [Kim and Song, 2016]. In [He et al., 2016], lane marking features are detected according to intensity of the laser reflection, transformed into an intensity image and classified by the means of a Convolutional Neural Network (CNN). [Ghallabi et al., 2018] bases his work on the same fea-

ture, with a simple straight line model in order to enhance the cross track localization of the vehicle.

2.2.3 Lane detection and tracking

Selected features are processed in the detection step to infer the presence of one or more lanes. Features are fit into the chosen lane model and an initial estimate is generated. If these detections are successively confirmed by fresh measurements, they are validated as actual tracks and at tracking stage we can exploit spatial and temporal continuity constraints to update the lane estimates as the vehicle moves and new observations are available. In this part, noisy detections have not to be confirmed as tracks. The algorithm resistance to outliers is tested, as well as its reaction to false positives. Additionally, new observations of already detected lanes have to be correctly associated to existing tracks and contribute to their estimate and update. In these steps, the representation of uncertainties plays a crucial role when classical association techniques are applied. [Hasberg and Hensel, 2008] estimates the geometry of a railway track using a spline model and localization samples from GNSS receivers installed on the trains. Each point of the curve has an associated uncertainty, which is taken into account in the update step of its Kalman Filter.

2.2.4 Track-level fusion

As smart sensors typically implement and deploy tracking algorithms, performing information fusion on their outcome would imply performing a form of track-level fusion. These methods applied to lane detection have generally gotten less attention in the academic sphere with respect to raw data fusion solutions. However, they can be particularly useful in technical and industrial situations such as the one we want to address. Specifically, recent developments from [MATLAB Sensor Fusion and Tracking Toolbox, 2021] eased accessibility for the development of track-level solutions. In Fig.2.5, the outline of a track-level fusion example for objects detection. The schematic of the workflow shows how radar and lidar measurements are respectively processed in an extended object tracker with Gaussian mixture probability hypothesis density (GM-PHD) and in a conventional joint probabilistic data association (JPDA) tracker configured with an interacting multiple model (IMM) filter.

Relevant works from the literature such as [Houenou et al., 2012] present a modular high level track-fusion architecture for a multisensor environment. Here the multisensor track-to-track association issue is addressed with a particular track-to-track distance computation. [Duraisamy et al., 2013] presents an overview of track level fusion algorithms such as fusion with Cross Covariance, Information Matrix fusion, Reconstructed Measurements method and Covariance Intersection. In these cases, several metrics are designed for comparing performances. Generalized Optimal SubPattern Assignment Metric (GOSPA) metric from [Rahmathullah et al., 2017] allows to penalize localization errors for detected targets and the errors due to missed and false targets.

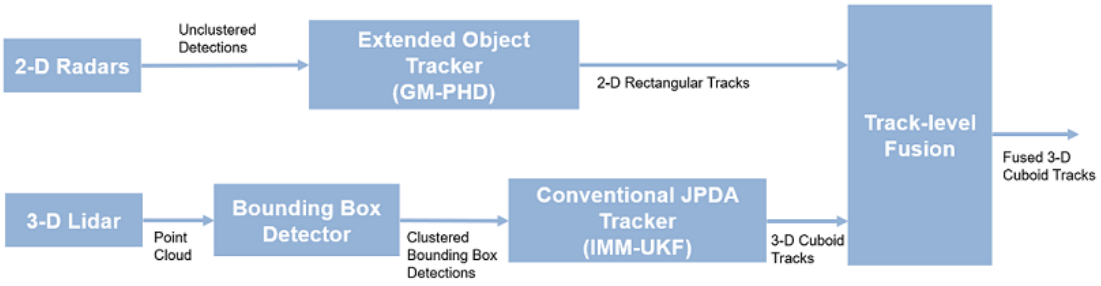


Figure 2.5: Example workflow from [MATLAB Sensor Fusion and Tracking Toolbox, 2021] featuring radar and lidar track-level fusion.

2.3 Map-providers: state-of-the-art

Similarly to what have been done for smart sensors, we will be describing the functioning and current state of the art of another important input of the Fusion module: the map-provider. Using the term "map-provider" we identify a complete system which is able to deliver (or provide) a representation of the surroundings of the ego-vehicle taking into account its current position. This representation is provided in form of structured data and generally extracted from a larger and more complete map. The aim is to exploit as much as possible this additional clue about the road network structure to support automated driving functionalities. Again, as for smart sensors, car manufacturers have the possibility to buy and integrate ready-to-use map-providing solutions choosing in a rather large market of specialized suppliers. Expertise for on-board delivering exploitable road maps for ADAS and AD are specific and require considerable amount of dedicated upstream work for tasks such as map building and maintenance. The main players in this sector, such as TomTom and HERE, are currently focusing on the creation of High Definition (HD) maps of all those sections of the road network where automated driving features are most likely to be allowed from the regulators. Fig. 2.6 shows a preview of TomTom's Road DNA [TomTom, 2019] which aims at characterizing meaningful information from the roadway in order to facilitate precise localization.

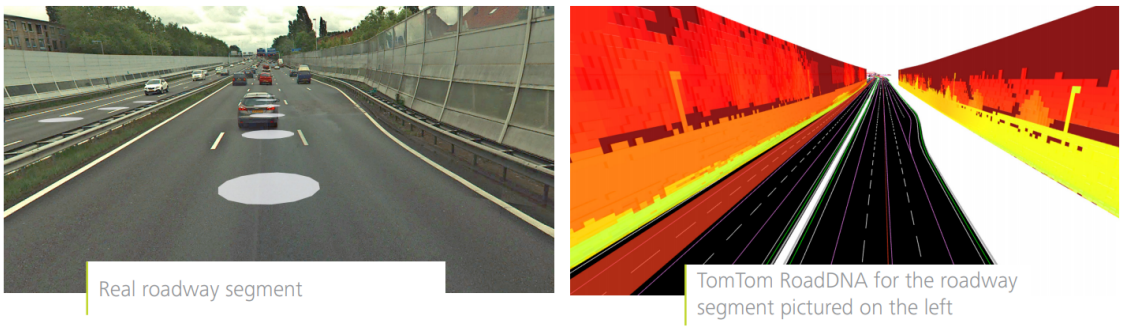


Figure 2.6: TomTom's Road DNA [TomTom, 2019] aims at characterizing meaningful information from the roadway in order to facilitate precise localization

2.3.1 Mapping

Since the early usage of paper or cassette-stocked maps [Newcomb, 2013], disposing of navigation maps has always been of great support to the driving task. Where most electronic maps allowed to be updated according to changes in the road networks, they all differed for specific characteristic such as the level of detail or the stocked information within the map. In order of chronological in-car availability and growing accuracy (Table 2.1) enabled by technological advancement, three main categories of digital navigation maps are presented in the following, all of which are milestones in the driving assistance progress.

Year	Map Type	Geometry	Accuracy
1930	Paper maps	Two-Dimensional (2D), at road level	
1990	Digital maps	2D, at road level	5–10 m
2000	Enhanced digital maps	Digital maps (including ADAS support)	50 cm
2010	HD Map	3D, Digital maps (at lane level)	10–20 cm

Table 2.1: In-car navigation map evolution [Liu et al., 2019].

A Digital Maps

Traditional digital maps were the first evolution step from in-car use of paper maps [Leite, 2018]. Interactive and accessible through in-dash screen, they allowed search functions and turn-by-turn guided navigation. The road network is represented as a 2D-graph where the shortest path from departure to destination is calculated and the driver is instructed through each manoeuvre, while their position within the map is know thanks to Global Navigation Satellite Systems (GNSS). Global Positioning System (GPS)-based localization is sufficient as the level of precision of the map is generally in the order of 5-10 m and the representation of the navigation path stops at road level.

B Enhanced Digital Maps

More advanced maps became useful for ADAS applications as additional information on the road network integrated the electronic map. The environment is still represented in two dimensions, however road connections can be detailed at lane-level while including specific attributes such as [TomTom, 2019]:

- Road curvature: can improve passenger comfort advising (or enforcing) on the appropriate speed and predictive cruise control.
- Gradient: or slope of the road, can improve fuel efficiency and safer usage of braking system (to be managed particularly carefully onto heavy vehicles in down-sloping path).
- Curvature at junction: knowledge of the "angularity" of a junction can support passenger comfort in safety conditions.

- Lane at junction: information on lane repartition at/before a junction elucidates on potential turning opportunities.
- Traffic signs: both existence and spatial position of important signage can support ADAS functioning and AV road segment localization.
- Speed restrictions: can enhance ACC.

These type of maps can reach a granularity of up to 50 cm and potentially support automated driving applications. With respect to their even more detailed counterpart, they are also called Standard Definition (SD) Maps. Unlike digital maps, that were usually designed to solely support the turn-by-turn navigation system, SD Maps may need to be transmitted to one or more ADAS ECUs on the vehicle implementing specific features on the basis of map data. For this purpose, the Advance Driver Assistant Systems Interface Specifications (ADASIS) Forum [ADASIS, 2002] has defined an ad-hoc protocol for easing the communication with on-board map-providers avoiding the usage of a different proprietary language with each different map. In the case of SD Maps, the ADASIS v2 Protocol [Ress et al., 2008] has been defined and it is currently studied [Bhonsle, 2016] and implemented in commercialized vehicles.

C High Definition Maps

Currently the most detailed format of navigation maps for automotive use, High Definition (HD) Maps represent the world in three dimensions and usually implement several layers of information. Several map makers have defined their own kind of HD Map, with different terminology and proprietary data structures, however most of them established an information hierarchy over three layers with similar meanings.

In the following, the naming proposed by [HERE, 2017] and displayed in Fig. 2.7 is adopted for describing the content of each layer, from the most general to the most specific:

1. **Road model.** An ordered sequence of shape points represents the geometry of the road. Start and end point of distinct roads are used to describe intersections. Increased density of shape points can enhance the curvature description of the road, however it would imply a larger storage of information. The better trade-off is held with a mathematically efficient representation where a new shape point is used in the presence of appreciable curvature variation. This layer can be extended with additional information, tending to the content of whole SD Maps. It is generally used at strategic level of navigation planning. The most convenient path to destination can be computed taking into account real-time information, where available, on current traffic and road conditions.
2. **Lane model.** Early implementation of 2D lane models, in the format of Road Network Description File (RDNF), have been used in the famous Defense Advanced Research Projects Agency (DARPA) Urban Challenge already [Buehler et al., 2009]. More recent realizations, as in the Bertha Drive

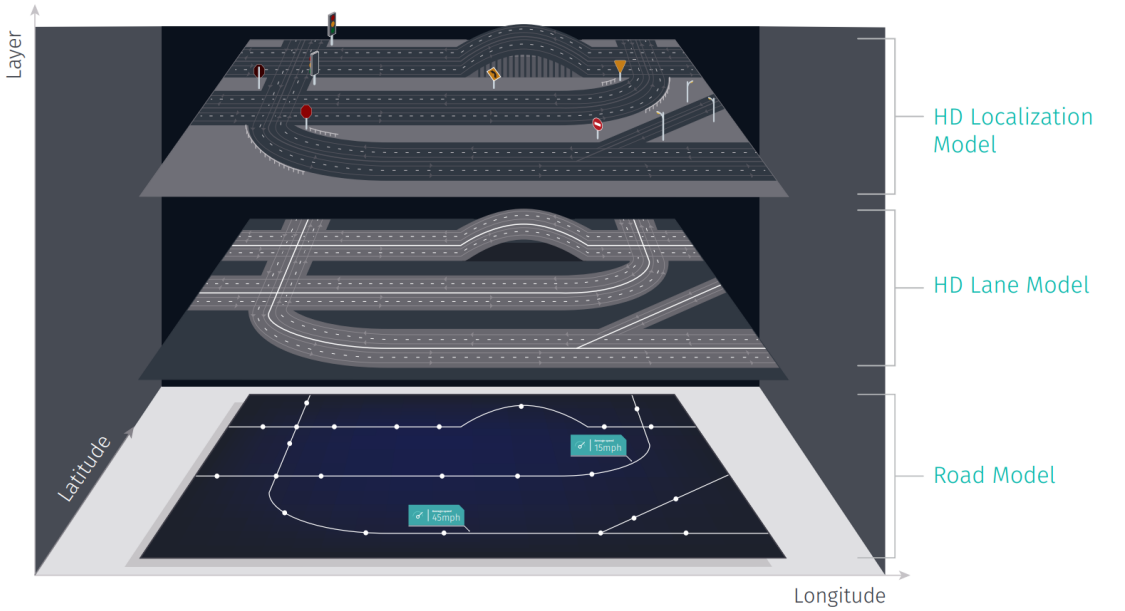


Figure 2.7: HERE three-layers HD Map [HERE, 2017]. From Road to Lane up to Localization data, the HD Map supports and enable advanced driving assistance

project [Ziegler et al., 2014], implement 3D lane models in the form of *lanelets*, which have been specifically designed to describe drivable sections in all their characteristics. A lanelet [Poggenhans et al., 2018a] contains the following information: highly accurate geometry model, lane attributes, traffic regulations, road furniture and parking, lane connectivity. In Fig.2.8, an example of lanelet is presented. At this level of abstraction, left and right bounds are analogous to the lane boundaries this research is focused on. They describe the geometry of the lane limits and integrated the aforementioned semantic information. This map layer is designed for navigation planning at manoeuvre level, performing lane-level operations such as changing lanes in anticipation of an intersection.

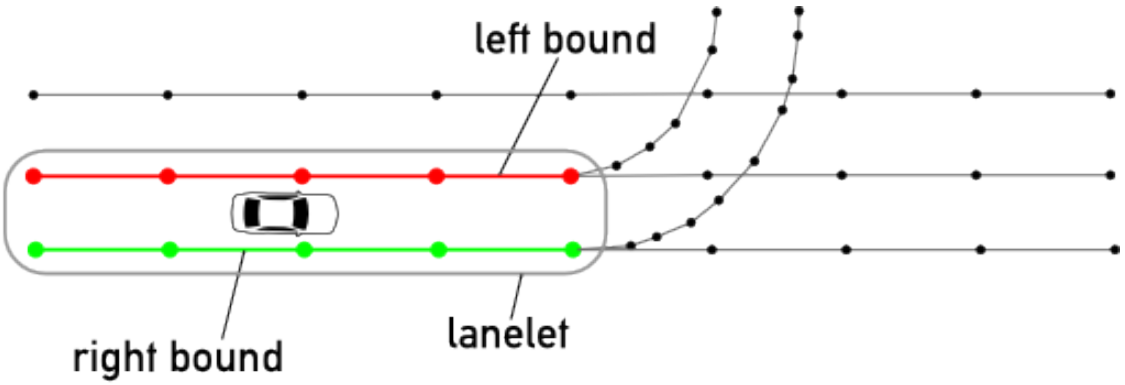


Figure 2.8: Lanelet example. Left and right bounds described the geometry of the lane boundaries and integrate additionaly attributes such as semantics and connectivity

3. **Localization model.** At this layer, the high definition of the map finds expression for supporting the precise localization of the vehicle within the map. Specific methods for this process are discussed in the followings (Section 2.3.2), yet we can introduce two distinct categories of road elements which entail distinct localization models. On the one hand, a feature-based localization model can be stored as a graph where each node represent a 3D landmark in the environment described by feature descriptors. Feature maps may be efficient and compact but they rely on the fact that data processing and feature extraction from sensor data can be done both offline, in the map making phase, and online. The more are the available landmarks stored in the feature map, the more the localization method can be effective. Lane boundaries with their type attribute (stop lines, curbs, solid, dashed), traffic signs, highway reflective markers can all be used for this purpose. On the other hand, dense information-based localisation model are generally more demanding in terms of memory. They can stock sensor data in 2D spatial grids (which can extend over to 2.5D with reflectivity and height information) or in view-based representations. The latter is the one preferred by some of the most advanced automated driving companies such as Waymo [Sun et al., 2020] which enable their robotaxi fleet to level 4 autonomous driving, albeit in only few selected areas of USA cities. This model can in fact deliver outstanding accuracy in map relative localization but it requires for lidar-based point cloud map to be stored and maintained. Voxelization of point clouds [Vo et al., 2015] may help in stocking the data more efficiently, yet keeping it up-to-date remains a major challenge. Many elements in the driving scene can vary or be perceived differently under different weather or seasonal conditions. [Maddern et al., 2015] even introduced the concept of Experience Based Navigation (EBN) to address this problem in the usage of point clouds.

The use of HD Maps in commercialized vehicles is currently not usual but its spread might not take too long. ADASIS proposed an extension of its protocol (releasing v3), supporting their integration and pushing the idea of electronic horizon (e-Horizon) in assisted driving. Taking advantage of detailed representation of the environment which are not limited by occlusions or maximum sensor range has proven to be an enabling factor in higher level of autonomy. Challenges remain for the maintenance of this framework while numerous objections have been raised to the need of HD-mapping every single road on Earth. Either way, some map makers (e.g. TomTom) are already moving in that direction and offering HD maps of most highways to their customers car manufacturers.

Creation process of HD Maps Before reviewing the existing localization methods, it is worth discussing how maps are created, in the case of HD Maps. The basic idea is to collect geo-referenced and timestamped sensor data in order to be able to aggregate and extract the appropriate localization model. Fleet of mapping vehicles are typically used which are generally equipped with lidars and cameras for environment perception and with Inertial Measurement Units

(IMU), GNSS receivers, proprioceptive motion sensors such as wheel odometers for tracking the position of the mapping vehicle.

Rather than using dedicated mapping vehicles, emerging approaches of crowd-sourced HD Map are being proposed in the AV industry. [Mobileye, 2017] introduced Roadbook and Road Experience Management (REM) to leverage on the millions of smart camera equipped vehicles for updating in real-time the shared and feature-based HD Map. Relying on the "wisdom of the crowd" could eventually be more reactive and reliable than adopting lower frequency updates delegated to few over-equipped vehicles, if the amount of data to be uploaded from the vehicle is as modest as promised (10kb/km) and the human supervision effort is also limited.

2.3.2 Localization

Awareness of its own position is essential in the context of automated and assisted driving where the ego-vehicle aims at correctly and intentionally navigating the environment. The approaches to this problem, which is the estimation of the observer position within a frame of reference, are based on many of the concepts coming from the world of robotic navigation. Unlike use cases of indoor navigation, where autonomous robots can rely on fixed beacons and active sensors installed in the environment, the localization of an intelligent vehicle is addressed outdoor and using proprioceptive and exteroceptive sensors mounted on-board. Intelligent road equipment and communication vehicle-to-infrastructure (V2I) consists of a valid analogous to indoor localization [Chen et al., 2015], yet this case is not discussed in the following as vehicle-to-everything (V2X) solutions are currently far from mainstream adoption in the automotive industry. Hereinafter, approaches for vehicle localization are presented reviewing existing solution to this problem in two distinct cases: position tracking and global localization.

A Position tracking

This is actually the case of most commercialized driving assistance applications. At vehicle ignition, the starting position is supposed to be known. Its evolution wants to be correctly observed to understand the relative positioning with respect to other entities in the driving scene, whether they are static or dynamic.

The displacement of the ego-vehicle can be estimated comparing two consecutive measurements from the same sensor. In case of proprioceptives, the change in the pose can be computed via direct integration of accelerometers, odometers and gyroscope data. The solution to the dead reckoning navigation problem is a traditional approach to position tracking [Chung et al., 2001] [Park et al., 1997].

On the other hand, estimating the same displacement using exteroceptive sensors can be more challenging and, dependently on the specific sensor, may need the use of more advanced techniques. The problem to be solved can be formulated as follows. Given sensor readings $S_{0..n} = \{S_0, S_1, ..., S_n\}$, calculate the transformation matrix $T_{k-1,k} \in \mathbb{R}^{4x4}$ describing the rotation and translation necessary to displace the ego-vehicle from pose P_{k-1} to pose P_k. Let us now review

some of the existing solutions in the literature, which differ for the deployed sensors:

- Camera: the use of vision sensor for the ego-motion estimation has been widely explored. Single or stereo cameras can be adopted. In both cases they can be referred to as *visual odometry* (VO) approaches. Most effective solutions from the earliest [Longuet-Higgins, 1981] are composed of four main steps: feature extraction, feature matching, motion estimation and local optimization. More recent propositions [Wang et al., 2017] focus on developing end-to-end trained deep models, avoiding the standard pipeline and leveraging on deep learning techniques.
- Lidar: with laser scanners the term *point cloud registration* is preferred when the displacement is being estimated on the basis of two sequential point clouds. For this purpose, the traditional approach since [Besl and McKay, 1992] is the Iterative Closest Point (ICP) algorithm. It consists of four main steps: selection (which is similar to feature extraction in VO), matching between selected points, weighting the correspondences between points accordingly with how reliable and relevant to the transformation and finally minimization. At this last step, a specific cost function is minimized optimizing for the best geometric transformation from P_{k-1} to P_k. Current reference for the state of the art in point cloud registration is the Lidar Odometry And Mapping (LOAM) technique [Zhang and Singh, 2018].
- Other sensors such as radars [Schuster et al., 2016] and sonars [Ribas et al., 2008] are less commonly used propriceptive sensors in addressing the position tracking problem, but might be of interest in specific use cases.

B Global localization

If position tracking can be conceived as a continuous estimation problem from a known starting position, when facing global localization the starting configuration is not relevant anymore. Approaches to this problem can be purely based on GNSS-based or extend their prior knowledge with an HD Map. The first category has been extensively studied in [Obradovic et al., 2007], where GNSS pseudo ranges are used as main source and a Kalman filter fuses odometer and gyroscope sensors into a dead-reckoning process. The second category concerns all those methods adopted for exploiting the HD Maps presented in Section 2.3.1. Depending on the accuracy-level of the localization technique, we might be able to answer one or more of the three questions presented in Fig. 2.9:

- **What road?** Most navigation systems, even purely GNSS-based, can provide this level of accuracy. The position of the vehicle is correctly estimated within 5-10 m using an ordinary Global Positioning System (GPS).
- **Which lane?** In order to distinguish in which of the lanes composing the identified road is situated the ego-vehicle, an accuracy of about 1 m is required.

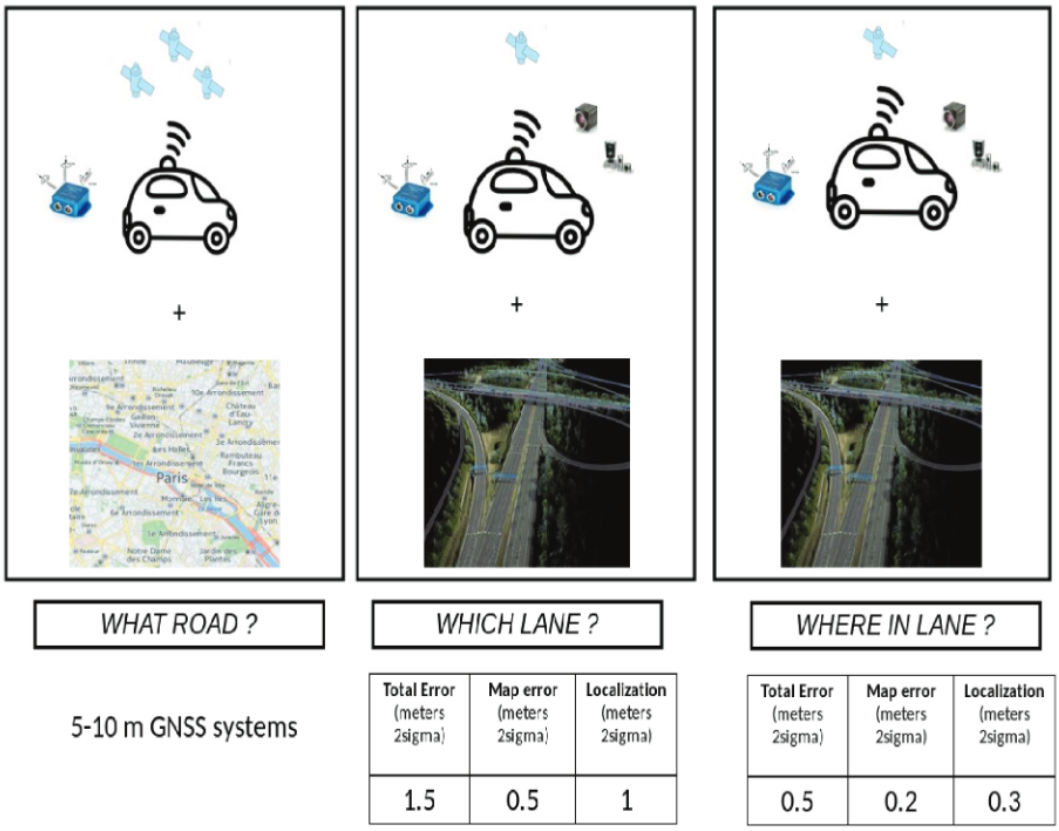


Figure 2.9: What road? Which lane? Where? [Consortium et al., 2004]. This macro characterization of localization systems assists in the identification of technical requirements for specific applications. In the illustration, more accurate localization entails more sophisticated equippement.

- **Where in lane?** A precise position of the vehicle within its lane, within 30 cm of its actual localization. This accuracy is attained with more sophisticated systems such as differential GPS and Real Time Kinematic (RTK).

This differentiation in three levels of accuracy may be beneficial in the description of the requirements for the realization of a give ADAS. For example, in order to offer map-based ACC which adapts the speed of the vehicle taking into account the speed restrictions of the road, a "What road" localization might be sufficient. In case of distinct limitations for distinct lanes, "What lane" accuracy might be required but no further precision would be needed.

Recent methods based on HD Maps and exteroceptive sensors have proven to be capable of centimeter-level localization. In the case of feature-based localization models (Section 2.3.1.C), these techniques generally adopt a pipeline composed of three steps:

1. Feature extraction from sensor data.
2. Matching with map features.

3. Position error minimization.

Varying the type of extracted feature and the deployed sensors, several solution have been studied. [Li et al., 2018] uses smart camera detections of lane markings, achieving lane-level localization while improving safety with integrity monitoring. [Ghallabi et al., 2018] exploits reflectivity of lane markings detected in highway using laser scanners. Another lidar application of is proposed by [Welte et al., 2020], which method improves data association in the matching step utilizing specific buffer adjustment.

2.3.3 Simultaneous Localization And Mapping (SLAM)

We presented so the Mapping task and the estimation of the vehicle Localization as two distinct problems. As a matter of fact, these two challenges are intrinsically connected and ordinarily addressed as one. In the case we are not *mapping with known pose*, than the challenge named Simultaneous Localization And Mapping (SLAM) arises. Traditional [Smith and Cheeseman, 1986] challenge in robotic sector, the aim is to build a consistent map of the environment while *simultaneously* estimating the position of the robot within it. In their two part tutorial, [Durrant-Whyte and Bailey, 2006] and [Bailey and Durrant-Whyte, 2006] illustrate the EKF-based SLAM where the Gaussian distribution of current robot and landmark positions are recursively estimated.

2.4 Objective of this thesis

This research aims at developing information fusion solutions in the scope of lane boundaries detection. The above has served as introduction to the context we will be working in and where the data fusion will happen. As the contextual system architecture has been presented together with the functioning of the inputs, which might be issued from both smart sensors and map-providers, it is now introduced the specific case study we want to address.

2.4.1 L3 automated driving project

The aim of this project is to realize a working prototype of L3 automated vehicle. According to the definition proposed by SAE [SAE International, 2021], this is the first level of automation where actual automated driving features are available. Specifically, the driver is not assisted anymore: when *certain conditions* are met, automated driving features can be engaged letting the driving system "take control". The driver is now to all effects a passenger, even though they are still in the driver's seat. In fact, they may initiate leisure activities during this phase, such as reading or enjoying entertainment features proposed by the vehicle itself, but without being allowed to complete isolation. They have to be able, at any time the vehicle may require it, to take back the vehicle control within a *limited time-frame*. The automated system may, in fact, have verified that the required conditions for automated driving are not met anymore and the passenger shall be ready to "become driver" again.

It is worth specifying what kind of conditions are being referred to and what is the order of magnitude of the *limited time-frame*. These specifics, in fact, make all the difference. For example, the desired conditions may be so hard to be met (e.g. on-board sensors have to detect lane markings at 500 m distance at least) making the "self-driving mode" virtually impossible to be activated. Yet again, a 20 ms time-frame would make the car-to-driver hand-off impossible to be attained for the passenger. For these reasons, the correct balance between usability and safety have to be identified in designing the system. In this project, the hand-off time-frame has been sized in the 10-15 s. This time interval is both sufficient for the passenger to switch from its current activity back to the driving and task and for the vehicle to plan the eventual Minimum Risk Maneuver (MRM). The case of an user not being able to take back control of the vehicle shall be foreseen and handled. The automated system demanding and not obtaining an hand-off should engage in the MRM. The term MRM identifies an action to be performed by the automated vehicle which minimizes the risk of occurring in hazardous situation, even beyond the required conditions for self-driving. For instance, in highway scenario, the MRM would be the minimal trajectory for reaching the first available road-shoulder [Zhang et al., 2021]. The clarification of the car-to-driver handoff is a delicate subject as it defines where the responsibility in the vehicle behaviour passes from the car (manufacturer) to the driver. This distinction would be crucial in the event of an accident and the implication of insurance and law regulations.

The required conditions for the activation of the automated driving features, in this specific project, have been defined to be the followings:

- The vehicle has to be located in the Autonomous Driving Operational Design Domain (ODD). These areas are generally sections of motorways which have been qualified to be eligible for this purpose. Additionally, an HD Map of these roads is generally issued by the installed map-provider.
- Reliable localization of the vehicle has to be available, in order to validate that it is within the ODD and the use of HD-Map-based navigation is viable.
- A minimal sensor (sub-)set for safe navigation has to operational and no flag of malfunctioning has to be reported.
- Minimal quality of lane markings (or generic lane boundaries) is respected to guarantee comfortable driving to the passenger.
- Estimation of the lane repartition is available at sufficient range to perform predictive control of the vehicle and/or complex maneuvers (lane change or other vehicles takeover).

In the case of compliance with all of these requirements, the driver is allowed (and prompted from the HMI) to enter the "automated driving mode". After activation, the following functionalities are available to the passenger:

- They can indicate at any moment the desired speed of the vehicle (conditioned to speed regulation) and it will be maintained as possible.

- Turn directions can be used to apply for a lane change which is actuated as it is possible.
- An additional slider can be used to regulate the minimal inter-distance to be maintained from the preceding vehicle.

These complex functionalities are being implemented while it is being guaranteed safety inter-distance from other vehicles, safe reactions to cut-in or unexpected braking of other road users. The realization of the system is deployed according to a specific traditional autonomy stack (Section 2.1.2) of modules connected in pipeline. In order to comply with the automated driving requirements, each distinct module has therefore specific objective to maintain.

Lane boundaries estimation objectives Let us detail the specific objectives in the area addressed by this study, the estimation of the surrounding lane boundaries.

First of all, we are introducing some fundamental terminology. Hereinafter, the term "lane boundary" equivalently identifies lane markings, barriers and any other road element relevant to the partition of the roadway into lanes. This environment modeling approach aims at identifying the edges of these fundamental travel corridors where vehicles can drive safely and efficiently. A reliable and accurate characterization of the roadway and its lanes would enormously benefit a lane-based navigation. Positively supported tasks would be lane assignment, path prediction, lateral and longitudinal control of the vehicle. It is proposed in this work to exploit multiple sources of smart sensor and map-provided data in order to support these tasks which come with specific requirements on the lane representation:

- **Lane assignment** is the task of correctly understanding in which of the surrounding lane the other vehicles are driving. It is required an estimation of the position of the lane boundaries, at medium-long range.
- **Path prediction** is the task of forecasting the future trajectory of the other vehicles. Kinematics models of the road agents are central in this matter, yet a knowledge of the lane repartition can offer decisive factors in this prediction. It is required an estimation of the position of the lane boundaries, at medium-long range.
- **Lateral control** is the task of actuating on the lateral position of the ego-vehicle in order to follow the planned trajectory. It is required an estimation of the position, orientation and curvature of the ego-lane boundaries, at short range.
- **Longitudinal control** is the task of actuating on the longitudinal position of the ego-vehicle in order to follow the planned trajectory. It is required an estimation of the position, orientation and curvature of the ego-lane boundaries, at medium range.

Two general principles in the lane boundaries estimation which are being followed are simple to be expressed but are generally reflect in not trivial challenges:

1. **Be robust**: when lane boundaries are being detected, it is vital to detect the all and only existing lane boundaries in the environment. More specifically, we want to avoid missed-detections and false-detections.

2. **Be precise**: when estimating the state of a detected lane boundary, it is important to be as precise as possible in its state estimation. In example of curvature estimation, we want to as true as possible

2.4.2 L3 prototype vehicle and sensor set

The deployed vehicle for our experiments (an ad-hoc configured Renault Espace) is shown in Fig. 2.10 from two different viewpoints. Additionally, a detail of the front bumper shows how the sensors have been integrated. The vehicle is equipped for the perception of its surroundings with:

- Smart FrontCam, 30Hz, FoV: $53° \times 120\ m$
- Smart AVM (4 cameras), 20Hz, FoV: $360° \times 20\ m$

These sensors implement device-specific data processing algorithms. Specifically, the Smart FrontCam can send out up to 4 measurements per delivery. The Smart AVM has similar capabilities, however we narrow down its usage to the ego-lane estimation (up to 2 measurements). This limitation is due to the difference in maturity of the two sensors.

Another specific unit handles the localization of the vehicle. It implements particle-based localization algorithms applied within a remotely provided HD-map (streamed via Ethernet according to ADASISv3 [Ress et al., 2008] standards), using a commercial positioning system.

2.4.3 Custom dataset of recorded sensor data

Sensor output and map data has been recorded for testing, driving the prototype vehicle on the French A86 (Créteil-Versailles). This highway presents the target use case where the automated driving features are going to be available. In fact, the map-provider setup installed on vehicle can deliver an HD Map of these roadway, as it is part of "autonomous driving suitable" road. The on-board main computer can execute specific ADAS functionatilies running a precompiled library or record sensor data useful for offline testing of fusion algorithms. In "recording mode" the following data stream are being recorded:

- Perception sensors measurements (which include the above mentioned smart sensors and others)
- All map-provider deliveries
- Proprioceptive sensors measurements
- Context (3) cameras video stream

Figure 2.10: L3 prototype vehicle. On the left side of the illustration, the vehicle from two different viewpoints. On the right side, a detail of the front bumper featuring a front lidar and one of the sensors of the Smart AVM.

A proprietary data format has been defined to coherently timestamp each data package according to a common clock. Recorded data is stocked in "capsules" of maximum duration 5 minutes. Each capsule is associated with a metadata file, adding useful information on the driving conditions. A complementary kind of information is given by the above-mentioned observations, timestamped and classified in four classes:

- Weather: sunny, rainy, cloudy, snowy, foggy
- Road type: highway, urban, country road
- Traffic: light, heavy
- Event: roadwork, tunnel, bridge, toll, roundabout

Additional comments of meaningful events can be expressed by the copilot in free form. Some examples may be "hay truck", "wrong blinker lane change" or "red light + pedestrian".

2.5 Concluding remarks

Presenting the specific use case and challenges to be addressed in this research entails the definition of two distinct situations. Despite the advanced developments in solutions for precise localization of the vehicle, it is likely to occur in unexploitable HD maps or unavailable map-providers. Based on this distinction, the rest of the manuscript is divided as follows:

- Multi-sensor fusion for lane boundaries estimation
- Map-aided multi-sensor fusion for lane boundaries estimation

In both cases, the proposed solutions are presented in details for clarifying the methodology and the implementation. Quantitative results obtained from the experimental setup and recorded dataset are being discussed and eventually confirmed by on-vehicle trials in closed loop configuration.

Chapter 3

Multi-sensor fusion for lane boundaries estimation

This chapter addresses the problem of lane boundaries estimation using measurements issued of smart sensors. In order to benefit the multi-sensor fusion, which generally result in improved reliability and accuracy provided by diversity in perception technology, a model for smart sensor lane boundaries detections is firstly presented. In what follows, the proposed tracking solution is detailed, taking into account the use case objectives previously stated. A different lane boundaries model is adopted in the tracking procedure, which allows for better flexibility and manageability. Ultimately, a novel clothoid spline representation is issued as a result to the subsequent modules. This methodology is studied in three different experimental setups where it is at first implemented, then quantitatively evaluated and lastly on-road tested on the L3 prototype vehicle.

3.1 Introduction

In the challenge of lane boundaries estimation, we defined our goal as delivering the best representation of the surrounding lane repartition when using observations from smart sensors. These sensors are generally characterized by distinct field of view, technology, feature extraction algorithm and measurement frequency. This last quantity is extremely important as it affects the availability rate of updated information of the road scene. In practice, commercialized smart sensors generally have a sufficient frequency for the target application we described. These rates are however specific to the sensor and, in general, different from each other. We are working with asynchronous sensors, but the proposed solution aims at delivering an updated lane boundaries estimation with a constant and reliable rate, therefore the input measurements will be processed periodically after being stored in specific buffer. The workflow of our method is shown in Fig. 3.1. On the left, the environment is outlined highlighting the field of view of the two sensors available in the experimental platform. It is also depicted, on the ego-vehicle, the frame of reference used for presenting any information in the scene: the body frame. On the right, the blocks describe the

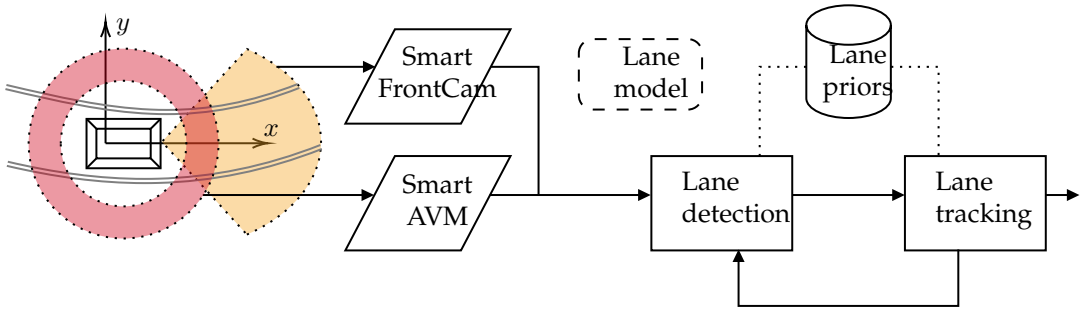


Figure 3.1: Lane boundaries detection and tracking pipeline.

main steps of the proposed solution, presented in the following of this chapter.

3.2 Smart sensor model

We present at first the model adopted for presenting input data to our lane boundaries tracking solution. The sensor set of the target architecture is supposedly composed of smart sensors, capable of extracting meaningful features for lane boundaries detection. Independently of the sensor nature, their measurements are detection of lane boundaries in the driving scene which described in each delivery of smart sensor data. In the following, the content of a smart sensor delivery is illustrated. The meaning of each field is described, focusing on the probabilistic representation of lane boundaries.

3.2.1 Smart sensor delivery

Let us introduce the adopted notation. Despite the involved smart sensors are generally asynchronous, the time repartition is considered discrete as the measurements are being processed periodically with regular and constant frequency. The smart sensor delivery processed at instant t issued of a generic sensor $\mathbf{Sens}$ is detoned with the set $\mathbf{z_t^{Sens}}$. Measures (or single detections of lane bounding elements) $M_i \in \mathbf{z_t^{Sens}}$ at each delivery are represented in the body frame (frame F_M, with origin located at the middle of the rear axis of the car) and modeled with polynomial curves, as follows:

$$M_i = [c_0, c_1, c_2, c_3, x_{min}, x_{max}, \mathbf{\Sigma}_P, M_{type}] \in \mathbf{z_t^{Sens}} \tag{3.1}$$

The individual components of each M_i respectively represent:

- $c0, ..., c3$ are the coefficients defining the polynomial curve that describes the detected lane boundary
- x_{min} and x_{max} define the abscissa limits where the polynomial is defined
- $\mathbf{\Sigma}_P \in \mathbb{R}^{5\times5}$ is the covariance matrix with respect to the aforementioned components. Its use is detailed in the following.

- $M_{type} \in \{marking, barrier, curb, ...\}$ describes the type of the detected road element

It is noted the continuous description of the measurement given by:

$$P(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3,\ x \in [x_{min}, x_{max}] \tag{3.2}$$

This description entails a physical meaning for each of the polynomial coefficients: c_0 represents the lateral offset of the polynomial curve at abscissa $x = 0$, c_1 represents the tangent of the polynomial curve at abscissa $x = 0$, c_2 is an indicator of the curvature of the polynomial curve, c_3 is an indicator of the curvature rate of the polynomial curve. This polynomial model for measurements reflects a standard output provided by off-the-shelf devices adopted in the automotive industry.

The measurement uncertainty is given by the covariance matrix $\mathbf{\Sigma}_P$ with respect to each parameter. Each random variable in the measurement M has its correlation expressed with respect to itself and to the other variables. Under the hypothesis of independent and uncorrelated random variables $\sigma^2_{c_i c_i}$ expresses the uncertainty on the measurement of the polynomial coefficient c_i. The function domain $[x_{min}, x_{max}]$ where the polynomial $P(x)$ is defined consists itself of an interval of uncertain bounds. It is also likely that the calibration process (correct positioning of the smart sensor within F_M for the transformation of sensor measurements from sensor frame to body frame) could also come with imperfect accuracy. In an example case of inaccurate calibration in the longitudinal positioning of $\epsilon = 1$, referring to the state of a lane boundary measurement at longitudinal distance x would imply to be mistaking $P(x)$ with $P(x \pm \epsilon)$. For this matter, in this representation, the uncertainty on the longitudinal distance is uniquely expressed with a single random variable x and a single covariance σ^2_{xx}, ideally covering both calibration and interval size uncertainty.

Therefore, supposing independence of the random variables, the covariance matrix would be filled and given in the following form:

$$\mathbf{\Sigma}_P = \begin{bmatrix} \sigma^2_{xx} & & & 0 & 0 \\ & \sigma^2_{c_0 c_0} & & & 0 \\ & & \sigma^2_{c_1 c_1} & & \\ 0 & & & \sigma^2_{c_2 c_2} & \\ 0 & 0 & & & \sigma^2_{c_3 c_3} \end{bmatrix} \tag{3.3}$$

A graphical representation of its meaning is given in Fig. 3.2, where an example of measurement and corresponding "envelope" of uncertainty are shown in orange.

Empty $\mathbf{\Sigma}_P$ in sensor delivery No guarantee is given that the sensor will be able and willing to indicate the uncertainty for each detection. Therefore, empirical models can be adopted for characterizing the behaviour of each sensor in the case of empty covariance matrices. The models feature specific coefficients that can be tuned and adapted for different sensors. This workaround for this deficiency consists of experimentally characterizing the device and to filling out this matrix

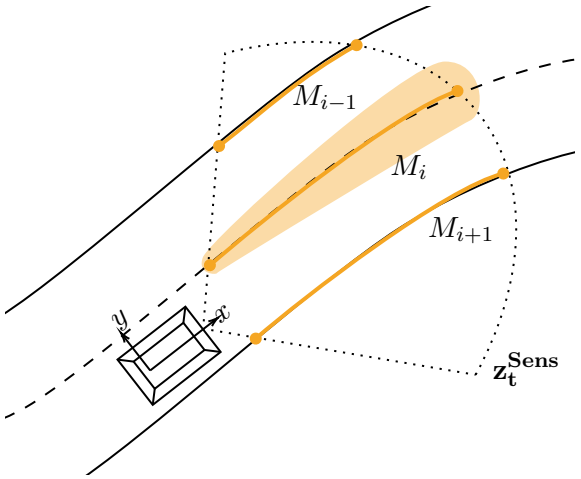


Figure 3.2: Smart sensor model. In orange, three polynomial measurements - detections of lane boundaries in the driving scene. For M_i, the corresponding "envelope" of uncertainty is shown as light orange area.

with constant values (this identification work has been done at Renault for the standard deviation of Smart FrontCam's coefficients c_0 and c_1, results reported in the internal technical report [Michelet-Gignoux, 2019]).

In the following, an example of empirical model is given where the inaccuracy of the measurement of the state variables grows exponentially with its euclidean distance from the sensor. An initial uncertainty, in the form of covariance matrix, is defined as:

$$\mathbf{\Sigma}_P(0) = \begin{bmatrix} \sigma_{xx}^2 & 0 & 0 \\ 0 & \sigma_{yy}^2 & 0 \\ 0 & 0 & \sigma_{\theta\theta}^2 \end{bmatrix} \tag{3.4}$$

The covariance matrix $\mathbf{\Sigma}_P(0)$ contains initial parameters of the model to be characterized with respect to the sensor performance in measuring position (state variables x and y) and orientation (variable Θ) of the lane boundary. For a generic measurement sample $(x, P(x))$, at euclidean distance $d = d(x, P(x)) = \sqrt{x^2 + P(x)^2}$, the measurement noise scales an the initial $\mathbf{\Sigma}_P(0)$ by a factor that grows exponentially with the euclidean distance. The measurement noise evolution is described as follows:

$$\mathbf{\Sigma}_P(d) = exp(\alpha_M d)\mathbf{\Sigma}_P(0) \tag{3.5}$$

where the characterization of the parameter $\alpha_M d$ is essential for the representativeness of the model. Its correct tuning is also essential for keeping reasonable value of uncertainty within the range and field of view of the sensor. Values of the parameter $\alpha_M d$ can be estimated fitting the exponential models to the empirical error measured from measurement data obtained from the sensor and compared to a reference ground truth.

The choice of this model favours the rapid growth of uncertainty for any state variable regardless, in both lateral and longitudinal distance. In the effort of using a same generic noise model for any generic smart sensor, irrespective of the

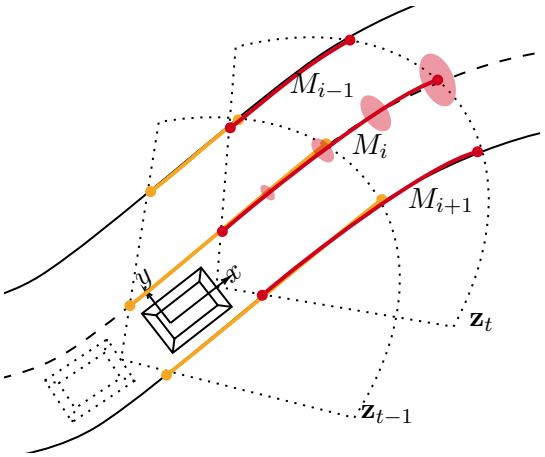


Figure 3.3: Three lane boundaries are being detected at instant $t-1$ ($\mathbf{z_{t-1}}$ in orange) and at instant t ($\mathbf{z_t}$ in red).

specific sensor technology, the model risks of being unrelated from the mathematical descriptions of the specific phenomena that are causes of the measurement noise. Indeed, for vision sensors such as cameras and AVMs, a model with linear or quadratic scaling noise would be better physically appropriate - where the exponential model lacks.

3.3 Proposed solution: Feature-tracking

This section describes the core of the proposed fusion architecture. Our goal is to estimate the structure of the surrounding lane boundaries exploiting sensor deliveries issued of different sensors and at different time instant (Fig. 3.3). Hereinafter, we introduce two distinct models for representing the lane boundaries. The first is more flexible and close to the actual design of roads and highways with respect to the smart sensor model of Section 3.2. The second one is basic but ideal in the tracking procedure.

Lane boundaries model A semi-parametric model (Fig. 3.5) is preferred to assure a more general representation for the tracked lane boundaries, let us introduce it. In the followings, the notation adopted to indicate the collection of the tracked lane boundaries at instant t uses the term $\hat{\mathbf{x}}_\mathbf{t}$, where the *hat* denotes the fact that this entity represents an estimation of the surrounding road elements. The chosen model for each individual track (or individual lane boundary) $T_i \in \hat{\mathbf{x}}_\mathbf{t}$ (track collection at instant t) is the clothoid spline. Each track is a curve composed of a variable but finite number of clothoid segments individually defined as follows:

$$S_j = [x_0, y_0, \psi_0, \kappa_0, \kappa_1, l, \mathbf{\Sigma}_S] \in T_i \tag{3.6}$$

Each component of S_j has a physical meaning in the description of the curve: x_0 and y_0 are the coordinates of the point where the clothoid segment originates, ψ_0

and κ_0 are respectively the initial orientation and curvature of the clothoid segment and κ_1 is the curvature rate. It is noted that the interest of using this specific curve lies in its definition where its curvature linearly grows with its curvilinear abscissa starting from κ_0 at a κ_1. The continuous description of each segment is given at curvilinear abscissa s by the Fresnel integrals [Marinelli et al., 2017]:

$$x(s) = x_0 + \int_0^s cos\left(\frac{1}{2}\kappa_1\tau^2 + \kappa_0\tau + \psi_0\right) d\tau,\ s \in [0, l] \tag{3.7}$$

$$y(s) = y_0 + \int_0^s sin\left(\frac{1}{2}\kappa_1\tau^2 + \kappa_0\tau + \psi_0\right) d\tau,\ s \in [0, l] \tag{3.8}$$

The uncertainty on this representation is given by the covariance matrix $\mathbf{\Sigma}_S$ with respect to each parameter of each segment. This transcendental functions require approximation methods to be handled but they accurately reflect the techniques used for road infrastructure design [Marinelli et al., 2017]. This representation is also immediately suitable for curvature-based control of the vehicle.

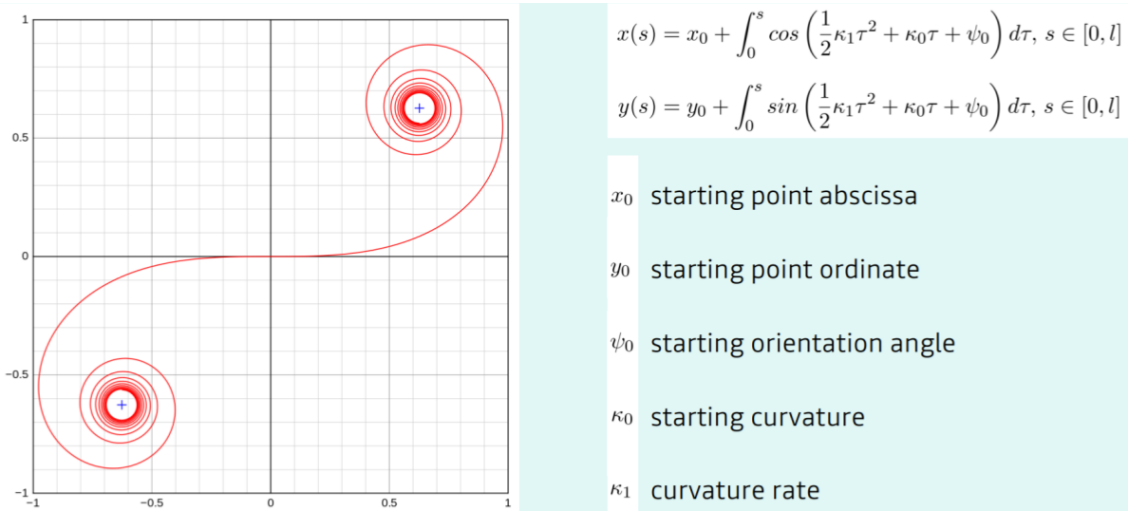


Figure 3.4: Illustration and description of a clothoid curve.

Road features model Estimating the road boundaries under the clothoid spline representation has several advantages, as we illustrated. However, the formulation and parametric description of this curve brings additional complexity and numerical difficulty in managing these mathematical objects. The basic idea is that a set of ordered control points jointly with an appropriately chosen interpolation method can completely describe a geometric curve. In our framework, these control points are tracked and referred to as features. A curve C_i consists of a finite number of features F_j defined as:

$$F_j = [x_j, y_j, \theta_j, \mathbf{\Sigma}_F] \in C_i \tag{3.9}$$

This representation describes a curve specifying abscissa x_j, ordinate y_j and heading θ_j of its control points. Considering the application domain and the

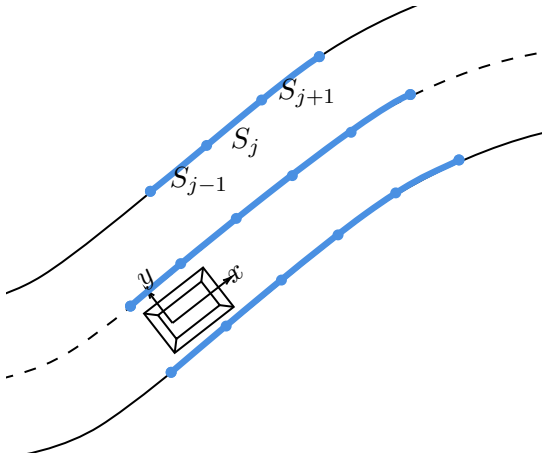


Figure 3.5: Lane boundary model. Distinct clothoid segments allow for a more flexible description of the individual road element.

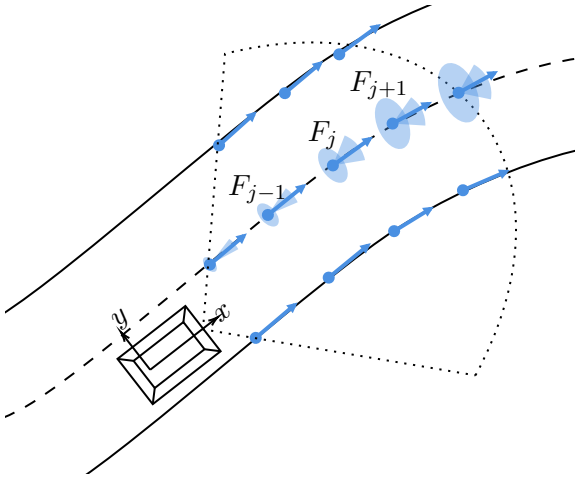


Figure 3.6: Road feature model. Features F_j are sampled along the measurement at constant inter-distance. In light blue, ellipses and cones are respectively used to graphically display position and orientation uncertainty.

uncertain nature of the described entity, these values constitute a random vector and $\mathbf{\Sigma}_F$ is its covariance matrix.

In Fig. 3.6, features F_j are sampled along the measurement at constant inter-distance. Ellipses and cones are respectively used to graphically display position and orientation uncertainty.

Method overview In order to track and cumulatively refine the estimation of a feature set, the adopted procedure consists of the following steps:

- Initialization
- Prediction
- Association

- Update

For each detected and initialized feature, a filtering process is carried on using a traditional Kalman filter. This method is selected for its low computational cost and optimality under Gaussian noise assumption, which is supposed to be our case.

3.3.1 Initialization

Whether the last measurements are not being associated to any of the existent tracks or the tracking procedure is just being started, the initialization of new track is addressed. First of all, the measurement might be subject to specific checks in order to avoid unwanted observations beforehand. For example, in case of a known field of view of the sensor, it might be wise to avoid any detection coming from outside of this surface. Additionally, some smart sensor are capable of assigning a confidence value to each of the measurements. Whether this value is expressed as a probability of existence or as discrete score of confidence, it is possible to impose a minimum threshold in order to include in the tracking only observations considered to be of "good quality". This preliminary selection may improve the precision of the final outcome, but the overall undetected lane boundaries (false negatives) count might raise too. The procedure for track initialization mainly aims at converting the information in the measurement M into the road feature model, which is discrete and more manageable in the tracking process. Two policies are being proposed to extract the most out of each observation, which defines at what point the measure is being sampled and a new road feature is being initialized:

- **Constant abscissa inter-distance sampling** For a given longitudinal distance parameter Δx, a new feature F will be initialized for each distinct $x_j \in [x_{min}, x_{max}]$ where:
$$x_j = x_{min} + j\Delta x \tag{3.10}$$

- **Constant length inter-distance sampling** For a given length parameter Δl, a new feature F will be initialized for each distinct $s_j \in [0, l]$ where l is the total length of the measure M:
$$s_j = j\Delta l \tag{3.11}$$
This sampling method is more adapt for observation with significant curvature rate. The explicit expression of a polynomial curve length can be used, however, if efficient methods for computation of clothoid abscissa are implemented, it can be considered a conversion of the polynomial measurement into clothoid segment. $x(s_j)$ would be then obtained from the Fresnel integrals of Equation 3.7.

Whatever the policy, these x_j samples are used to define the initial conditions of each state variable. For a chosen $x_j \in [x_{min}, x_{max}]$:

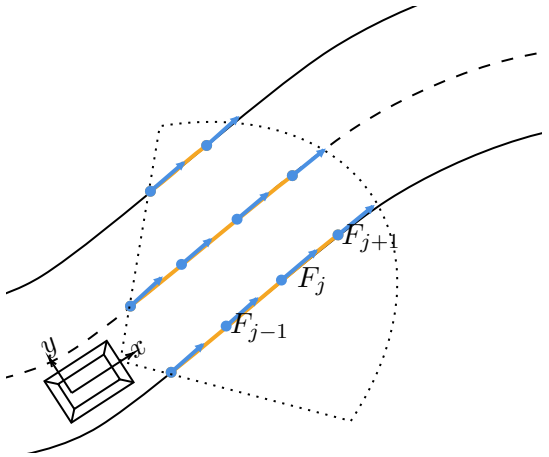


Figure 3.7: Feature set initialization. Measurement in orange and sampled features in blue.

$$F_j = \begin{bmatrix} x_j \\ y_j \\ \theta_j \end{bmatrix} = \begin{bmatrix} x_j \\ P(x_j) \\ \arctan(P'(x_j)) \end{bmatrix} = \begin{bmatrix} x_j \\ c_0 + c_1 x_j + c_2 {x_j}^2 + c_3 {x_j}^3 \\ \arctan(c_1 + 2c_2 x_j + 3c_3 {x_j}^2) \end{bmatrix} \tag{3.12}$$

$\mathbf{\Sigma}_F$ is set in accordance with the measurement model in Section 3.2 also taking into account which sensor is providing the measurement.

In Fig. 3.7, a graphical representation of the initialization procedure is given. From measurements M, polynomial curve in light orange in the field of view of the sensor, features F_j are sampled initializing the filtering. Other useful information for the tracking, such as date of initialization and M_{type} are stored in the lane boundary track.

We propagate the uncertainty of the polynomial coefficients $\mathbf{\Sigma}_P$ with a first order approximation yielding:

$$\mathbf{\Sigma}_F = Var(\begin{bmatrix} x_j \\ y_j \\ \theta_j \end{bmatrix}) = \begin{bmatrix} \frac{\partial F_j}{\partial M_i} \end{bmatrix} \mathbf{\Sigma}_P \begin{bmatrix} \frac{\partial F_j}{\partial M_i} \end{bmatrix}^T \in \mathbb{R}^{3\times 3} \tag{3.13}$$

3.3.2 Prediction

The goal of the prediction phase is to perform temporal alignment of previously tracked elements to the latest sensor delivery to be processed. The fusion procedure is in fact performed periodically (40 ms period) while each smart sensor has a distinct frequency in addiction to not being synchronized in any manner with the other observers of the road. Another benefit of using the road feature model comes out in this transformation. At this step, the appropriate rotation and translation is applied to each curve in such a way that tracks are temporally and spatially coherent with the freshest sensor delivery. If rather than using simple road feature we would have wanted to perform this transformation on a polynomial curve, the result would have been hardly manageable (the rotation

of a polynomial is generally not a polynomial anymore). Still, where this would usually be handled with an ordinary geometric transformation, in this context it consists of a prediction. In fact, the actual parameters required for this operation are as well outcome of an estimation process. In the case of state prediction, an evolution model is generally introduced and its definition strongly impacts the final outcome. Within this context, however, the evolution model of our targets is perfectly known and trivial: control points of lane boundaries do not (in our scope) evolve in time, exclusively the reference frame is moving. This movement is separately estimated and defined at each iteration as:

$$\Delta Ego_t = [dx, dy, d\theta, \mathbf{\Sigma}_E] \tag{3.14}$$

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

$$\hat{\mathbf{x}}_{\mathbf{t-1}} = {}^{t}T_{t-1} \oplus {}^{t-1}\hat{\mathbf{x}}_{\mathbf{t-1}} \tag{3.15}$$

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

$${}^{t}T_{t-1} = \begin{bmatrix} cos(d\theta) & -sin(d\theta) & dx \\ sin(d\theta) & cos(d\theta) & dy \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

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

$$\mathbf{x_t} = \mathbf{x_{t-1}} + \mathbf{w_t} \tag{3.17}$$

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

$$\mathbf{w_t} \sim \mathcal{N}(0, \mathbf{\Sigma}_E) \tag{3.18}$$

3.3.3 Association

The association between measurements and tracks is crucial to successively integrate fresh upcoming information. In order to do this, an appropriate metric to express the distance between a feature set C_i and a measurement M has been defined. It is supposed that, between $t-1$ and t, the ego-motion is sufficiently small to allow at least the orthogonal projection of one feature on the measure, if they represent the same lane boundary. Under this assumption, each feature is projected on each measure as illustrated in Fig. 3.8. Each successful projection of $F_j \in C_i$ appoints to a correspondent feature $p_\perp(F_j)$ sampled along the measure M at $(x_\perp, y_\perp)$. The Mahalanobis distance between the two can be computed as:

$$d(p_\perp(F_j), F_j) = \sqrt{(p_\perp(F_j) - F_j)^T (\mathbf{\Sigma}_M(x_\perp, y_\perp) + \mathbf{\Sigma}_F)^{-1} (p_\perp(F_j) - F_j)} \tag{3.19}$$

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

$$d(M, C_i) = \max_{F_j \in C_i} d(p_\perp(F_j), F_j) \tag{3.20}$$

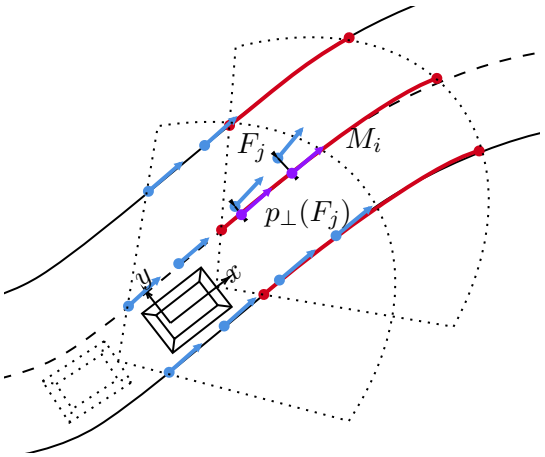


Figure 3.8: Tracked road features in blue are being projected for association onto the latest measurements in red.

A distance matrix built on these values allows the use of a Global Neighrest Neighbourd (GNN) algorithm to finalize the association (implementation details included in Appendix B).

3.3.4 Update

Once the association between the measure M and the feature set C_i is confirmed, the aim is to update the state of existing features and to possibly extend the track with newly discovered elements. Existing features are updated in the filter update step with their projection $p_\perp(F_j)$ on the associated measurement. State variables and covariance update of F_j is done according to the equations of Kalman filter state update, using the corresponding measurement feature $p_\perp(F_j)$ and obtaining the latest estimate of the road feature $\hat{F}_j$:

$$\hat{F}_j = F_j + K(F_j - p_\perp(F_j)) \tag{3.21}$$

$$\hat{\mathbf{\Sigma}_F} = (I - K)\mathbf{\Sigma}_F \tag{3.22}$$

It is recalled that measurement noise $(\mathbf{\Sigma}_M(x_\perp, y_\perp))$ is involved in the computation of the Kalman gain K.

Where available, the remaining length of the measurement is exploited to initialize, as in Section 3.3.1, newly discovered features and extend C_i. Fig. 3.9 shows unchanged, updated and new features respectively displayed in grey, green and blue.

3.3.5 Delivered output

Preparing the tracked lane boundaries for next module is necessary for exploitability of the results of the procedure. As said, the road feature model is adopted to ease the tracking but it has to be "reconstructed" into a continuous representation. The above mentioned spline representation is attained at each iteration interpolating the curve between two consecutive features by means of the

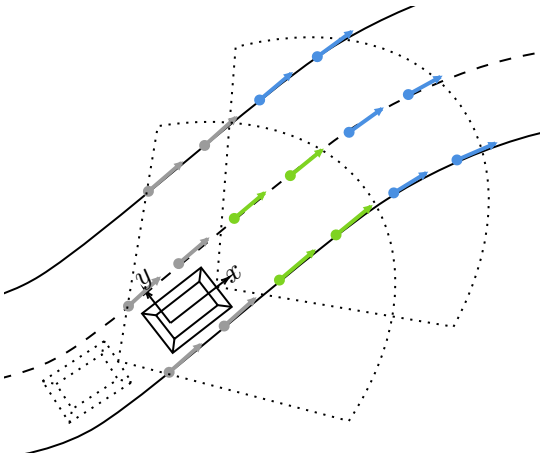


Figure 3.9: Feature set update. Unchanged, updated and new features respectively displayed in grey, green and blue.

algorithm proposed in [Bertolazzi and Frego, 2015]. This procedure builds the clothoid spline by determining the appropriate initial curvature κ_0, curvature rate κ_1 and clothoid segment length l for each given couple of F_j and F_{j+1}. This method guarantees a G^1 degree of continuity of the spline, which is a mandatory requirement for an exploitable result. Therefore, for two consecutive clothoid segments S_j and S_{j+1}, it is verified that $x_j(l_j) = x_{j+1}(0)$, $y_i(l_j) = y_{j+1}(0)$ and $\psi_j(l_j) = \psi_{j+1}(0)$. This interpolation algorithm is efficient and ideal for the family of regular clothoids encountered in highway navigation. Attaining G^2 (curvature) continuous curves would require more computationally expensive algorithms (such as [Bertolazzi and Frego, 2018] which would imply resolving an optimization problem for each couple of features) exceeding our working hypotheses of low computational requirements.

An additional prediction (similar to Section 3.3.2) is being applied to the final result, in order to compensate the temporal delay between the fusion computation and the actual application of the control on the vehicle. The actual delivered output is a representation of the surrounding lane boundaries at the specific instant in time when the control module will be computing the action to be performed by the actuators.

3.4 Experimental results

The aim of the following experiments is to test the effectiveness of the proposed solution, using smart sensors as input. The feasibility of the on-board fusion also has been verified, targeting highway scenarios in fluid traffic at medium-high speed (up to 60 km/h). The specificity of our research focus did not allow for an evaluation on public well-known datasets. For this reason, we collected custom-tailored data to be replayed in the ad-hoc ADAS development framework described hereinafter. This is one of the three different experimental setup where our solution is being validated. The three following experiments describe how the proposed method is being qualitatively tested using recorded data from

the L3 prototype vehicle, quantitatively validated with respect to cartographic ground truth and ultimately put to the "on-road" test.

3.4.1 Development setup

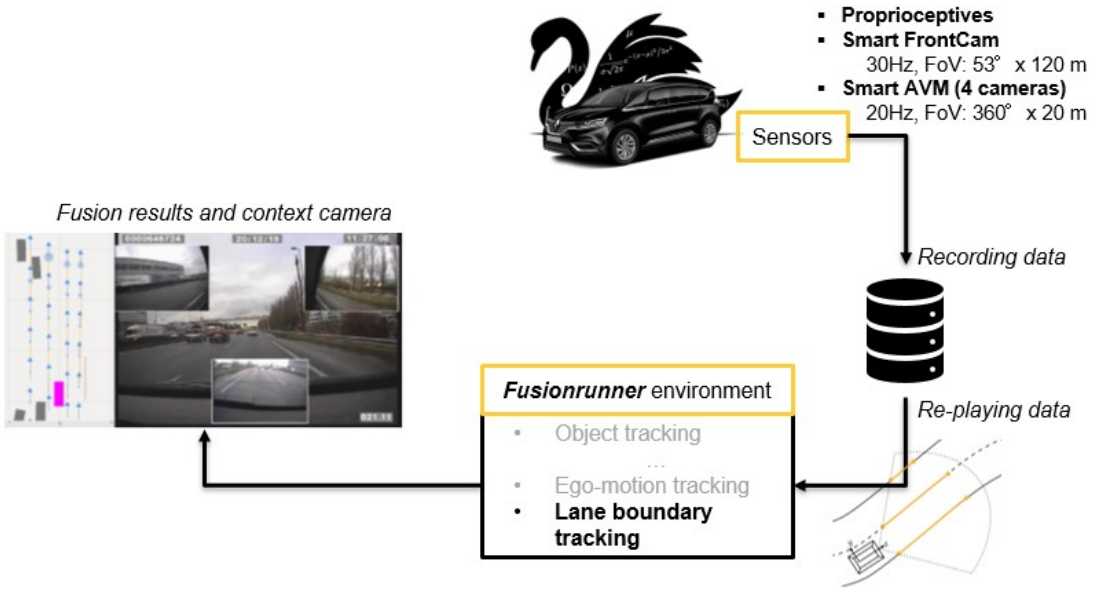


Figure 3.10: Development setup. From the driving of the prototype vehicle (top) sensor data is recorded and stocked (right). Data replaying enables the testing of information fusion solutions for ADAS/AD, implemented in the proprietary framework *Fusionrunner* (bottom). The results are displayed (left) in bird-eye view and validated with with the help of context cameras.

Fig. 3.10 illustrates the "offline" development setup, where the solution is implemented, executed on recorded data and the output is displayed on screen.

As previously mentioned, the sensor set is composed of two different smart sensors, Smart FrontCam and Smart AVM (Around View Monitoring) with different fields of view. The Renault L3 prototype vehicle is also equipped with proprioceptive sensors to estimate its displacement. All of these data has been recorded and replayed in a proprietary environment for ADAS development that guarantees correct timestamp management and allows for testing of tracking solutions.

In the development interface (in Fig. 3.11), it is presented on the left a bird eye view of the environment centered on the ego-vehicle which is displayed in pink. On the right the video stream from the context cameras (which are different from the deployed smart sensors) is shown.

A prior qualitative analysis of the fusion result has been done through the development platform output interface and with the aid of context cameras, confirming the soundness of the result on highway scenarios. In fact, the deployed association criteria can robustly discriminate measurements and tracks that refer to different lane boundaries (covering also close together road edges and mark-

Figure 3.11: On the left, tracking results (blue) and measurements (orange for FrontCam, brown for AVM) are displayed in a bird-eye view. On the right, the context cameras.

ings). It has also been reasonably observed that, processing the two data sources, a fresher estimation is available at an higher rate. This corresponds in average to the sum of the sensors frequency, namely 50 Hz in our experiment.

3.4.2 Evaluation setup

A further quantitative evaluation of the solution has been carried on and in Fig. 3.12 we show the evaluation setup. Same as before recorded sensor data is used to track lane boundaries which are now compared to a lane level ground truth (extracted from an HD map). In fact, the prototype vehicle is also equipped with an RTK localization system which is able to precisely positionate itself within an HD map that contains a trustworthy representation of the surrounding lane boundaries. This comparison enables the computation of performance indicators such as the lateral error of the estimation at a given range distance.

We considered the Smart FrontCam alone as our baseline. Over several kilometers of recorded data, the distribution of lateral error with respect to the ground truth at 0m and at 10m have been studied (where the intersection of FoVs occurs). We verified that both the baseline and the Fusion result contains a bias w.r.t. the HD map. However, according to the variance of the distribution we can affirm that the fusion result is smoother than the Smart FrontCam alone. This can positively affect the lateral control of the vehicle.

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

$$\mathbf{e_t^{FC}} = \mathbf{x_t} - \mathbf{z_t^{FC}} \tag{3.23}$$

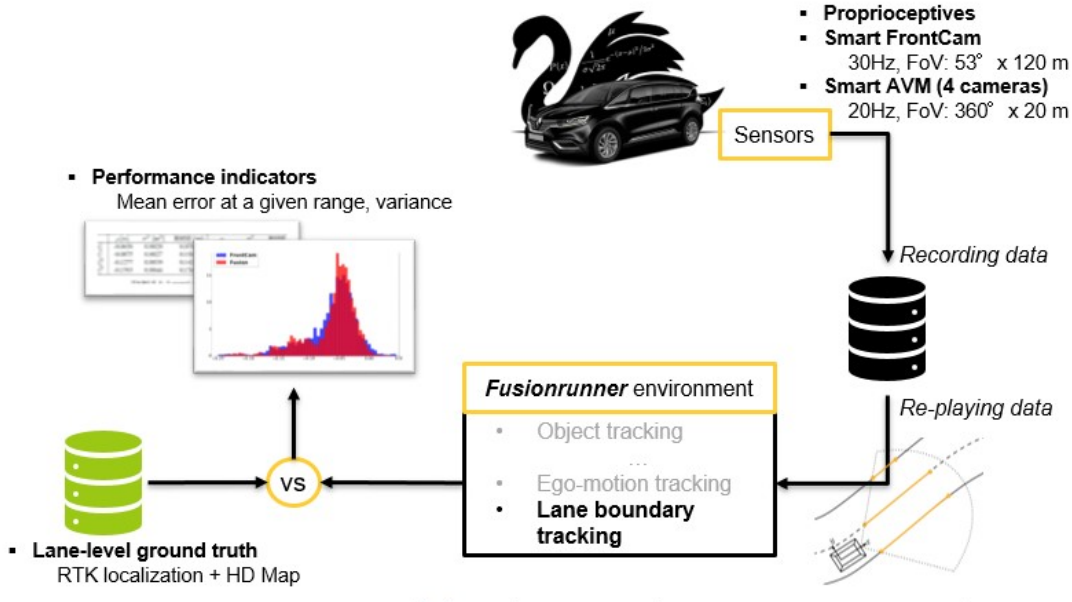


Figure 3.12: Evaluation setup. Tracking results are now being compared to the lane-level ground truth. This allows quantitative benchmarking and algorithm validation.

$$\mathbf{e_t^{Fusion}} = \mathbf{x_t} - \hat{\mathbf{x}}_\mathbf{t} \tag{3.24}$$

where appropriate format transformations are applied to perform these comparisons. In the scope of this work, the analysis focus has been limited to the following error indicators:

$$\mathbf{e_t} = [e_0^L, e_1^L, e_0^R, e_1^R] \tag{3.25}$$

where, for left L and right R markings, a separated lateral error is considered for different interval ranges of longitudinal distance (specifically, in [0,10] m and [10,20] m). These range intervals cover the FoVs intersection, where the multi-sensor fusion occurs.
At first, a characterization of the sensor set has been carried on. In Fig. 3.13, the histogram in blue shows an indicator of the FrontCam error distribution. On the one hand, the reported error density supports the Gaussian hypothesis on the measurement noise and the model presented in Section 3.2. This analysis also guided the fine-tuning of the model's covariance matrix and α_M coefficient. On the other hand, we can observe that this result does not appear to be *zero mean* as supposed. This offset (in the centimeter magnitude) can be attributed to the relative mismatching between the lane-level ground truth and our global positioning system. In fact, this gap occurs likewise in the error distribution of the fusion result which is presented in red, always Fig. 3.13. This lightly narrower density shows the smoothing effect of the filtering process and the contribution of the complementary AVM sensor. More precisely, a comparison in term of mean, variance and Root Mean Square Error (RMSE) of the selected indicators has been finalized. The produced benchmark is presented in Table 3.1 and confirms previous observations. These results are based on a capsule (Section 2.4.3) of highway data where the HD-map was available. For 250 s of recorded sensor

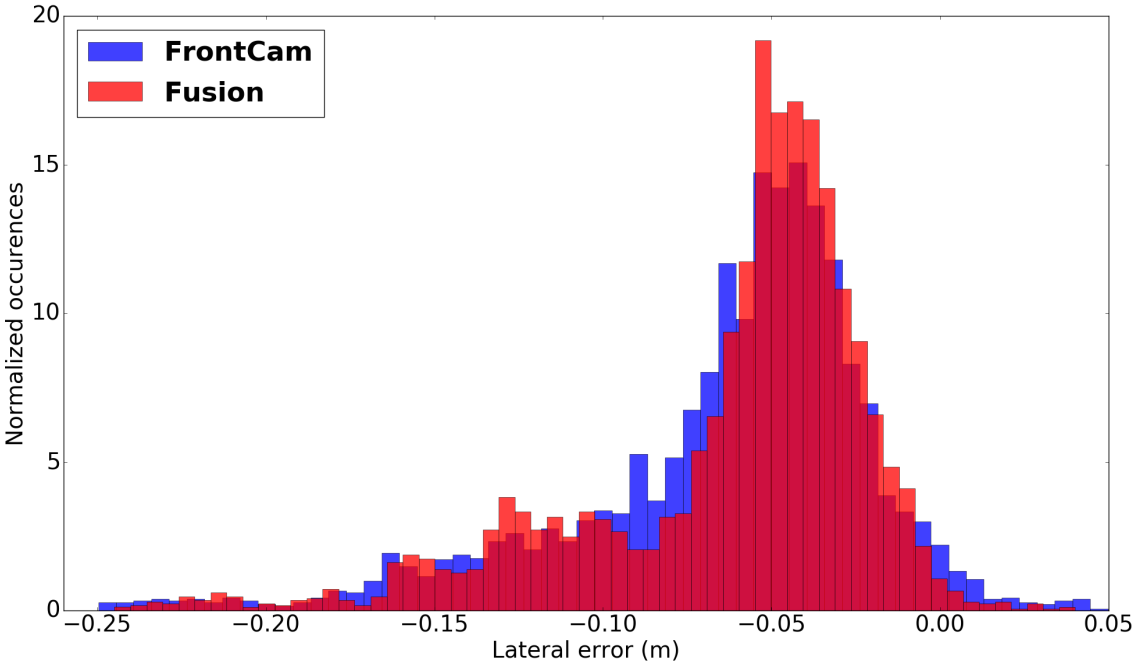


Figure 3.13: Superposed, normalized error distributions e_0^L for FrontCam (blue) and Fusion (see-throught, red)

data while driving the prototype vehicle at medium-high speed, the travelled distance amounts to about 7 km.

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

Table 3.1: Lateral error benchmark

3.4.3 On-board setup

This work-flow configuration is presented as a closed loop in Fig. 3.14. At this stage, all the computation are being executed on-board. The sensor deliveries are being directly processed in the on-board computer. The machine which is deployed on the L3 prototype runs RTMaps (development environment for multisensor applications) over a Win32 system. The latest implementation of the proposed solution is installed through a pre-compiled library containing all the module necessary for the target ADAS application. This architecture is designed to guarantee an easy migration to a smaller embedded computer (namely the Fusion ECU), which would replace the on-board PC in the (eventual) commercialized vehicle.

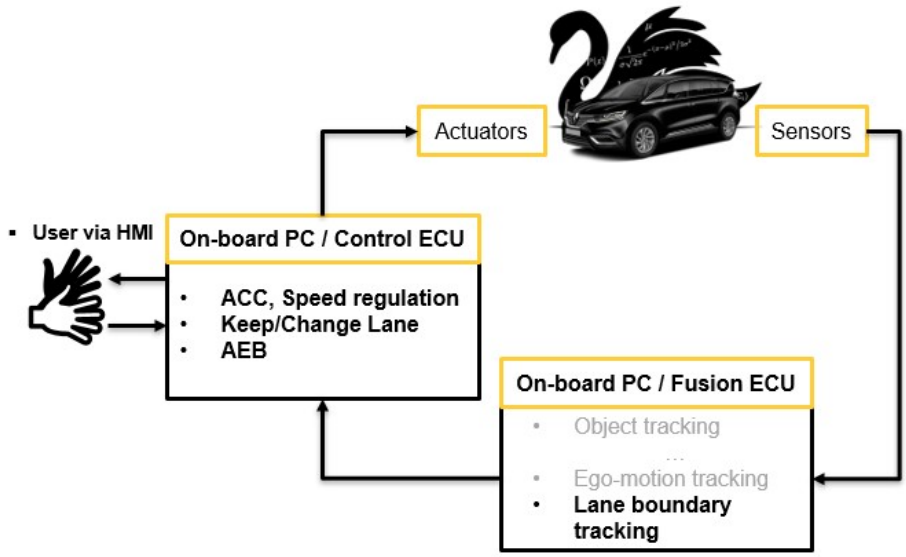


Figure 3.14: On-board setup. In this closed-loop configuration, Feature-tracking is performed in real-time. The clothoid-spline lane boundary tracks are used for the longitudinal and lateral control of the vehicle. The passenger can activated automated driving features such as ACC and Automated Lane Change

The experimentation in close-loop configuration featured lateral and longitudinal control algorithms developed by other teams in Renault. Feeding the control module with the ultimate output of our lane boundary tracking solution resulted in a successful execution of the target automated driving features of L3. First-left and first-right tracked lane boundaries (with respect to the ego-vehicle position) enabled LCA in highway scenario at elevated driving speed. When available, detection of further lane markings in the scene and availability of other lanes, our multisensor estimated lane repartition enabled automated lane change (upon HMI user activation via direction indicators).

3.4.4 Evaluation and results discussion

In the followings we discuss some of the most relevant obtained results.

Smart sensor measurements analysis Prior to the implementation of any tracking solution, a preliminary signal analysis has been carried on on the content of the smart sensor deliveries. Plotting the main components of the smart sensor model of Section 3.2, it is possible to drawn some considerations on the evolution of the polynomial coefficients issued of two different smart sensors over the same driving trajectory. In Fig. 3.15 the value for c_0 to c_3 of the left-side ego-lane boundary is presented over time. From the top to the bottom graph we can observe that:

- c_0 coefficient represents the lateral offset of the polynomial curve at abscissa $x = 0$. The two signals evolve in an equivalent manner. It could be pointed

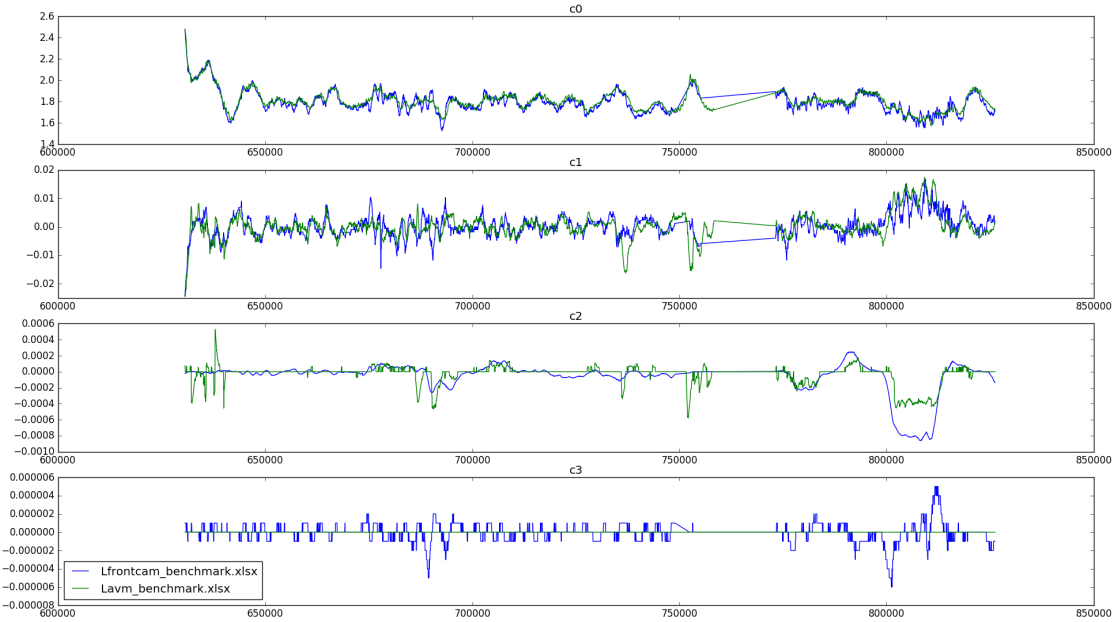


Figure 3.15: Left ego-lane boundaries detection: analysis of polynomial coefficients for FrontCam (blue) and AVM (green).

out that the AVM trend is occasionally smoother, which is understandable. The field of view of this sensor actually contains the $x = 0$ region, which allows for c_0 to be directly observable.

- c_1 coefficient represents the tangent of the polynomial curve at abscissa $x = 0$. Similar observations as for c_0 might be drawn as this coefficient is being direcly observed by the AVM sensor, in this case too.
- c_2 is an indicator of the curvature of the polynomial curve. In this case, the two signals are evolving quite differently. The FrontCam is apparently able to observe the curvature evolution in a smoother way. This could be due to the large and further range of its field of view. However, this smoothness could be also motivated by an heavier effect of the tracking in the smart sensor itself.
- c_3 is an indicator of the curvature rate of the polynomial curve. Evidently, the polynomials issued of the AVM deliveries are confirmed to be of second degree, as $c_3(t) = 0, \forall t$. For the FrontCam, it is observed that the possible values for this field are just a few and evidently discrete.

Time-correlated measurements and uncertainty An additional study of signal analysis has been carried on, within the development setup introduced in Section 3.4.1. In this case, the tracking result is being compared to the FrontCam measurements used in the estimation at that same instant. In order to better understand the causes of the observed phenomena, the result of this preliminary lane

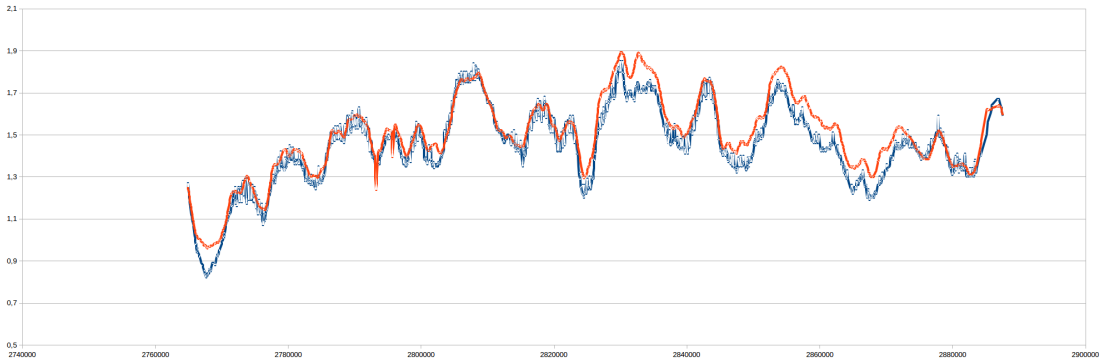


Figure 3.16: Poor tuning results in lane boundaries tracking over time. The tracked lateral offset (in red) becomes rapidly overconfident in its estimation with respect to the measurements (in blue).

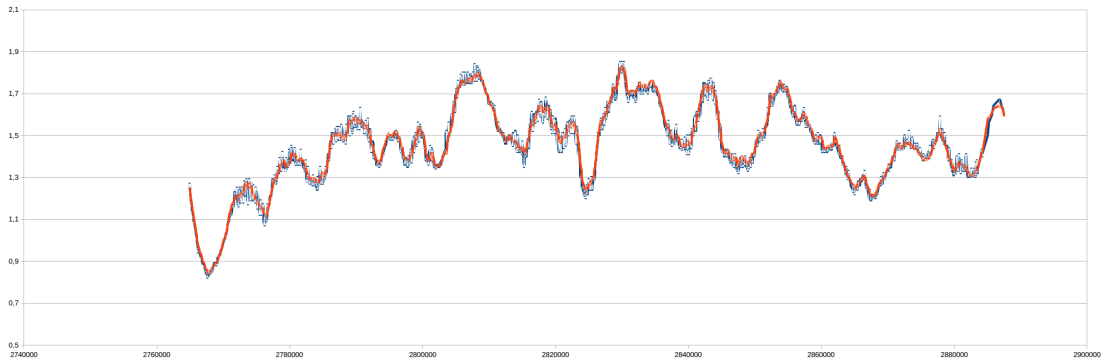


Figure 3.17: Correct tuning results over the same driving record of Fig. 3.16. The tracked lateral offset (in red) is more responsive to the measurements (in blue).

boundary tracking solution did not include the information from any of AVM deliveries. This example is indeed a case of single-source tracking of lane boundaries, where temporal fusion of FrontCam measurements is being deployed.

Given the above, the lateral offset between the measure (in blue) and the track (in red) is being compared, at the level of the front bumper abscissa ($x = 3.19m$).

In Fig. 3.16, the result of an initial version of the tracking algorithm is displayed. The track is smoothing the measure while being too rapidly confident in its estimate. The usage of a Kalman filter in the presence of temporally correlated measurements is indeed not ideal and may lead to a loss of consistency in the result. If the chosen information fusion method wants to be kept, this overconfidence issue can been addressed with two countermeasures: 1) the income of time-correlated measurements can be limited and/or 2) the measure noise model can be appropriately tuned. In our case, we decide not to limit the input sensor frequency. The measurement issued of both FrontCam and AVM, in the continuation, are all being processed applying an adapted uncertainty dilatation.

In Fig. 3.17, the result of the correctly tuned filter is displayed, which confirms its smoothing effect and do not falls victim of overconfident estimation.

Clothoid-spline model representativeness In Fig. 3.18, it is already possible to appreciate the flexibility of the model and a minimum correctness of the tracked lines. These are anyhow qualitative considerations based on the comparison of the tracking result and the associated context camera. As it may be expected, we confirmed that low traffic and straight road allow for a correct estimation of multiple lane markings up to a hundred meters. With traffic occlusions the range is reduced and lines that might be in the field of view are being missed.

Usage of HD-Maps as lane-level ground truth The RTK localization systems allows for a precise localization of the ego-vehicle using terrestrial antennae and post-processing with ad-hoc software. The obtained estimation of the vehicle pose is the de-facto ground truth for its positioning in the world frame, however this methodology does not give us access to a completely ideal lane-level representation in the body frame.

In order to exploit the information on the lane boundaries stored in the HD-Map according to WGS84 (World Geodetic System 1984, EPSG:4326), a transformation from the geographic coordinate system (GCS) is necessary. Two different transformations have been tested in these steps of generation of the ground truth: Mercator projection (TMERC) and Universal Transverse Mercator (UTM). Their effects are illustrated in Fig. 3.19 and easily observable. In the illustration, both sides show a measurement issued of the Smart FrontCam (polynomial curve in blue) which is reported in the same reference frame of an HD-Map of the surroundings. Sampled points along the curve are being projected onto the map line (green dots along the black line) of reference in order to compute the lateral error made by the FrontCam at different ranges. The lateral error for the closest point to the sensor ($x = x_{min}$) has been computed over time for the two considered transformations.

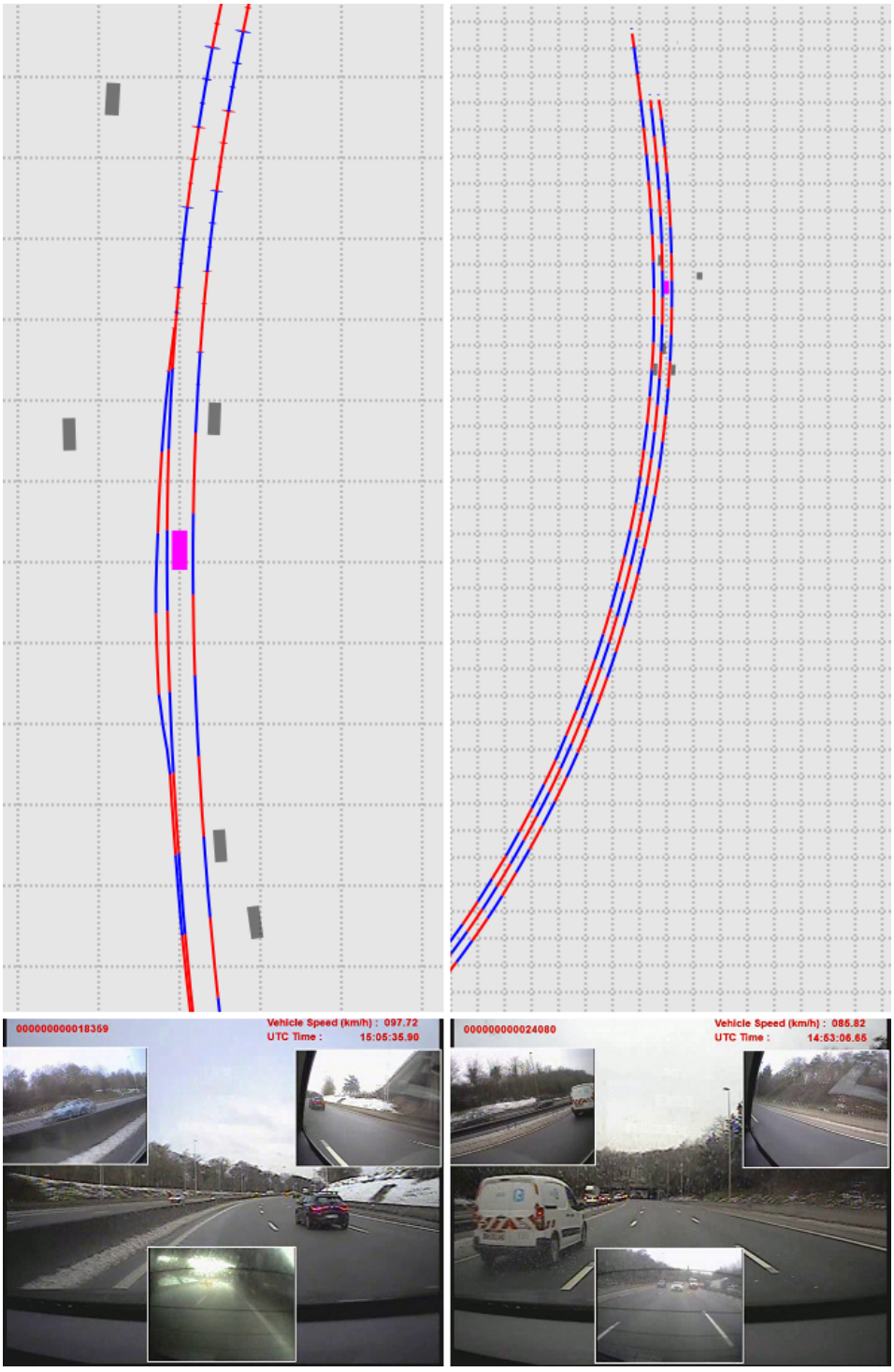


Figure 3.18: Two examples of tracked lines. Top: Ego-vehicle (purple box), other vehicles (grey boxes). Result (red and blue clothoid segments). Bottom: Corresponding context camera view.

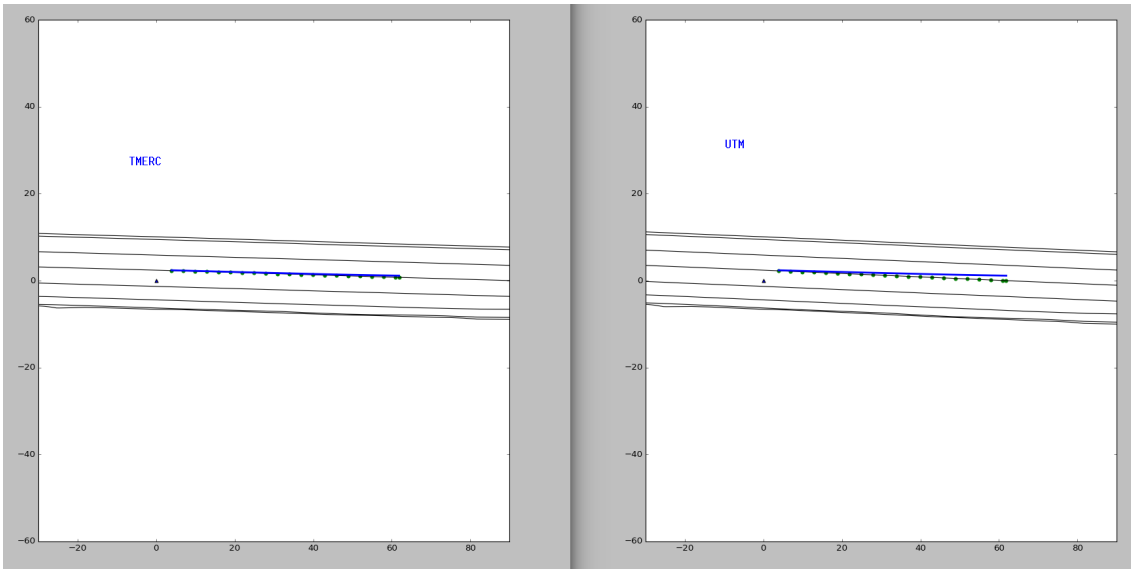


Figure 3.19: Comparing two different transformations for the generation of the ground truth: Mercator projection (TMERC) and Universal Transverse Mercator (UTM). Both sides show a measurement issued of the Smart FrontCam (polynomial curve in blue) which is reported in the same reference frame of an HD-Map of the surroundings. Sampled points along the curve are being projected onto the map line (green dots along the black line) of reference.

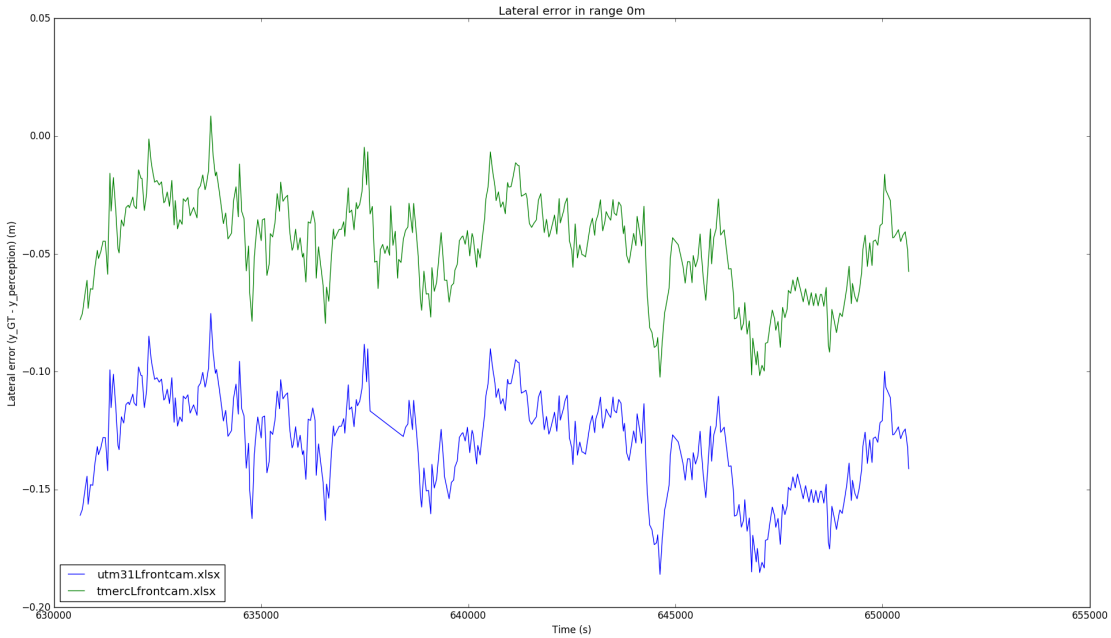


Figure 3.20: The lateral error for the closest point to the sensor ($x = x_{min}$) has been computed over time for the two considered transformations. The discrepancy between TMERC and UTM impacts the computation of the lateral error of about 10 cm.

Figure 3.21: Left: tracking result (red and blue clothoid segments). Right: context camera. The fourth track from the left has been initialized from a FrontCam false detection. Noisy measurements are mistakenly accepted but correctly not associated to existing tracks.

The discrepancy between TMERC and UTM impacts the computation of the lateral error of about 10 cm, as the graph in Fig. 3.20, which is enormous relatively to our application. The UTM transformation has been finally adopted, yet this analysis highlighted the fact that the uncertainty of each distinct operation applied to the involved data may have a huge impact on the final outcome.

False detections robustness Mistaken associations are not frequent, yet in Fig. 3.21 we can see the effects of noisy measurements. At the moment, the tracking algorithm is not robust enough to reject them. As they are not being associated to any existing track, they generate new ones leading to poor results.

3.5 Conclusions

In this chapter, a multi-sensor tracking approach for generic lane boundaries is proposed. Although the experimental fusion of two smart sensors reported only a slight improvement in term lateral RMSE, the solution confirmed its validity and coherency w.r.t. the lane-level ground truth. Its real-time implementation and execution can support potentially any multi-modal smart sensor set, providing redundancy and perception diversity in the overall lane geometry estimation.

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

Chapter 4

Map-aided multi-sensor fusion for lane boundaries estimation

4.1 Introduction

In this chapter, we approach the lane boundaries estimation problem in the case where cartographic information is additionally made available. We will be recalling the system architecture within this use case and presenting the map-provider from the data fusion perspective. This additional component, the operation of which is described Section 2.3, has its output modelled according to the involved uncertainties. Both mapping and localization processes come, in fact, with possible inaccuracies which have to be taken into account before processing the data in the identification of the surroundings.

At first, we exploit the map data in order to filter false-positive measurements issued by smart sensors. This application helps in sanitizing the input of the previously presented fusion method, in case of up-to-date map and in-what-lane correct localization. Where this second hypothesis is called into question, an ad-hoc indicator is defined to assess the coherence between the two sources and possibly point out a localization fault.

These applications all consider the contribution of the map as a support to a sensors-centered approach. We also extend the previously presented lane boundaries tracking solution with an outlook of *HD-map as a sensor*, resulting in effective tracking of road heading and curvature on map waypoints.

4.2 Map-provider model

In Section 2.3, we reviewed some of the existing works in mapping and localization. These are the two assets which are essentials in the design of a so-called map-provider. With a modular architecture in mind, we define "map-provider" a set of technologies able to produce a representation of the vehicle surroundings in the form of a map, at a given time t.

Les us define the global frame F_O, the frame in which the map has been originally built at mapping time. As we are focusing on lane boundaries estimation,

these are the only elements of interest. However, depending on the target application, additional entities or attributes may be included in the cartographic representation. At this time, we consider the minimal amount of information relevant to our task. A lane boundary L is defined as a collection of landmark points ${}^{O}\mathbf{X}_{i=1..N_i}$ describing a polyline. At mapping time, the attributes of each ${}^{O}\mathbf{X}_{i=1..N_i}$ are identified and fixed in the global representation in F_O. Since this estimation process is accurate but generally not perfect, we are interested in carrying on a measure of this inaccuracy. Under the Gaussian hypothesis, the covariance matrix $Var({}^{O}\mathbf{X}_{i=1..N_i})$ completely describes the distribution and can be graphically presented in ellipses form, as in Figure 4.1. This description of the lane boundaries supposes the mapping process to be able to estimate the uncertainty associated to each of the landmarks ${}^{O}\mathbf{X}_{i=1..N_i}$ and allows for this uncertainty to be different on every occasion. In fact, the estimation of each landmark may be affected by weather, number of observations or other factors specific to the mapping process. These factors may include impact of roadworks, road network changes and even continental drift. Given the complexity in monitoring and modelling these occurrences, it might be more practical to indicate a general measure of accuracy of a whole map. In this case we are supposing that:

$$Var({}^{O}\mathbf{X}_i) = Var({}^{O}\mathbf{X}_{Map}) \forall i = 1..N_i \tag{4.1}$$

It can be noted that the number of landmarks per lane boundary N_i might also be specific to the mapping process. In general, the density of landmarks along the length of each curve is not constant. To offer the best trade-off between amount of points (hence storage space) and expressiveness of the polyline, this density generally varies with the curvature of the road.

Les us now define the body frame F_M. Attached to the ego-vehicle, its origin is conventionally located at the middle of the rear axis of the car, as illustrated in Figure 4.1. It is the reference frame for our target application which implies that positions and attributes of all relevant elements surrounding the vehicle will have to be eventually expressed in F_M. Map information is no exception and it is the purpose of the localization process to determine the correct transformation from F_O to F_M. Specifically, given a map positional landmark ${}^{O}\mathbf{X}_i = \begin{bmatrix} {}^{O}x_i \\ {}^{O}y_i \end{bmatrix}$ in its original frame, its position is reported in the body frame following the coordinates transformation based on the ego-vehicle pose:

$${}^{M}\mathbf{X}_i = {}^{M}\mathbf{R}_O \left(\begin{bmatrix} {}^{O}x_i \\ {}^{O}y_i \end{bmatrix} - \begin{bmatrix} {}^{O}x_M \\ {}^{O}y_M \end{bmatrix} \right) \tag{4.2}$$

where ${}^{M}\mathbf{R}_O$ represents the rotation matrix from F_O to F_M:

$${}^{M}\mathbf{R}_O = {}^{O}\mathbf{R}_M{}^{T} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.3}$$

Equation 4.2 describes the coordinate transformation which consists of two operations. Translation and rotation are applied and they are based on the estimated pose of the ego-vehicle within the map at issue. This transformation opens up

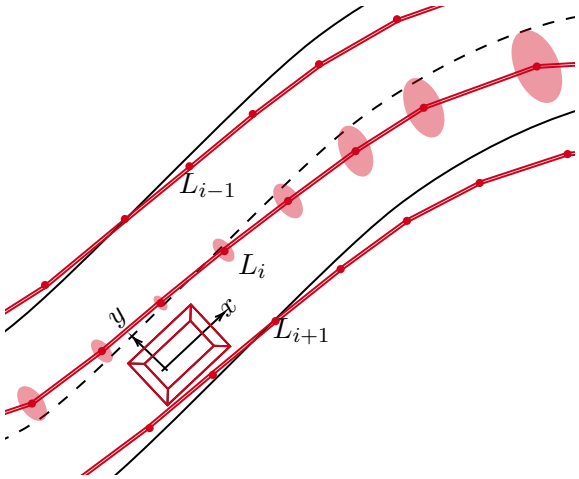


Figure 4.1: Map-provider model. Lane boundaries are given in the local frame F_M following the geometrical transformation based on ${}^{O}\mathbf{X}_M$. For L_i, light red ellipses show how the uncertainty evolves, notably influenced by σ_θ^2.

for more uncertainty in the representation, in terms of the three random variables involved: ${}^{O}x_M$,${}^{O}y_M$,θ.

Based on these considerations, we give in the followings the details of the delivery issued by the map-provider. Its content is dimensioned to be minimal in the size and complete with the aim of an adequate representation of the involved uncertainties. It follows a study of the evolution of positional uncertainty of the map landmarks with typical values of mapping and localization standard deviations. Finally, the adopted method for lane boundaries association is detailed, before presenting its application in the next sections.

4.2.1 Use case map-provider

As for smart sensors, the integration of a map-provider for the implementation of driving assistance features can be done within an existing pipeline in the form of a closed sub-system, or a black box. In our research, the map-provider is being developed internally. We are therefore able to describe its functioning before characterizing its deliveries. Let us now outline some of the key elements for the workflow of this unit:

- **HD map database**: the map-provider disposes of an entire HD-map of target ODD. The ODD of the use case application is highway driving. Specifically, the involved map manufacturer produced an HD map for all highways of France.
- **Inputs for in-map localization**: for this purpose multiple information are being processed in order to provide the most accurate ego-localization. At first, a non-precise ego absolute localization is delivered by a traditional GNSS unit. This is the absolute positioning before any treatment or dead-reckoning. Afterwards, perception measurements (such as lateral offsets or

landmark positions) are being received. This is supposed to be issued of a separated sensor set reserved for localization (e.g. the front lidar mentioned in Section 2.4.2 for barriers and landmarks is not used for lane boundaries estimation).

- **Map-matching and position filtering**: starting from the GNSS measurement, generated localization candidates (or particles) are assigned a different weight wheather the detected landmarks match the HD map content or not. In addition, particle filtering for localization generally uses vehicle direction of the closest centerline for map-matching.
- **Localization candidates**: the best (or multiple best) candidates for the vehicle localization position are being selected. It is noted, however, that if we suppose to keep N multiple candidates then it would imply to perform N transformations and to deliver N map-provider deliveries to the client (fusion) module. Map-provider deliveries are ego-centered, content can be essentially different for a distinct candidate position.
- **Electronic horizon computation**: it is performed a map transformation from absolute to relative coordinates. This is the specific transformation affected by localization uncertainty and detailed in Section 4.2. It is additionally affected by mapping approximations and modeling such as UTM and Mercator. Finally, cropping and selection in relative HD-map elements is done according to client module requirements. In our use case, the Most Probable Path (MPP) is being kept together with child roads at one-level depth. Map way-points are limited to 500m in front of the vehicle and 100m at its back. The content of the electronic horizon is also limited by specific message size and bandwidth requirements. Fixing the ego position in the map implies the definition of the ego-lane and ego-road.
- **Map-provider wrapping and delivery**: the map-provider delivery format is filled with lane boundaries, mapping uncertainty (per point if available) and localization uncertainty.

4.2.2 Map-provider delivery

The idea is to consider the localization module (or map provider) just as another smart sensor, able to supply a representation of some elements of the surrounding environment. As for smart sensor with lane detection capabilities, the elements of interest are lane boundaries presented in the local frame (F_M frame, with origin located at the middle of the rear axis of the car). Let $\mathbf{z_t}$ be the sensor delivery supplied by the map provider at instant t, featuring the following components:

1. $t_{measuredate}$
2. ${}^{O}\mathbf{\Sigma}_M$
3. ${}^{M}L_{l=1..N_l}$

Let us now details the meaning of the quantities in the sensor delivery.

A $t_{measuredate}$

For simplicity, we suppose $\mathbf{z_t}$ measurements to be available at each instant t that might be appropriate for the lane boundaries estimation process (for example, to allow temporal alignation with other sensor deliveries). In practice, these sensor deliveries depend on the specific localization module and its design. We can equivalenty suppose the existence of a reliable evolution model of the ego-movement, which enables us for a "backwards prediction" (via application of the appropriately estimated rotation and translation) of that sensor delivery at any t. However, each sensor delivery refers to a specific instant in time $t_{measuredate}$ where the set of lane boundaries ${}^{M}L_{l=1..N_l}$ has been observed to be surrounding the ego-vehicle. In case of a smart camera, it would refer to the timestamp attached to the raw image at the moment of capture. In case of a map provider, the meaning of this quantity is different. We could be more interested to the timestamp attached to the pose issued of the localization process of the car: the position ${}^{O}\mathbf{X}_M = \begin{bmatrix} {}^{O}x_M \\ {}^{O}y_M \end{bmatrix}$ and orientation θ, in the global frame F_O. Discussing instead the "moment of capture" of map landmarks would mean to refer to the instant of map construction, process that generally happens way far before its actual deployment in a working architecture (with the exception of real-time crowdsouced map servers).

B ${}^{O}\mathbf{\Sigma}_M$ **localization uncertainty**

The previously mentioned quantities ${}^{O}\mathbf{X}_M$ and θ are random variables and their distributions impacts the exploitability of the set of lane boundaries ${}^{M}L_{l=1..N_l}$ issued by the map-provider. In this case, we consider ourselves as clients of this external unit. That is the reason why we will not be interested in (nor aware of) the actual pose of the vehicle in F_O. In fact,

$$ {}^{O}\mathbf{\Sigma}_M = Var(\begin{bmatrix} {}^{O}x_M \\ {}^{O}y_M \\ \theta \end{bmatrix}) = \begin{bmatrix} \sigma_x^2 & {}^{O}p_{12} & {}^{O}p_{13} \\ {}^{O}p_{21} & \sigma_y^2 & {}^{O}p_{23} \\ {}^{O}p_{31} & {}^{O}p_{32} & \sigma_\theta^2 \end{bmatrix} \tag{4.4} $$

would be enough to characterize the localization uncertainty, whereas the random variables are unbiased and Gaussians. We are specifically interested to the standard deviations of the longitudinal and lateral localization of the vehicle, respectively σ_x and σ_y. Likewise, the uncertainty on the orientation of the vehicle σ_θ affects the usability of the map-provider delivery.

C ${}^{M}L_l$ **lane boundary**

Each ${}^{M}L_{l=1..N_l}$ is a curve representing one of the lane boundaries in the environment. It is defined as follows:

- ${}^{M}\mathbf{X}_{i=1..N_i}$, $Var({}^{O}\mathbf{X}_{i=1..N_i})$ describing probabilistic position and distribution of its forming map waypoints

- $L_{type} \in \{marking, barrier, curb, ...\}$ describes the type of the detected road element

In the case of map-providers, the lane boundary model is a finite collection of landmark points ${}^{M}\mathbf{X}_i = \begin{bmatrix} {}^{M}x_i \\ {}^{M}y_i \end{bmatrix}$. Where available, the spatial uncertainty that comes with each map point ${}^{O}\mathbf{X}_i$ can be expressed with the following covariance matrix (referring to the mapping error, supposed to be Gaussian):

$$Var({}^{O}\mathbf{X}_i) = \begin{bmatrix} var({}^{O}x_i) & {}^{O}p_{12} \\ {}^{O}p_{21} & var({}^{O}y_i) \end{bmatrix} \tag{4.5}$$

This contributes to the uncertainty of each point ${}^{M}\mathbf{X}_i$ which is expressed with respect to the observer (ego-vehicle) and also affected by the localization process of the car within the global frame F_O. In fact, as from [Yu et al., 2020]:

$${}^{M}\mathbf{X}_i = \begin{bmatrix} {}^{M}x_i \\ {}^{M}y_i \end{bmatrix} = {}^{M}\mathbf{R}_O \left(\begin{bmatrix} {}^{O}x_i \\ {}^{O}y_i \end{bmatrix} - \begin{bmatrix} {}^{O}x_M \\ {}^{O}y_M \end{bmatrix} \right) = f({}^{O}\mathbf{X}_M, \theta, {}^{O}\mathbf{X}_i) \tag{4.6}$$

where ${}^{M}\mathbf{R}_O$ represents the rotation matrix from F_O to F_M:

$${}^{M}\mathbf{R}_O = {}^{O}\mathbf{R}_M{}^{T} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.7}$$

giving:

$$f({}^{O}\mathbf{X}_M, \theta, {}^{O}\mathbf{X}_i) = \begin{bmatrix} ({}^{O}x_i - {}^{O}x_M)\cos(\theta) + ({}^{O}y_i - {}^{O}y_M)\sin(\theta) \\ -({}^{O}x_i - {}^{O}x_M)\sin(\theta) + ({}^{O}y_i - {}^{O}y_M)\cos(\theta) \end{bmatrix} \tag{4.8}$$

We model the transfer of uncertainty from the five random variables ${}^{O}\mathbf{X}_5 = ({}^{O}\mathbf{X}_M, \theta, {}^{O}\mathbf{X}_i)$ to each map point ${}^{M}\mathbf{X}_i$ with a first order approximation of equation (3):

$$Var({}^{M}\mathbf{X}_i) = \left[\frac{\partial f}{\partial {}^{O}\mathbf{X}_5} \right] \begin{bmatrix} {}^{O}\mathbf{\Sigma}_M & \mathbf{0} \\ \mathbf{0} & Var({}^{O}\mathbf{X}_i) \end{bmatrix} \left[\frac{\partial f}{\partial {}^{O}\mathbf{X}_5} \right]^T \tag{4.9}$$

where the Jacobian for f results in

$$\left[\frac{\partial f}{\partial {}^{O}\mathbf{X}_5} \right] = \begin{bmatrix} -\cos(\theta) & -\sin(\theta) & -({}^{O}x_i - {}^{O}x_M)\sin(\theta) + ({}^{O}y_i - {}^{O}y_M)\cos(\theta) & \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) & -({}^{O}x_i - {}^{O}x_M)\cos(\theta) - ({}^{O}y_i - {}^{O}y_M)\sin(\theta) & -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.10}$$

=

$$\begin{bmatrix} -\cos(\theta) & -\sin(\theta) & {}^{M}y_i & \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) & -{}^{M}x_i & -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.11}$$

This expression allows the computation of $Var({}^{M}\mathbf{X}_i)$ for any map point given in frame F_M by the map provider, whereas it is delivered with the corresponding localization uncertainty ${}^{O}\mathbf{\Sigma}_M$ and map (or single landmark) uncertainty $Var({}^{O}\mathbf{X}_i)$.

4.2.3 Study of uncertainty

As uncertainty of the map around the vehicle can be calculated, here in the following its evolution with example plausible values is presented. The examples show how the uncertainty evolves around the ego-vehicle on map-provided lane boundaries. Hereinafter, the ego-vehicle is positioned and centered in its lane where the road is composed of other two lanes per-side. The lane boundaries are presented in blue, the presence of a distinct map node is supposed at 5 meters interdistance. For each map node, the uncertainty of its position is represented with an ellipse based on the actual values in its covariance matrix $Var(^{M}\mathbf{X}_i)$ The resulting uncertainty ellipses are shown in different figures for different values of orientation standard deviation σ_θ, which evidently influences the outcome on the distance.

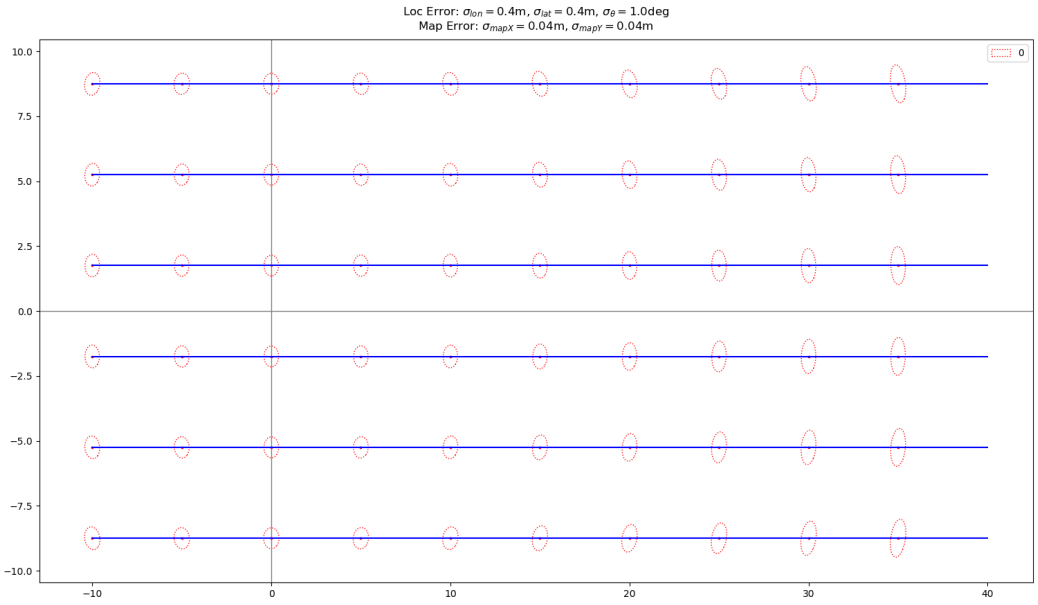


Figure 4.2: Position uncertainty ellipses for localization error of heading σ_θ=1°

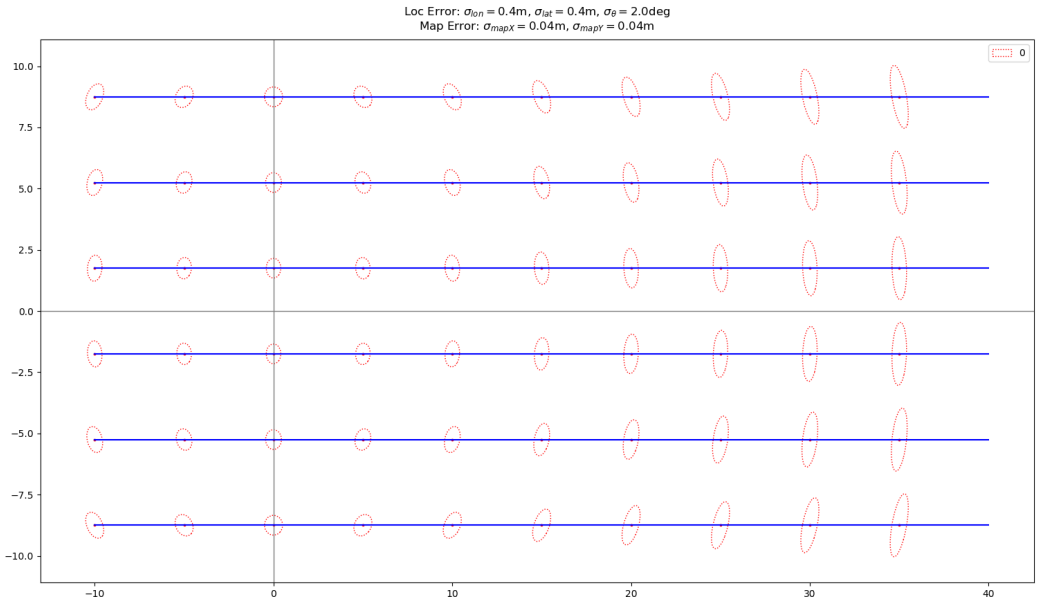


Figure 4.3: Position uncertainty ellipses for localization error of heading σ_θ=2°

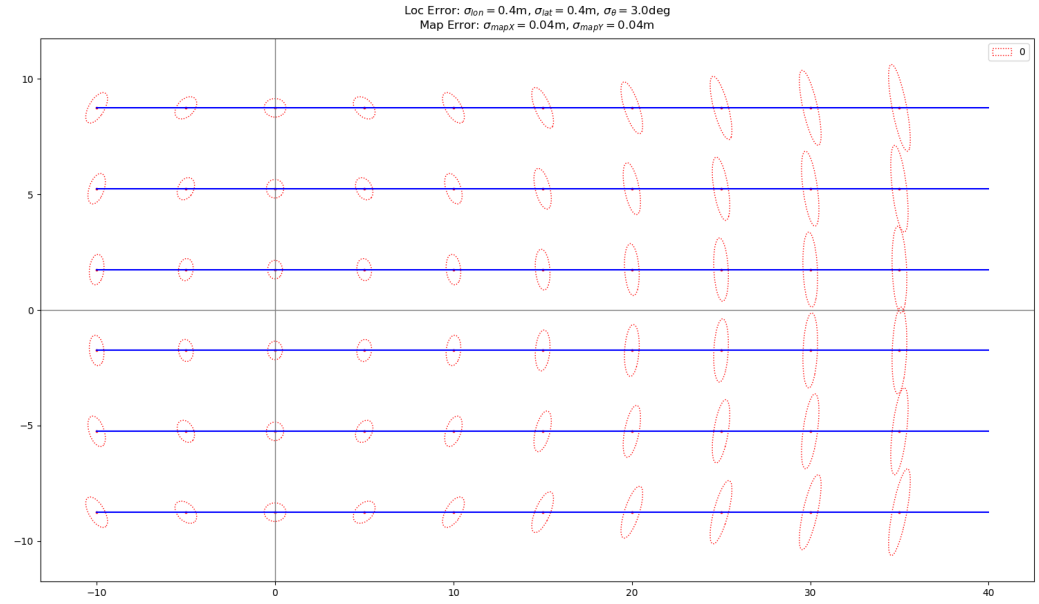


Figure 4.4: Position uncertainty ellipses for localization error of heading σ_θ=3°

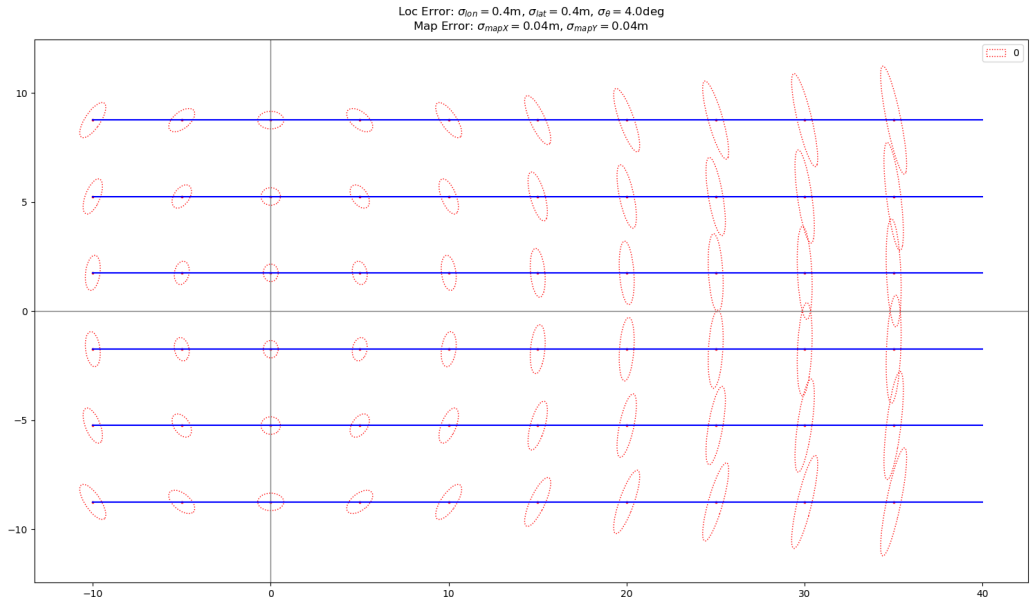


Figure 4.5: Position uncertainty ellipses for localization error of heading σ_θ=4°

It is noted that for short distances the impact of the heading error is negligible, as expected. This behaviour can be observed equivalently for any of the studied values of σ_θ standard deviation, from Fig. 4.2 to Fig. 4.5. However, for higher longitudinal distances, the heading uncertainty rapidly affects the usability of the map-provider delivery. In Fig. 4.5, the overlapping of 3σ-ellipses is already visible at relatively short range such as 25 meters in longitudinal distance.

4.2.4 Notes on uncertainty of the lane boundaries representation and integrity

In the previous, the proposed model allowed for taking into account both localization and mapping uncertainty in the usage of HD-map data. The estimation of the uncertainty associated with a given lane boundary representation can be enabler for the definition of a perception integrity framework where safety is ensured under expressed risk conditions. In fact, in civil aviation application, the localization of the ego-vehicle (aircraft) is subject to state integrity monitoring for guaranteeing system safety. In this work, the study of perception integrity has not been deepen, however some fundamental notions of the state integrity monitoring can be transported from localization integrity monitoring [Reid et al., 2019] to perception integrity monitoring.

The goal here, differently from ego-localization estimation, is to estimate the state of a third object in the environment and to be able to represent the uncertainty of this estimation in a measurable manner that also expresses the risk implied by this uncertainty. This third object being a road feature F with state vector $[x, y, \theta]$, the final application that uses the representation of the environment in the form of road features should be able to specify a maximum accepted Integrity Risk and the Alert Limits (AL). Depending on the safety requirements

which are characteristics of the system application, these quantities would imply the definition of a perception integrity framework where:

1. At first, the Protection Levels (PL) based on state variable uncertainty are computed such that the probability of the PL to violated is < Integrity Risk. This is done for each state variable, defining:
   - Longitudinal PL for variable x
   - Lateral PL for variable y
   - Heading PL for variable θ
2. If PL > AL, the integrity monitoring system throws a warning noticing the managing unit that safety is not ensured using that road representation

The implementation of this framework would require proper uncertainty estimation and correct definition of the safety requirements from the application. Future developments in the definition and application of perception integrity monitoring could enable safe and large scale deployment of automated systems graded SAE Level 3 and higher.

4.3 Proposed solution: Map-tracking

So far, we discussed and detailed the modelling of map-provider output for it to be processed in a fusion module. Equivalently, in Section 3.2 the smart sensor delivery has been presented. These are the two templates for information sources supposed to be available in the use case here discussed. In the following, the process designed for combining these different sources is presented.

This use case is essentially different from the first that been addressed. Multiple inputs are being processed and, additionally, they are issued of "providers" of different nature. Where in Section 3.3 the deliveries from Smart FrontCam and Smart AVM came from distinct sensors, the adopted model was the same. As the map-provider operates differently, its deliveries are also processed in another way.

In Fig. 4.6, a graph represents the interactions among the components in the solution and the temporal evolution of the lane boundaries estimation $\mathbf{x_t}$. In this realization, we will be referring to $\mathbf{x_t}$ by the name of map-track. The map-track $\mathbf{x_t}$ is still a set of tracked lane boundaries, each of them is described by map waypoints named road features F in the tracking procedure. In the illustration, the information converges towards the nodes $\mathbf{x_t}$ where the incoming links indicate the available information at the iteration t.

At any given instant, distinct operations are being executed accordingly with the latest sensor delivery received. Specifically:

- Map-provider (first) : initialization
- Ego-movement : prediction
- Smart Sensor delivery : association (and update)
- Map-provider (new) : update

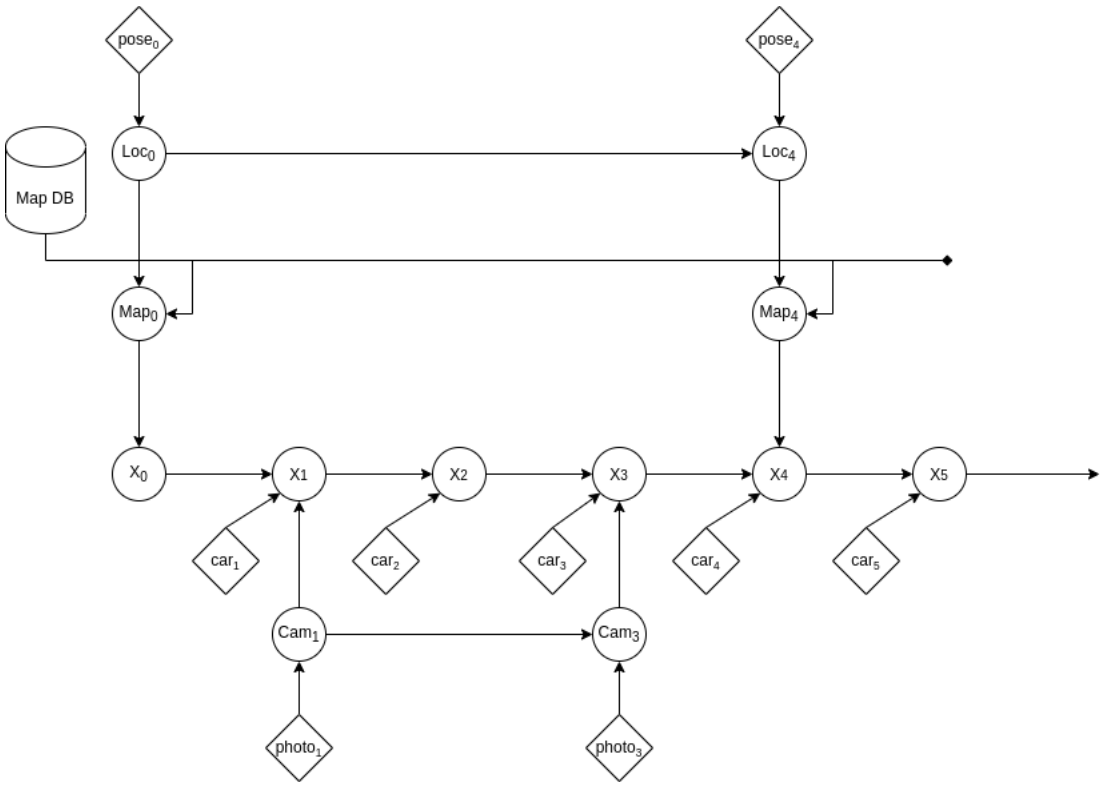


Figure 4.6: Smart FrontCam and map-provider deliveries are asynchronous. At time instant i, the estimation of the surrounding lane boundaries x_i takes into account current measurements and previous estimations.

4.3.1 Initialization

As in the case presented in Fig. 4.6, (at node $\mathbf{x_0}$) the initialization of the first map-track is conditional to first delivery of the map-provider. Where this delivery is not presented, the lane boundaries tracking would be in fact downscaled to the method illustrated in the previous chapter, as the input to be processed would be entirely issued of smart sensors.

According to the model in Section 4.2, the same information can be conveniently presented in form of road features, and initializes the map-track as follows:

$$F_i = \begin{bmatrix} {}^M x_i \\ {}^M y_i \\ \bar{h} \\ \bar{c} \end{bmatrix}, {}^M\mathbf{\Sigma}_F = Var({}^M\mathbf{X}_i) \tag{4.12}$$

The structure of the map-track reflects the format used by the map-provider for its deliveries. The road features model is adopted and the uncertainty on each state variable is expressed in the body frame F_M. With respect to the previous tracking solution, we will be additionally tracking the curvature of the road at that given point. However, as map-provider deliveries generally contains only positional information about map waypoints, the heading $\bar{h}$ and curvature $\bar{c}$ state

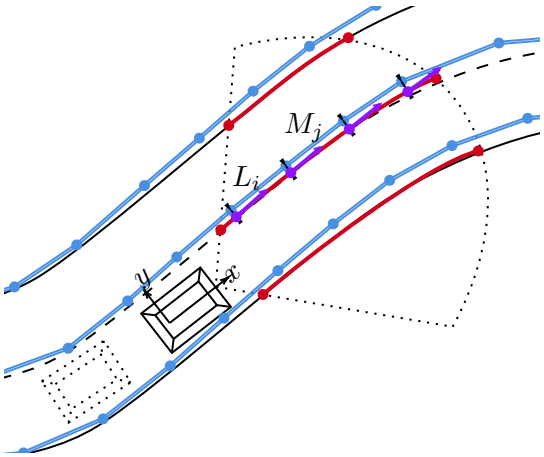


Figure 4.7: Road features from the map-track (in blue) are being projected onto the latest sensors measurements (in red). Association distances are computed according to the uncertainty models and the result of the projection $p_\perp(F_j)$ (in purple).

variable will to be properly initialized until a successful association of the map-track line with a smart sensor measurement.

4.3.2 Prediction

Ego-movement estimation is used to predict the map-track. Similarly to Section 3.3.2, the estimated rotation and translation of the ego-vehicle is applied to the content of the map-track. Accordigly, the covariance matrix ${}^{M}\mathbf{\Sigma}_F$ is updated following the prediction step of the Kalman filter. This operations implies that if the estimation of the ego-movement is especially uncertain, the position (and the other state variables) of any given road features around the car would be consequently uncertain.

4.3.3 Association

When matching lane boundaries detections from different sources, positional information is not the only clue we can lean on. In fact, both HD-maps and smart sensors provide type information for each presented lane element. This classification allows for an initial screening of the possible matches between candidates. As step-by-step detailed in Algorithm 1, we shall consider an association to be viable when the lane boundary types M_{type} and L_{type} do correspond (for example, they are both lane markings), or when M_{type} is unknown.

It may occur that an on-board sensor has not been able to label all the measurements within one of the classification possibilities. This preliminary test enhances correctness and considerably saves computational time. At this point, we are dealing with two probabilistic representations of the lane boundaries surrounding the ego-vehicle and presented in the local frame. In order to identify any existing correspondence between the two representations, they shall be expressed in a comparable form. Smart sensor measurements are expressed with a

Algorithm 1 Lane boundaries association

```
procedure ASSOCIATE(z_t^Sens, z_t^Map)
    for each measure M_i in z_t^Sens do
        for each map line L_j in z_t^Map do
            if M_type = unknown or M_type = L_type then
                D_{i,j} ← d(M_i, L_j)
            else
                D_{i,j} ← dmax
            end if
        end for
    end for
    Return GNN(D,dmax)
end procedure
```

continuous model, whereas the information given by the map-track is punctual - discrete. Let us make use of this, looking for a correspondent in the sensor measurements per each of the road features in the map-track. It is supposed that a sufficient number of map points is available for each lane boundary, allowing at least one orthogonal projection of these features on the measurement, if they represent the actual same lane boundary. Under this assumption, each map feature is projected on each of the measurements as illustrated in Fig. 4.7.

Each successful projection of $F_j \in L$ appoints to a corresponding road feature $p_\perp(F_j)$ sampled along the measure M at $(x_\perp, y_\perp)$. The Mahalanobis distance between the two can be computed as:

$$d^2(p_\perp(F_j), F_j) = (p_\perp(F_j) - F_j)^T({}^M\mathbf{\Sigma}_F + Var({}^M\mathbf{X}_i))^{-1}(p_\perp(F_j) - F_j) \tag{4.13}$$

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

$$d(M, L) = \max_{F_j \in L} d(p_\perp(F_j), F_j) \tag{4.14}$$

A distance matrix built on these values allows the use of a Global Neighrest Neighbourd (GNN) algorithm to finalize the association (implementation details included in Appendix B).

4.3.4 Update (smart sensor delivery)

In the case of smart sensor delivery, the content of the map-track can be updated but it is possible to consider different strategies for including this information in the estimation.

Road feature labelling After completing a successful association between a smart sensor measurement M and a map-track lane boundary L, it is confirmed that the two represent the same physical object. In this update strategy, it is proposed not to update the state variables of the tracked road features with the smart sensor

measurements. However, we want to store the fact that an on-board sensor has recently observed the existence of that (set of) road features. This association gives more credibility to the map-provided information, which is confirmed by the freshest environmental observation.

Successfully associated road features are being, therefore, labelled with a unique identifier per each sensor in the current architecture. In our case, labels assigned to each road features may be: $\{\emptyset, FrontCam, AVM, \{FrontCam, AVM\}\}$. A road feature observed by the whole sensor set would give maximum confidence to the estimation.

Road feature state update Updating the state variables of the road feature, which have been firstly generated from a map-provider delivery, with sensor measurements would imply to put on the same level the to kind of processed input. In this strategy, the road feature state variables of the map-track F_j are updated with their corresponding equivalent feature on the measurements, previously indicated with $p_{\perp}(F_j)$. Again, as in Section 3.3.4, the Kalman update step is adopted for the information fusion. In this case, the implications of indifferently fusing sensor and map data may be hazardous. The accurate (and occasionally hand-built) topological structure of the map is put at risk while inaccurate sensor measurements are being integrated.

Road feature partial state update The idea of partially integrating the information from sensor measurements may be wiser and easier to maintain. It is proposed to update a subset of the state variables in a road feature. The approach is different for qualitative and quantitative state variables. For the former, while updating the lane boundary type variable L_{type}, more confidence may be attributed to the live observed information which would overwrite the map-stored lane boundary type. The latter, the actual state variable of the road feature, would be updated as previously indicated, though leaving unchanged critical state variables. This approach would preserve the map topology when the subset of updated variables do not include x and y which indicate the spatial position of the road feature.

4.3.5 Update (map-provider sensor delivery)

A new map delivery would generally add new information at a distant range and crop out a section of the past path. As the delivered subsection of the map depends on the vehicle position, new lanes and roads may be included other than new map points for the existing lane boundaries. Nodes of the map that were already being tracked in the current map-track would also be present and the tracking process for those nodes do not need to be reinitialized.

For these reasons, unique identifiers for map nodes are adopted in order to match the initialized nodes in the current map-track with their counterpart in the map delivery. In the absence of a unique identifier, the original absolute coordinates of the map node in the global frame may be a valid replacement for the matching.

4.4 Experimental results

In this section, experiments and tools deployed for testing our models and algorithm are presented. At first, we recall the technical setup which includes currently relevant vision sensors and a development platform. It follows the description of two experimental applications that take advantage of the proposed methodology and serve for evaluation: anomalies detection for smart sensors and multi-hypothesis localization scoring. Lastly, adopting the road feature partial state update strategy for heading and curvature tracking, the effect of the filtering are measured against two distinct references.

4.4.1 Setup

The core equipment useful for the completion of our tests is installed on our prototype autonomous vehicle (an ad-hoc configured Renault Espace). It is equipped for the perception of its surroundings with:

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

These sensors implement device-specific data processing algorithms and issue measurements in the format specified in Section 3.2. Specifically, the Smart FrontCam can send out up to 4 measurements per delivery. The Smart AVM has similar capabilities, however we narrow down its usage to the ego-lane estimation (up to 2 measurements). This limitation is due to the difference in maturity of the two sensors.

Another specific unit handles the localization of the vehicle and function as a map-provider. It implements particle-based localization algorithms applied within a remotely provided HD-map (streamed via Ethernet according to ADASISv3 [Ress et al., 2008] standards), using a commercial positioning system. Its output is presented in the local frame and complies with the probabilistic model proposed in Section 4.2. Sensor output and map data has been recorded for testing, driving the prototype vehicle on the French A86 (Créteil-Versailles).

The lane boundaries association algorithm, detailed in Section 4.3.3, has been implemented in a proprietary environment for ADAS design. On this platform, the development is carried out mostly in C language, following MISRA C [MIRA, 2008] guidelines for safe, reliable and portable code for embedded systems. The presented solution is, in fact, specifically designed to comply with real-time constraints and low resources availability typical of ADAS ECUs. An interface example of the platform is reported in Fig. 4.8. The video stream shows a view of the driving scene at each instant and it can be used for a qualitative evaluation of the results, as it is shown hereinafter.

4.4.2 Anomalies detection for smart sensors

This first application is proposed as prior qualitative analysis of the anomalies detected by our solution and it has been carried on within the interface of the

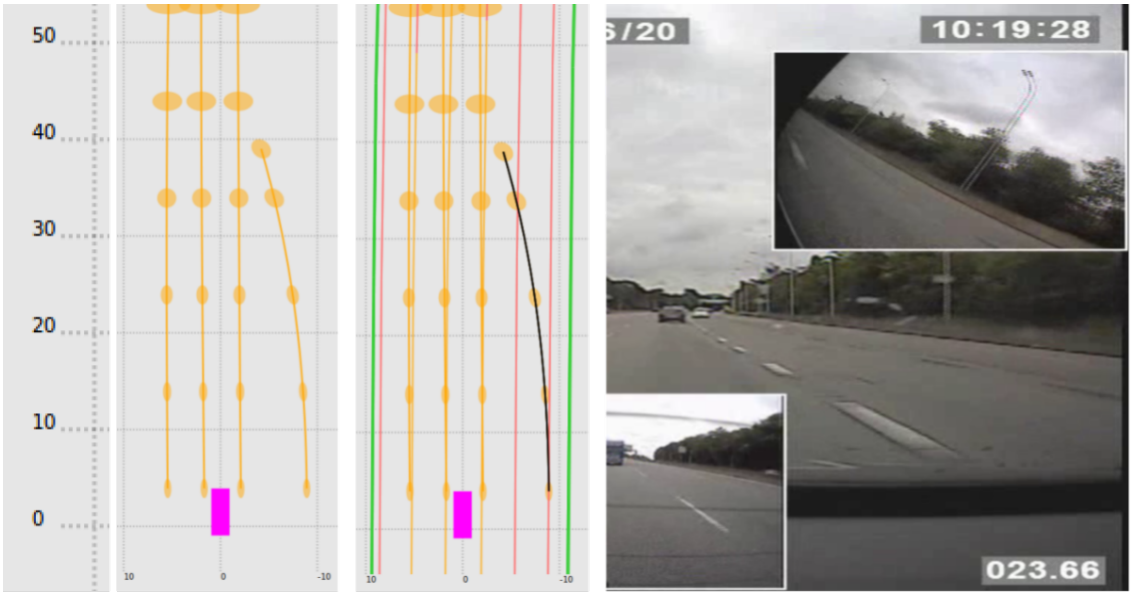


Figure 4.8: Left: orange FrontCam measurements are displayed in a bird-eye view. Center: map-provided lane boundaries (red and green lines) are associated to the measurements with the exception of one (highlighted in black). Right: the context cameras confirm the false positive detection.

software development platform itself. Making use of the bird-eye view data visualization, we want to display the measurements and, specifically, highlight those that have not been successfully associated to any of the map-provided lane elements.

As a matter of fact, in Fig. 4.8, one of the measurements issued of the smart sensor is correctly classified as false detection and highlighted in black. This picture presents on left the delivery of the Smart FrontCam "as is", a set of polynomial measurements displayed in orange. In the middle, the map-provider delivery has been added (lane boundaries are shown in red or green, whereas they represent road barriers) and the outcome of the association algorithm pointed out. Finally, the perspective of the context camera, displayed on right the side, confirms that the wrongly detected lane marking does not exist while the other measurements are properly associated to the map.

In this case and in similar others, we managed to identify a potentially dangerous false detection. Filtering it out would positively benefit the overall performance of the automated vehicle. We consider this experimental application as a positive and qualitative evaluation of the previously introduced models.

4.4.3 Scoring for multi-hypothesis localization

The second application brings to a further analysis, which aims at a quantitative evaluation of the solution despite the fact that no ground truth is available for this kind of experimental applications. As the recorded data comes from highway drivings and the number of the surrounding lanes N_l is known through the

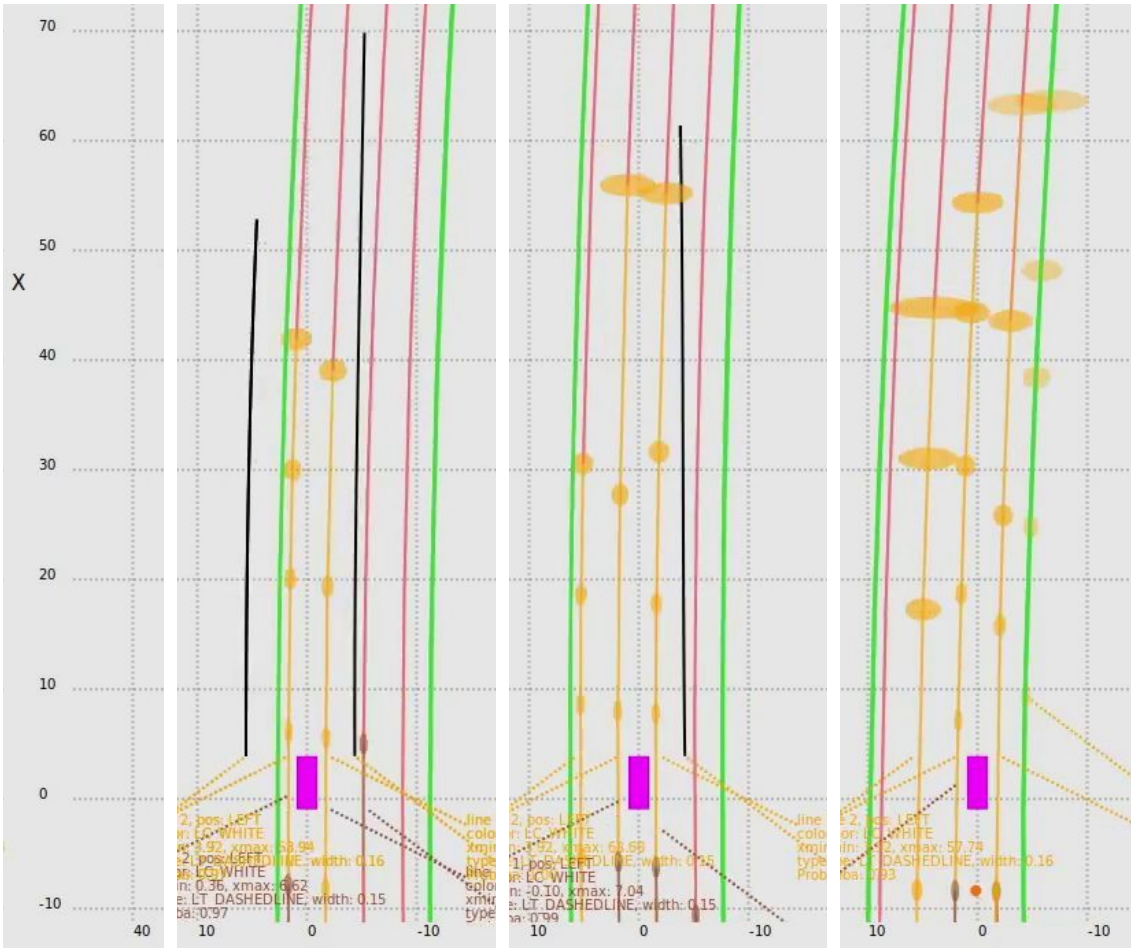


Figure 4.9: $N_l = 3$ variations of a same driving record. Respectively from the left: *Altered1*, *Altered2* and *Correct*. Highlighted in black, some of the measurements have not been successfully associated to any of the map-provided lane boundaries.

HD-map, we propose the following test and use case for our solution:

- The original driving record is labelled as *Correct*, considering that the localization system correctly positionate the vehicle (at least) within the correct lane.
- Based on *Correct*, $N_l - 1$ alterations of the driving record are generated, exploiting the HD-map to relocate the ego-vehicle within the other lanes. Context camera also confirms that these datasets are *Altered* hypothesis of localization.

Fig. 4.9 presents our specific case, where driving data recorded on a three-laned section of the highway provides $N_l = 3$ candidate hyphotheses of ego-localization.

Defining now a common evaluation metric, we can expect this indicator to be a discriminatory factor in distinguishing the *Correct* recording from the *Altered* ones. We chose the *Precision* metric, which is a meaningful index in those conditions where false positives are of strong impact on the application outcome,

as it is here. In this experiment, a false positive FP is a measurement issued of an on-board sensor which has not been successfully associated with the use of Algorithm 1 and a complete HD-map. True positives TP are therefore all the remaining received measurements. We compute $Precision(t-5,t)$ separately for each smart sensor over a time window of 5 seconds and it is defined as the following ratio:

$$Precision(t-5,t) = \frac{TP}{TP+FP} \tag{4.15}$$

Fig. 4.10 show the evolution of this indicator throughout our driving record of about two minutes, presented in three variations - one per drivable lane.

As would be expected from the setup, we are actually capable of differentiating *Correct* from *Altered1* and *Altered2* by making use of the *Precision* indicator. In terms of lane boundaries association, the computed *Precision* of the Smart FrontCam is crucial to make this distinction. On the other hand, all of the N_l recordings report similar trends from the standpoint of the Smart AVM. For this last sensor, we can suppose that not enough information is disposable in the delivered measurements to allow for the disambiguation of the altered datasets. Due to its limited field of view, it understandable that no contradiction is reported in terms of lane boundaries association regardless of the lane to be *Correct* or *Altered*. Table 4.1 ultimately presents the global $Precision(t_{start}, t_{end})$ indexes in the form of percentages, computed over the whole recorded data. This benchmark is based on a capsule (Section 2.4.3) of highway data. For 300 s of recorded sensor data while driving the prototype vehicle at medium-high speed, the travelled distance amounts to slightly less than 10 km.

	Smart FrontCam	Smart AVM
$Correct$	89.43%	99.07%
$Altered1$	57.31%	98.31%
$Altered2$	79.00%	98.60%

Table 4.1: *Precision* over complete driving

4.4.4 Tracking results

In case of positive association between lane boundaries in the map-track and sensor measurements, the Update step describes how the latest information is integrated in the estimation. In the following investigation, the tracking effects on the estimation of heading and curvature have been studied. A comparison between the state of measurement and track is carried on, separately computing the error for different position indicators of the measurements ($\in LL, L, R, RR$). Samples for this evaluation have been grouped for distinct interval of longitudinal distance. The Root Mean Square Error (RMSE) has been computed considering the difference between the state variable of interest and the reference.

Heading and curvature reference In our experimental setup, the identification of an appropriate reference to be deployed as ground truth is not trivial. A sup-

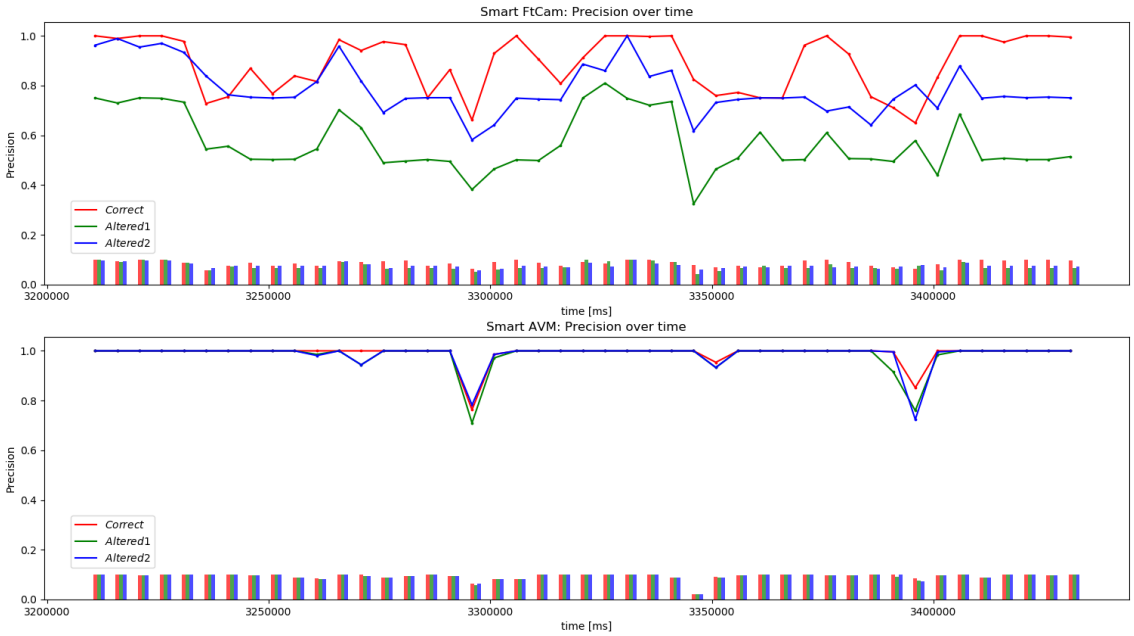


Figure 4.10: In the case of Smart FrontCam, *Precision* results generally higher for *Correct*, which makes possible its distinction from *Altered1* and *Altered2*. On the other hand, all of the N_l recordings report similar trends from the standpoint of the Smart AVM.

posedly perfect "oracle" would be unaffected by the estimation error that have been modelled in the smart sensor and map-provider deliveries. Hereinafter, the HD-map have been considered to be the most accurate representation of the lane boundaries available in the setup. This representation has been exploited for the computation of heading and curvature reference for each map node. Specifically, two hypotheses have been considered for the lane boundaries model and ground truth computation:

- **Circular arc**: at each map node, it has been supposed that the lane boundary passing through that node is a circular arc. Therefore, the equation of this arc is being computed exploiting additionally the two adjacent map points, generally the previous and the subsequent if available.
- **Clothoid spline**: at each map node, it has been supposed that the lane boundary passing through that node is a clothoid spline. The computation of this semi-parametric curve is not trivial and depends on the additional points considered for its estimation, other that of the computation method. In our case, the adopted method is presented in [Baran et al., 2010] and uses optimization for the regression of the clothoid spline.

Two distinct benchmarks are presented in the following, using the two different references mentioned above.

RMSE of heading and curvature w.r.t. HD-map reference As expected, the RMSE computed for the measure grows with the longitudinal distance. It can be observed in Fig. 4.11 and Fig. 4.12 that performing tracking of heading and curvature data on the map nodes positively affects the estimation of these variables, when the reference is based on the circular arc hypothesis.

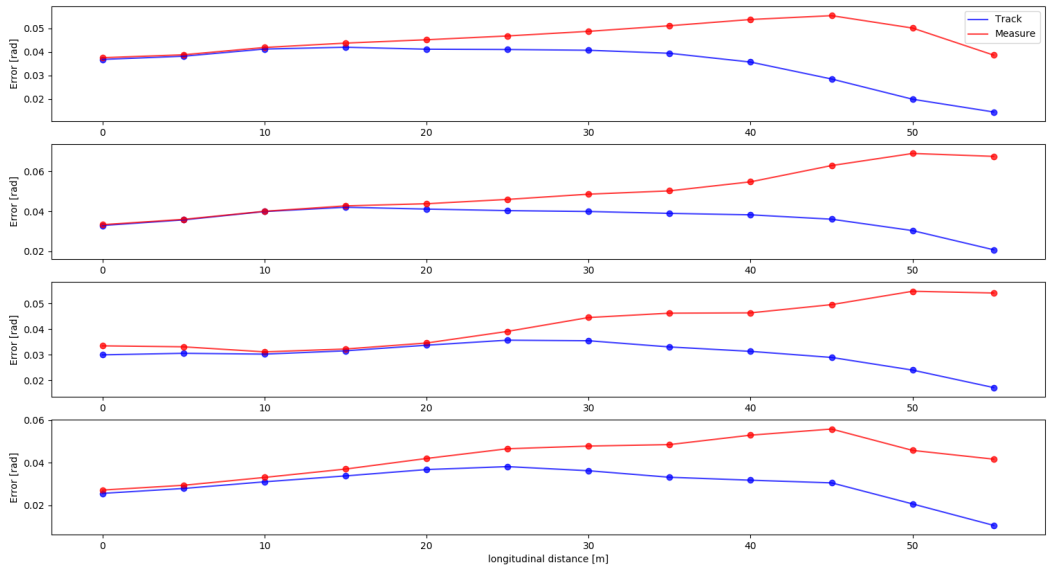


Figure 4.11: Heading error w.r.t. HD-map

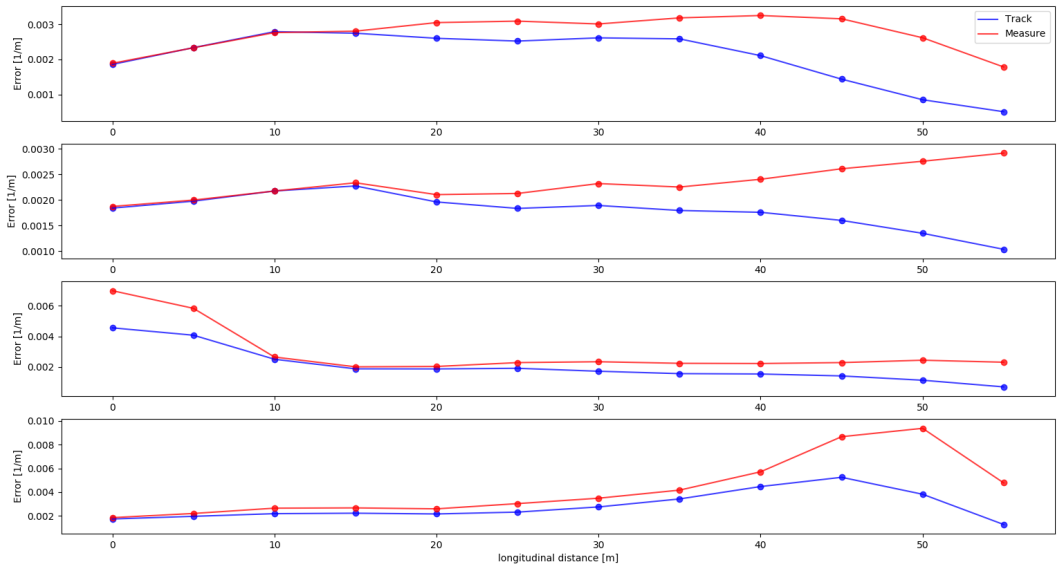


Figure 4.12: Curvature error w.r.t. HD-map

RMSE of heading and curvature w.r.t. Cornu-map reference Analogous results are obtained with the adoption of the clothoid spline reference, which could be considered to be a more accurate model of the road. The effectiveness of the im-

plemented lane boundaries tracking solution is confirmed in Fig. 4.13 and Fig. 4.14.

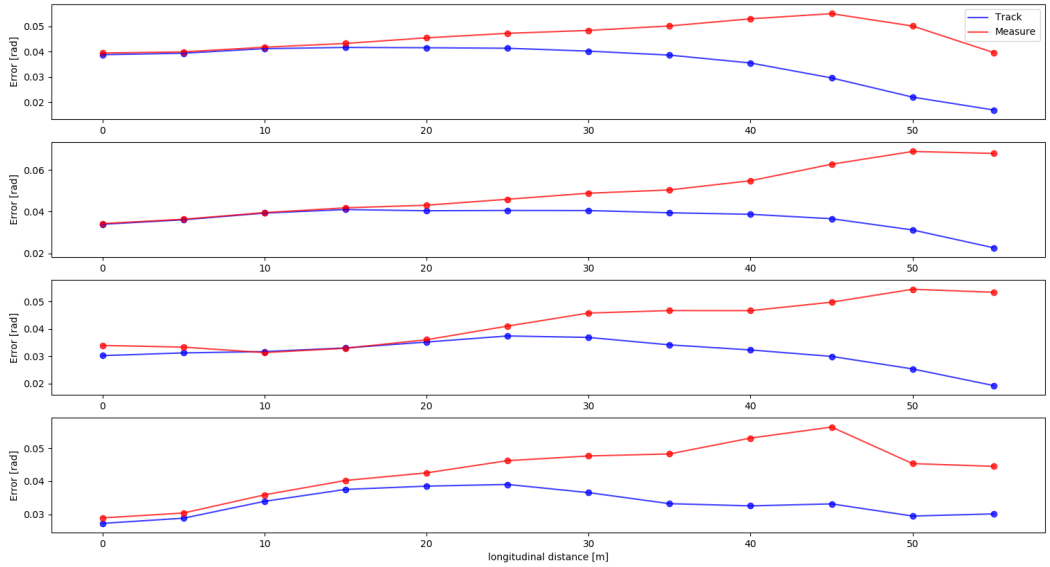


Figure 4.13: Heading error w.r.t. Cornu-map

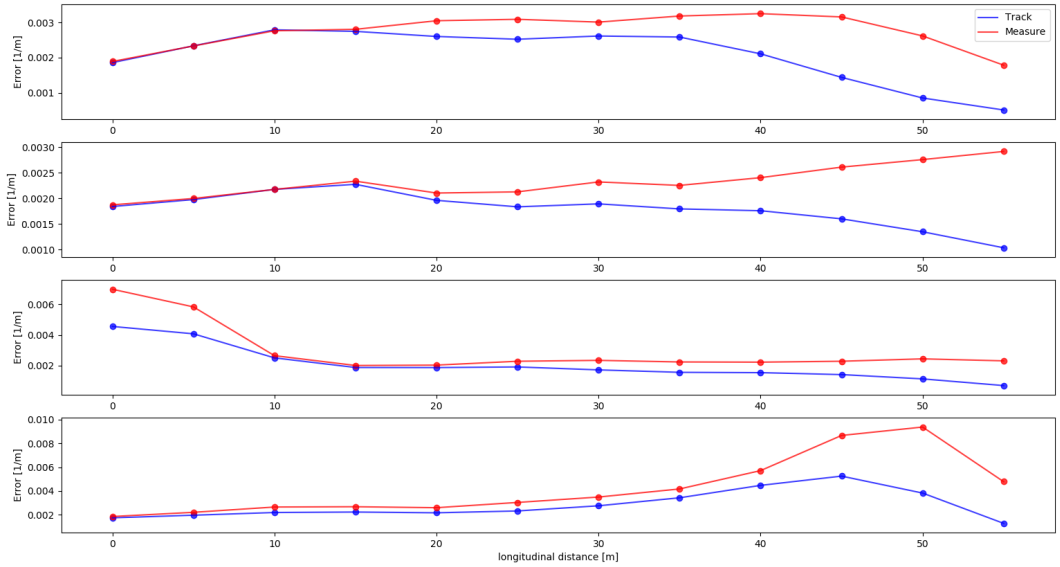


Figure 4.14: Curvature error w.r.t. Cornu-map

4.5 Conclusions

In this chapter, we presented our methodology and work in the task of lane boundaries estimation based on smart sensors and map-provider measurements. The information fusion solution takes input from the two different sources applying distinct models for representing the uncertainty in their deliveries. In this

instance, the model proposed for the map-provider takes into account of both localization and mapping uncertainties.

The results issued of experimental applications show that the use of lane boundaries association enables false positives detection for on-board smart sensors, whether the measurements sufficiently contradicts the content of the map. Specifically, applying our method to the Smart FrontCam resulted effective and correctly performed in the scoring of multiple hypotheses of ego-vehicle localization within the HD-map. Additionally, the tracking of sensor-issued heading and curvature information on the map nodes resulted effective with respect to two different references.

These experiments completed our work in regards of the lane boundaries estimation in presence of multiple information sources. In the rest of the manuscript, the general conclusions that can be drawn from this research are presented together with the immediate pespectives that would be interesting to explore when starting from what have been accomplished up to here.

Chapter 5

Conclusion

5.1 Concluding remarks

In this research, the task of lane boundaries detection in automated driving have been formulated over the SAE-L3 industrial use case and addressed with multiple information sources and data fusion methodology.

At first, the general context of ADAS and AD have been introduced to the reader together with some specificities from the automotive industry, such as the integration of smart sensors. The state of the art of automated driving pipelines and lane detection related methods have been subsequently presented, before formulating the problem as estimation of the surrounding lane boundaries. Two distinct solutions have been proposed and presented in the above and we present in following some considerations in their regards.

Feature-tracking In the first case, measurements from different smart sensors are being accumulated and fused in a set of global lane boundaries tracks. Our approach allowed for the adoption of a geometric lane model which is more complex than the polynomial curve implemented by off-the-shelf sensors. Furthermore, the information fusion method is independent from the nature of the input sensor and it performed well in a benchmark with respect to an HD-map lane-level reference. The deployment of this solution as a module of working architecture of L3 prototype vehicle confirmed the viability of the approach within real-time constraints and its compatibility with quasi-industrial implementation.

The advantages of using multiple sensor fusion in the lane boundaries estimation task are manifolds. The environment representation range can be expanded together with increased redundancy in the sensor set, implying better fault resistance. The overall safety is further enhanced adopting distinct technologies of perception where one can contribute in the shortcomings of the other. Lastly, the novel clothoid spline model allowed better representation of the road structure with curvature evolution that is more suitable for driving the vehicle in comfortable conditions for the passenger.

Map-tracking The first segment of the research work focused on the usage and fusion of smart sensors pointing out the possibility of faulty detections in the sen-

sor measurements. This scenario motivated the second part of this work where an additional source of a distinct nature have been included as support to the lane boundaries estimation task. Appropriate modelling of the involved uncertainties and inclusion of a map-provider in the system architecture allowed for experimenting map-aided approaches.

In these experiments, our lane boundaries association method has proven to be effective in the detection of erroneous measurements. This method has been additionally deployed in the computation of an indicator of coherence between map and on-board sensors, which can potentially detect faults in the localization system. Additionally, tracking of road heading and curvature on map waypoints resulted more effective that solely using the smart sensor measurements.

Limitations An assessment of the limitations of this research work can be done retrospectively, commenting on the main difficulties encountered:

- Disposing of ground truth at lane boundary level is technically challenging. There is no "oracle" given that even HD-Maps suffer from inaccuracies. In absence of a perfect reference, it has been supposed that one of the available information sources ranks to be better than the others (e.g. more accurate sensor has perfect calibration or post-processed localization is considered accurate).
- Working with smart sensors or other kind of integrated black boxes do not allow for the use of public datasets and the quantitative results are strictly sensor dependant.
- Uncertainty representation and estimation has a fundamental impact and it is not trivial. In our study, the Gaussian hypothesis have been supported by some of the experiments (Section 3.4.2) but still results in a strong hypothesis.

5.2 Perspectives

Given the work up to here and the comments on accomplished research and limitations, here are some of the most interesting perspectives which have been considered and partially explored.

Other vehicles as additional source The implemented fusion architecture is sufficiently general to be compatible with other possible environmental clues about the existence of lane boundaries. One might speculate on the presence of other vehicles in the roadway. In this use case, the following assumptions on the other vehicles in the driving scene are adopted:

- Vehicles drive at a constant and known lateral distance from a lane marker on their right side (plausible behaviour of an autonomous vehicle)
- Vehicles' pose is accurately estimated by the ego-vehicle on-board sensors

These are quiet strong hypotheses, but by relying on them we can suppose the presence of a road marking line beside each vehicle (also they are mostly equivalent to those in [Sakr et al., 2017]). To resist potential ghosts or noisy targets, a vehicle is traced only after a few seconds of track confirmation. A confidence index of the targets would interestingly contribute to this step. The traces left behind these other vehicles are processed as input measurements, in the form of clothoid splines. This experiment only consist of a proof of concept but, under the above-mentioned assumptions, can extend the lane detection range up to the object-detector's range (generally much higher).

In Fig. 5.1 (left and center), the cameras can detect lane markings up to 110 m (presented as before in orange) while exploiting the other vehicles the estimation reaches up to 150 m (vehicle's trail displayed in light and dark grey segments). Note also that the curve shape is reasonable because it is faithfully interpolated by the clothoid spline model.

Also in Fig. 5.1 (right), we present a case where the assumptions are not met. This situation would be quiet frequent in experimental recordings, where object detection is performed by on-board commercial sensors.
This additional test of "multiple-sources" estimation will be subject of further works, in more appropriate conditions or on a different set of assumptions.

Large scale testing on Fusion dataset Specific AD/ADAS features developed within the Renault framework are afterwards integrated in a class of upcoming commercialized vehicles. These vehicles can offer functions such as AEB and AD1 ("Autonomous Driving 1", a kind of enhanced ACC), implemented with suppliers sensors and in-house fusion algorithms. Before allowing for mass production, these solutions are being to be completed and tested. This validation requires for a certain level of performance to be guaranteed over a large number of kilometers. To achieve this, a fleet of prototype vehicles in currently recording sensor data on the most varies road of Europe (France, Netherlands and United Kingdom at the moment). This collection of data, which amouts of more than $60\,000$ km of driving recordings, is complemented with observations of events by the copilot (presence of roundabouts, weather conditions, vehicles cutting-in, etc.) and is consumed in this validation phase. A set of indicators corresponding to each demanded feature has been identified along with a performance target to be attained.

As a perspective for this research work, it would be of interest to evaluate the possible exploitation of this huge amount of data.

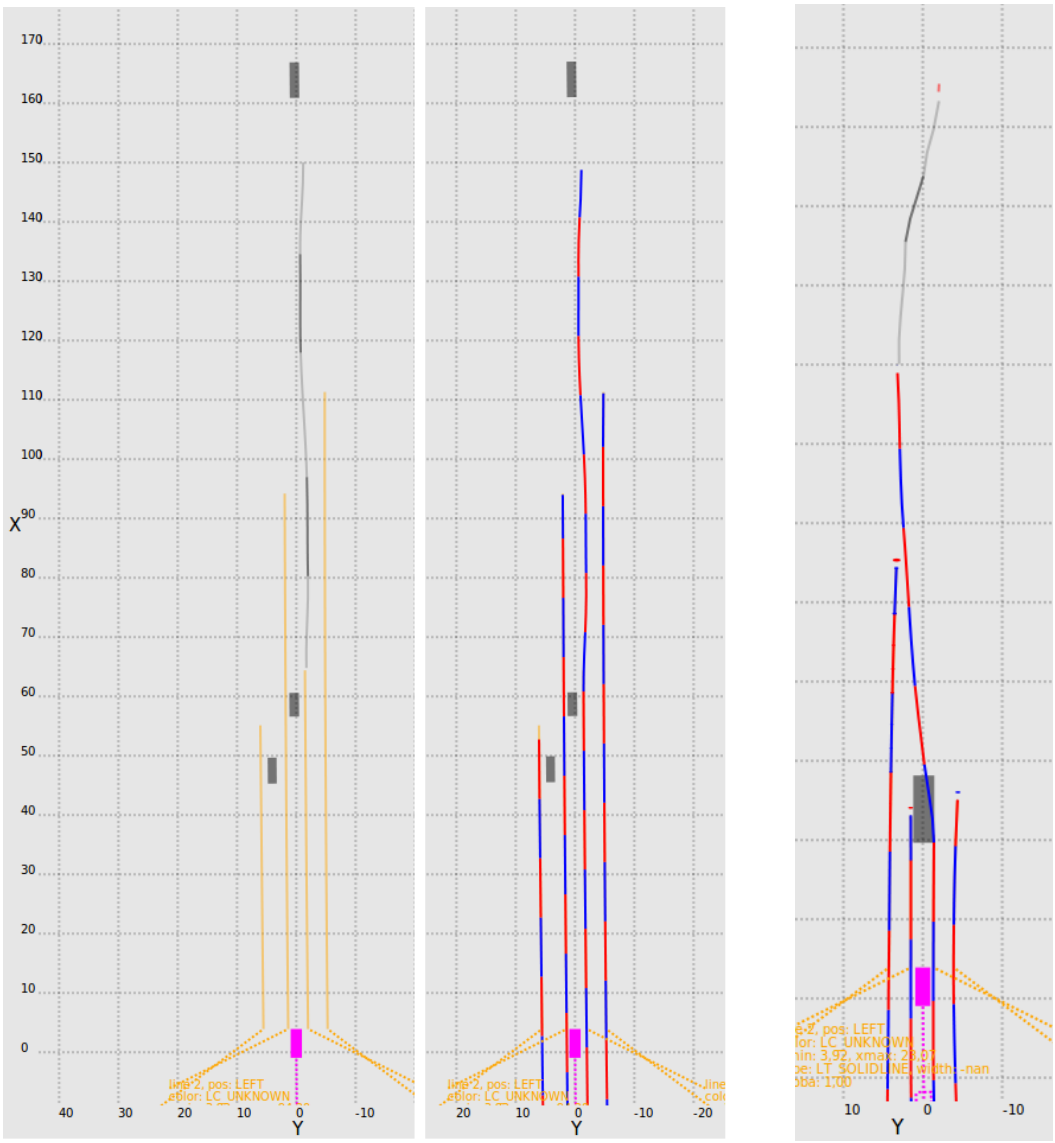


Figure 5.1: Left and center: both camera (orange) and other vehicles (light and dark grey trail) are exploited to track lane markings. Right: the front vehicle is not precisely tracked and leads to poor results.

Appendix A

Notes on the submitted patent application

This appendix details and illustrates the patent filed for application on the subject of the methodology presented in Chapter 4.

The invention consists of two main components. The first component is the lane boundaries association method applied to smart sensor maps and HD-maps. In this case, the association method is used to detect faulty measurements issued of the smart sensor. The second component leans on the first one to rank distinct hypothesis of localization of the vehicle within the HD-map.

The drafting of the technical memo for the preparation of the patent began in April 2021. The final text of the patent has been redacted in french language and consists of 31 pages. It has been completed and submitted to the Institut National de la Propriété Industrielle (INPI) in October 2021. The patent application has identifier n°2110938, hereinafter it is noted its full reference:

- **F. Camarda, B. Durand, F. Davoine, V. Cherfaoui**. *Procédé de détection d'une limite d'une voie de circulation*. Renault/CNRS patent. Applied for European patenting at Institut National de la Propriété Industrielle (INPI) under the identifier n°2110938, Oct 2021.

Some of the illustrations from the patent are presented in the following. Here is the description of the illustrations that follow:

- Fig. A.1a is a first schematic view from above of the vehicle on a road comprising two traffic lanes, on which are represented limits of a traffic lane as detected by a detection means on board the vehicle.
- Fig. A.1b is a second schematic top view of the vehicle, in which lane boundaries determined by means of map data are shown.
- Fig. A.1c is a third schematic top view of the vehicle, on which are represented, by road features modeling, the limits of the traffic lane as detected by the detection means on board of the vehicle.
- Fig. A.1d is a schematic top view of the vehicle on a road comprising five traffic lanes.

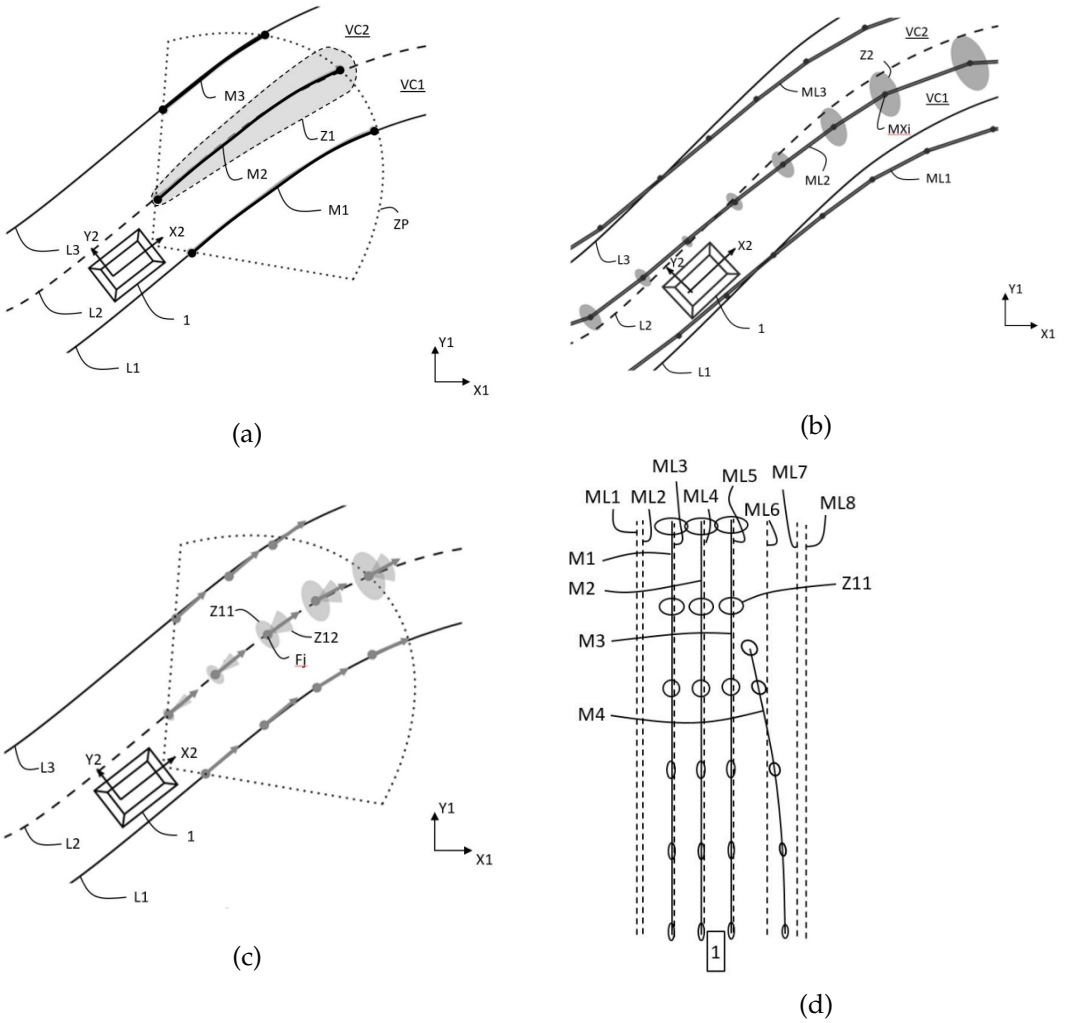


Figure A.1: Illustrations from the patent application

Appendix B

Global Nearest Neighbor algorithm implementation

The Global Nearest Neighbor (GNN) algorithm has a cost matrix xx as given input. In our case, the matrix contains distances between measurements (rows) and tracks (columns). Specifically, the cell $xx[i, j]$ contains the Mahalanobis distance between the measurement i and the track j. The aim of the algorithm is to find the best association between rows and columns that globally minimizes the association cost. The output is the solution column y to the Linear Assignment Problem (LAP) and indicates per each measure the index of the associate column, if any. In fact, it is possible for a measurement not to be associated to any of the tracks. Distances are saturated according to the gating variable $chi2$. The implementation of the GNN algorithm adopted in the ADAS development framework of our use case is the following:

```
/** GlobalNearestNeighbor function
*
* Computes y as the column solution of lap problem with cost matrix x
* If row i is associated to a column, then y(i) is the column number (i and y(i) starts at 0)
* If row i is not associated, y(i)=P_NT_LAP_NOTASSIGNED_VALUE
*
* y should have been allocated with a P_NT_LAP_MAX_DIM length
*
*/
uint32_t c_GlobalNearestNeighbor(float_t *xx,
                                 const uint32_t nRow,
                                 const uint32_t nCol,
                                 const float_t chi2,
                                 uint8_t *yy) {
    uint32_t status = S_OK;
    // check input size
    if ((nRow < 1u) || (nRow > P_NT_LAP_MAX_DIM) || (nCol < 1u) || (nCol > P_NT_LAP_MAX_DIM)) {
        status = S_FATAL_MATH_UNCOMPLIANT_MATRIX_DIM;
    } else {
        uint32_t ii, jj;
        // Reduce Column
        uint8_t skipCol[P_NT_LAP_MAX_DIM];
        for (ii = 0; ii < nCol; ii++) {
```

```
            skipCol[ii] = 1;
            for (jj = 0; jj < nRow; jj++) {
                if (xx[(jj * nCol) + ii] < chi2) {
                    skipCol[ii] = 0;
                    break;
                }
            }
        }
        jj = nCol;
        for (ii = 0; ii < nCol; ii++) {
            jj -= skipCol[ii];
        }
        uint32_t x1nCol = jj;
        // Reduce Row
        uint8_t skipRow[P_NT_LAP_MAX_DIM];
        for (jj = 0; jj < nRow; jj++) {
            skipRow[jj] = 1;
            for (ii = 0; ii < nCol; ii++) {
                if (xx[(jj * nCol) + ii] < chi2) {
                    skipRow[jj] = 0;
                    break;
                }
            }
        }
        jj = nRow;
        for (ii = 0; ii < nRow; ii++) {
            jj -= skipRow[ii];
        }
        uint32_t x1nRow = jj;
        if ((0u != x1nCol) && (0u != x1nRow)) {
            float_t *x1 = xx;
            uint8_t revCol[P_NT_LAP_MAX_DIM];
            uint8_t colsol[P_NT_LAP_MAX_DIM];
            float_t const *pX  = xx;
            float_t *pX1 = x1;
            for (uint32_t kk = 0; kk < nRow; kk++) {
                if (0u != skipRow[kk]) {
                    pX += nCol;
                    continue;
                }
                for (ii = 0; ii < nCol; ii++) {
                    float_t fx = *pX;
                    pX++;
                    if (0u != skipCol[ii]) {
                        continue;
                    }
                    *pX1 = fx;
                    pX1++;
                }
            }
            memset(revCol, 0, MAX((size_t)nRow, (size_t)nCol));
            jj = 0;
            for (ii = 0; ii < nCol; ii++) {
                if (0u != skipCol[ii]) {
                    continue;
                }
                revCol[jj] = (uint8_t)ii;
                jj++;
            }
```

```
83              uint8_t *y1 = skipCol;
84              FUSCOMCALL(status, c_lap(x1nRow, x1nCol, x1, y1, colsol));
85              jj = 0;
86              for (ii = 0u; ii < nRow; ii++) {
87                  if (0u != skipRow[ii]) {
88                      yy[ii] = P_NT_LAP_NOTASSIGNED_VALUE;
89                      continue;
90                  }
91                  if (y1[jj] != P_NT_LAP_NOTASSIGNED_VALUE) {
92                      if (x1[(jj * x1nCol) + y1[jj]] < chi2) {
93                          yy[ii] = revCol[y1[jj]];
94                      } else {
95                          yy[ii] = P_NT_LAP_NOTASSIGNED_VALUE;
96                      }
97                  } else {
98                      yy[ii] = P_NT_LAP_NOTASSIGNED_VALUE;
99                  }
100                 jj++;
101             }
102         } else {
103             for (ii = 0; ii < nRow; ii++) {
104                 yy[ii] = P_NT_LAP_NOTASSIGNED_VALUE;
105             }
106         }  // if (jj)
107     }
108     return status;
109 }
```

Bibliography

[Abramov et al., 2016] Abramov, A., Bayer, C., Heller, C., and Loy, C. (2016). Multi-lane perception using feature fusion based on GraphSLAM. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3108–3115.

[ADASIS, 2002] ADASIS (2002). Available at `https://adasis.org/`.

[Anderson et al., 2014] Anderson, J. M., Nidhi, K., Stanley, K. D., Sorensen, P., Samaras, C., and Oluwatola, O. A. (2014). Autonomous vehicle technology: A guide for policymakers. Rand Corporation.

[Arteris, 2018] Arteris, I. (2018). Available at `https://www.arteris.com/`.

[Bai et al., 2019] Bai, M., Mattyus, G., Homayounfar, N., Wang, S., Lakshmikanth, S. K., and Urtasun, R. (2019). Deep Multi-Sensor Lane Detection. arXiv:1905.01555 [cs]. arXiv: 1905.01555.

[Bailey and Durrant-Whyte, 2006] Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. IEEE robotics & automation magazine, 13(3):108–117.

[Bansal et al., 2018] Bansal, M., Krizhevsky, A., and Ogale, A. (2018). Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079.

[Baran et al., 2010] Baran, I., Lehtinen, J., and Popović, J. (2010). Sketching clothoid splines using shortest paths. In Computer Graphics Forum, volume 29, pages 655–664. Wiley Online Library.

[Bengler et al., 2014] Bengler, K., Dietmayer, K., Farber, B., Maurer, M., Stiller, C., and Winner, H. (2014). Three decades of driver assistance systems: Review and future perspectives. Intelligent Transportation Systems Magazine, IEEE, 6:6–22.

[Bertolazzi and Frego, 2015] Bertolazzi, E. and Frego, M. (2015). G1 fitting with clothoids. Mathematical Methods in the Applied Sciences, 38(5):881–897.

[Bertolazzi and Frego, 2018] Bertolazzi, E. and Frego, M. (2018). Interpolating clothoid splines with curvature continuity. Mathematical Methods in the Applied Sciences, 41(4):1723–1737.

[Bertozzi and Broggi, 1998] Bertozzi, M. and Broggi, A. (1998). GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Transactions on Image Processing, 7(1):62–81.

[Besl and McKay, 1992] Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In Sensor fusion IV: control paradigms and data structures, volume 1611, pages 586–606. International Society for Optics and Photonics.

[Bhonsle, 2016] Bhonsle, D. (2016). Development of an Automation Test Setup for Navigation Data Processing. PhD thesis.

[Bojarski et al., 2016] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

[Buehler et al., 2009] Buehler, M., Iagnemma, K., and Singh, S. (2009). The DARPA urban challenge: autonomous vehicles in city traffic, volume 56. springer.

[Carballo et al., 2020] Carballo, A., Lambert, J., Monrroy, A., Wong, D., Narksri, P., Kitsukawa, Y., Takeuchi, E., Kato, S., and Takeda, K. (2020). Libre: The multiple 3d lidar dataset. In 2020 IEEE Intelligent Vehicles Symposium (IV), pages 1094–1101.

[Chen et al., 2015] Chen, Z., Zou, H., Jiang, H., Zhu, Q., Soh, Y. C., and Xie, L. (2015). Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization. Sensors, 15(1):715–732.

[Chung et al., 2001] Chung, H., Ojeda, L., and Borenstein, J. (2001). Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope. IEEE Transactions on Robotics and Automation, 17(1):80–84.

[Consortium et al., 2004] Consortium, E. et al. (2004). Enhanced digital mapping project. final report. In EDMap Project eSafety Forum. Digital Maps Working Group Final Report. European Commission (eSafety Forum), Brussels.

[Derome, 2019] Derome, M. (2019). Comparison table of perception sensors. Technical report, Renault.

[Duraisamy et al., 2013] Duraisamy, B., Schwarz, T., and Wöhler, C. (2013). Track level fusion algorithms for automotive safety applications. In 2013 International Conference on Signal Processing, Image Processing & Pattern Recognition, pages 179–184. IEEE.

[Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. IEEE robotics & automation magazine, 13(2):99–110.

[Fatemi et al., 2014] Fatemi, M., Hammarstrand, L., Svensson, L., and García-Fernández, Á. F. (2014). Road geometry estimation using a precise clothoid

road model and observations of moving vehicles. In 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pages 238–244. IEEE.

[Ghallabi et al., 2018] Ghallabi, F., Nashashibi, F., El-Haj-Shhade, G., and Mittet, M.-A. (2018). LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map. In ITSC-2018 - 21st IEEE International Conference on Intelligent Transportation Systems, Maui, Hawaii, United States.

[Hasberg and Hensel, 2008] Hasberg, C. and Hensel, S. (2008). Online-estimation of road map elements using spline curves. In 2008 11th International Conference on Information Fusion, pages 1–7.

[He et al., 2016] He, B., Ai, R., Yan, Y., and Lang, X. (2016). Lane marking detection based on Convolution Neural Network from point clouds. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pages 2475–2480.

[HERE, 2017] HERE (2017). Hd maps for autonomous driving and driver assistance | here. Available at `https://www.here.com/platform/automotive-services/hd-maps/`.

[Houenou et al., 2012] Houenou, A., Bonnifait, P., Cherfaoui, V., and Boissou, J.-F. (2012). A track-to-track association method for automotive perception systems. In 2012 IEEE Intelligent Vehicles Symposium, pages 704–710. IEEE.

[Huang and Teller, 2011] Huang, A. S. and Teller, S. (2011). Probabilistic lane estimation for autonomous driving using basis curves. Autonomous Robots, 31(2-3):269–283.

[ISO, 2018] ISO (2018). Road vehicles – Functional safety. ISO 26262-1:2018, International Organization for Standardization, Geneva, Switzerland.

[Kammel and Pitzer, 2008] Kammel, S. and Pitzer, B. (2008). Lidar-based lane marker detection and mapping. In 2008 IEEE Intelligent Vehicles Symposium, pages 1137–1142, Eindhoven, Netherlands. IEEE.

[Kim and Song, 2016] Kim, T. and Song, B. (2016). Detection and Tracking of Road Barrier Based on Radar and Vision Sensor Fusion. Journal of Sensors, 2016:1–8.

[Kong et al., 2009] Kong, H., Audibert, J.-Y., and Ponce, J. (2009). Vanishing point detection for road detection. In 2009 ieee conference on computer vision and pattern recognition, pages 96–103. IEEE.

[Lee et al., 2021] Lee, M., Lee, J., Lee, D., Kim, W., Hwang, S., and Lee, S. (2021). Robust lane detection via expanded self attention. arXiv preprint arXiv:2102.07037.

[Leite, 2018] Leite, J. (2018). A brief history of gps in-car navigation - ndrive. Available at `https://ndrive.com/brief-history-gps-car-navigation/`.

[Lengyel and Szalay, 2019] Lengyel, H. and Szalay, Z. (2019). The Significance and Effect of the Traffic System Signaling to the Environment, Present and Future Traffic, pages 847–856.

[Li et al., 2018] Li, F., Bonnifait, P., and Ibañez-Guzmán, J. (2018). Map-aided dead-reckoning with lane-level maps and integrity monitoring. IEEE Transactions on Intelligent Vehicles, 3(1):81–91.

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

[Liu et al., 2019] Liu, R., Wang, J., and Zhang, B. (2019). High definition map for automated driving: Overview and analysis. Journal of Navigation, 73:1–18.

[Longuet-Higgins, 1981] Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. Nature, 293(5828):133–135.

[Maddern et al., 2015] Maddern, W., Pascoe, G., and Newman, P. (2015). Leveraging experience for large-scale lidar localisation in changing cities. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 1684–1691. IEEE.

[Marinelli et al., 2017] Marinelli, G., Bassani, M., Piras, M., and Lingua, A. (2017). Mobile mapping systems and spatial data collection strategies assessment in the identification of horizontal alignment of highways. Transportation Research Part C: Emerging Technologies, 79:257–273.

[MATLAB Sensor Fusion and Tracking Toolbox, 2021] MATLAB Sensor Fusion and Tracking Toolbox (2021). Matlab sensor fusion and tracking toolbox. The MathWorks, Natick, MA, USA.

[Michelet-Gignoux, 2019] Michelet-Gignoux, A. (2019). Mobileye camera characterization. Technical report, Renault.

[MIRA, 2008] MIRA (2008). MISRA-c:2004: guidelines for the use of the c language in critical systems. OCLC: 261374047.

[Mobileye, 2017] Mobileye (2017). Mobileye rem™ - road experience management. Available at `https://www.mobileye.com/our-technology/rem/`.

[Newcomb, 2013] Newcomb (2013). From hand-cranked maps to the cloud: Charting the history of in-car navigation. Available at `https://www.wired.com/2013/04/history-in-car-navigation/`.

[Obradovic et al., 2007] Obradovic, D., Lenz, H., and Schupfner, M. (2007). Fusion of sensor data in siemens car navigation system. IEEE Transactions on Vehicular Technology, 56(1):43–50.

[ONISR, 2020] ONISR, L. (2020). Sécurité routière en france. bilan de l'année 2004. Observatoire National Interministériel de Sécurité Routière.

[Pan et al., 2018] Pan, X., Shi, J., Luo, P., Wang, X., and Tang, X. (2018). Spatial as deep: Spatial cnn for traffic scene understanding. In Thirty-Second AAAI Conference on Artificial Intelligence.

[Park et al., 1997] Park, K., Chung, H., Choi, J., and Lee, J. G. (1997). Dead reckoning navigation for an autonomous mobile robot using a differential encoder and a gyroscope. In 1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97, pages 441–446.

[Poggenhans et al., 2018a] Poggenhans, F., Pauls, J.-H., Janosovits, J., Orf, S., Naumann, M., Kuhnt, F., and Mayr, M. (2018a). Lanelet2: A high-definition map framework for the future of automated driving. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pages 1672–1679. IEEE.

[Poggenhans et al., 2018b] Poggenhans, F., Salscheider, N. O., and Stiller, C. (2018b). Precise Localization in High-Definition Road Maps for Urban Regions. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2167–2174, Madrid. IEEE.

[Pomerleau, 1989] Pomerleau, D. A. (1989). ALVINN: An Autonomous Land Vehicle in a Neural Network. In Touretzky, D. S., editor, Advances in Neural Information Processing Systems 1, pages 305–313. Morgan-Kaufmann.

[Rahmathullah et al., 2017] Rahmathullah, A. S., García-Fernández, Á. F., and Svensson, L. (2017). Generalized optimal sub-pattern assignment metric. In 2017 20th International Conference on Information Fusion (Fusion), pages 1–8. IEEE.

[Reid et al., 2019] Reid, T. G., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., and Pandey, G. (2019). Localization requirements for autonomous vehicles. arXiv preprint arXiv:1906.01061.

[Ress et al., 2008] Ress, C., Balzer, D., Bracht, A., Durekovic, S., and Löwenau, J. (2008). Adasis protocol for advanced in-vehicle applications. page 7.

[Ribas et al., 2008] Ribas, D., Ridao, P., Tardós, J. D., and Neira, J. (2008). Underwater slam in man-made structured environments. Journal of Field Robotics, 25(11-12):898–921.

[SAE International, 2021] SAE International (2021). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Available at `https://saemobilus.sae.org/content/j3016_202104/`.

[Sakr et al., 2017] Sakr, A. H., Bansal, G., Vladimerou, V., Kusano, K., and Johnson, M. (2017). V2V and on-board sensor fusion for road geometry estimation. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pages 1–8.

[Schuster et al., 2016] Schuster, F., Keller, C. G., Rapp, M., Haueis, M., and Curio, C. (2016). Landmark based radar slam using graph optimization. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pages 2559–2564. IEEE.

[Smith and Cheeseman, 1986] Smith, R. C. and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. The international journal of Robotics Research, 5(4):56–68.

[Sun et al., 2020] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2446–2454.

[TomTom, 2019] TomTom (2019). Available at `https://www.tomtom.com/products/adas-map/`.

[TuSimple, 2021] TuSimple (2021). TuSimple/tusimple-benchmark. Available at `https://github.com/TuSimple/tusimple-benchmark/`.

[Urtasun, 2020] Urtasun, R. (2020). A future with affordable self-driving vehicles. Mathematical Challenges and Opportunities for Autonomous Vehicles 2020. Available at `https://youtu.be/V2JIxlwlIBY/`.

[Vo et al., 2015] Vo, A.-V., Truong-Hong, L., Laefer, D. F., and Bertolotto, M. (2015). Octree-based region growing for point cloud segmentation. ISPRS Journal of Photogrammetry and Remote Sensing, 104:88–100.

[Wang et al., 2017] Wang, S., Clark, R., Wen, H., and Trigoni, N. (2017). Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 2043–2050. IEEE.

[Welte et al., 2020] Welte, A., Xu, P., Bonnifait, P., and Zinoune, C. (2020). Improved Data Association Using Buffered Pose Adjustment for Map-Aided Localization. IEEE Robotics and Automation Letters, 5(4):6334–6341.

[Xu et al., 2020] Xu, H., Wang, S., Cai, X., Zhang, W., Liang, X., and Li, Z. (2020). Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16, pages 689–704. Springer.

[Yu et al., 2020] Yu, C., Cherfaoui, V., Bonnifait, P., and Yang, D.-g. (2020). Managing localization uncertainty to handle semantic lane information from georeferenced maps in evidential occupancy grids. Sensors, 20(2):352.

[Zeng et al., 2019] Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., and Urtasun, R. (2019). End-to-end interpretable neural motion planner. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8660–8669.

[Zhang et al., 2021] Zhang, G., Sugiarto, T., and Khayyer, P. (2021). Road-shoulder scanning using multi-sensor kalman filter for minimum risk maneuver. SAE WCX Digital Summit.

[Zhang and Singh, 2018] Zhang, J. and Singh, S. (2018). Laser–visual–inertial odometry and mapping with high robustness and low drift. Journal of Field Robotics, 35(8):1242–1264.

[Ziegler et al., 2014] Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C. G., et al. (2014). Making bertha drive—an autonomous journey on a historic route. IEEE Intelligent transportation systems magazine, 6(2):8–20.

[Zou et al., 2019] Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., and Wang, Q. (2019). Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks. arXiv:1903.02193 [cs]. arXiv: 1903.02193.