This full text paper was peer-reviewed at the direction of IEEE Instrumentation and Measurement Society prior to the acceptance and publication.

Challenges in aggregation of heterogeneous sensors for Autonomous Driving Systems

Jean-Pierre Giacalone
Renault SW Labs, Groupe Renault
Expert ADAS/AD sofware architecture
Autonomous Vehicle Algorithms
Sophia Antipolis, France
jean-pierre.giacalone@renault.com

Luc Bourgeois
Renault SA
Expert Leader ADAS/AD Systems
Guyancourt, France
luc.l.bourgeois@renault.com

Andrea Ancora
Renault SW Labs, Groupe Renault
Senior Member Technical Staff
Autonmous Vehicle Algorithms
Sophia Antipolis, France
andrea.ancora@renault.com

***Abstract*—This paper is highlighting the complexity associated with sensors aggregation in an autonomous driving system regarding aspects of sensing technologies, space coverage around ego vehicle, detection issues and software architecture. Building Advanced Driving Assistance systems (ADAS) and autonomous driving functions of Level 3 ([1]) and above mandates that these sources of complexity are properly addressed by novel approaches in dealing with sensors raw data and produced features. The paper tries to depict a structured view of the sources of complexity related to the driving scene that must be apprehended, then goes through situational examples for each theme and pitches key directions for improvement.**

Keywords*—*Autonomous Drive, multiple sensors, architecture

I. Introduction

An Autonomous Driving System can be summarized by four major execution stages: Perception, Decision, Control, Actuation. Such a system has strong safety requirements to fulfill its mission. As a result, every step above must exhibit the highest level of quality in performing its duty. Whatever energy, intelligence and enthusiasm engineers can put in constructing a clever (human-like?) Decision stage, if the Perception stage is not providing enough quality information, the overall driving experience would look very poor (lots of lateral movements and/or spurious slowdowns that can lead to nausea).

In this paper we will focus on the Perception stage to understand its constraints linked to the heterogeneity of sensors, linked to how the space around the autonomous vehicle is covered but, also, regarding overlapping areas, detection issues and software architecture implications to support a roadmap of sensing capabilities improvement. We will start by positioning clearly the complexity related to this Perception stage. Then, we will consider situational examples to help understand the current sensing limitations. We will detail how we believe the multi-sensing architecture should evolve, using a road capture case, and we will end with aspects related to sensing data structures normalization for efficient software components construction. Most of the ideas presented in this paper have been gathered through several studies done, for single lane as well as multi-lanes autonomous driving, in vehicle prototypes, some of them still operating on roads, and have helped gaining confidence in the Perception architecture required to deploy a truly Autonomous Driving experience in our cars.

II. Heterogeneity of sensors

Moving forward, in this paper, we will concentrate on the sensing components of the Perception and their organization up to the contribution to the Fusion function. By Fusion function it is meant, here, the step that combines distinctive features reported by sensors to deliver, as much as possible, a set of unique features with a high degree of confidence to the other stages mentioned above (see Figure 1.0).

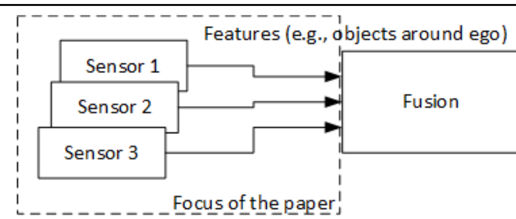

Fig 1.0 Sensors and Fusion relationship

A. *Complementary Technologies assembled in Perception*

In the figure 2.0, below, we highlight a typical driving situation on a highway. It comprises several agents located around the autonomous vehicle (at the bottom), circulating along a certain road geometry (shape, number of lanes). It shows classical properties that illustrate the sensing challenges: ranges of distances to deal with, occlusions, multiplicity of target shapes.

Sensors have been studied in various surveys like [2] and are their sensing capabilities are permanently in evolution due to the new trend for driving autonomy. We would like, here, to give an overview of sensing aspects of importance for the rest of this paper. Not a single sensor can cope with this diversity of properties. Cameras can cope with shapes, longitudinal distance accuracy and occluded situations by proper detection tuning but over a limited absolute range. Radars, on the contrary, operate on a technology that delivers accurate measurements of speeds and distances over a wider range but that could lead to misleading features extractions in the context of occlusions as depicted in the figure. Lidars use another technology (laser) that maps well 3D structures through point clouds to accurately aggregate shapes (provided there are enough laser beams that hit on these shapes), leading to good longitudinal and lateral accuracy as well as free space size. Finally, short range sensing like provided by sonars allow to accurately map free space around the autonomous vehicle (also named Ego, in this paper). In the farther distances, traditional sensing lose capability to report anything due to lack of line of sight in very occluded areas. In this case, what is called Electronic Horizon is required to bridge the line-of-sight gap. It is to be noted that the diversity in sensing accuracy just discussed for agents around Ego is also valid for sensing the road infrastructure (general shape, lanes, landmarks).

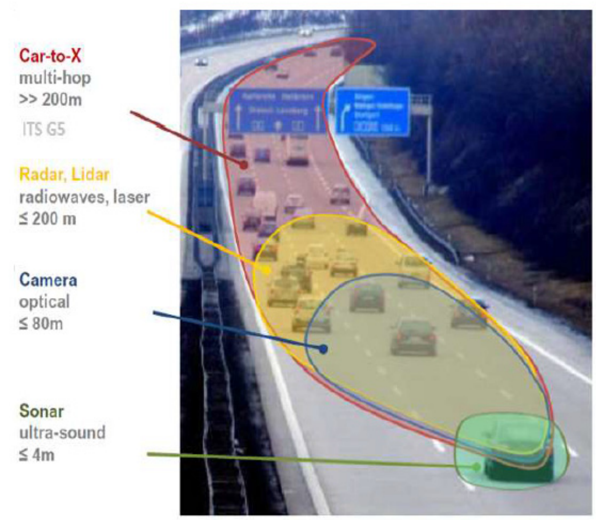

Fig 2.0 Sensors types and ranges

B. *Electronic Horizon as an additional sensing source*

The notion of Electronic Horizon comprises usage of indirect sources of information (as for instance depicted in [3]). Typically, those sources come through following means:

- High-definition road map technologies associated to roadbooks of landmarks,
- Radio transmission technologies (Car-2-X, V2X) between agents or between infrastructure and agents.

The first source usually pre-exists, stored into Ego, like a traditional navigation map, as it is uniquely dependent on the infrastructure. But it may also have a dynamic component that allows local updates (like for road works indications). The second source is dynamic in nature where information is provided through messages delivered over the radio technology. This also allows to report about agent's behavior as detected by mobiles equipped with the technology in emission mode. Those reports satisfy a relationship between latency (as the time difference between the occurrence of the event and its reception into Ego vehicle) and confidence on the occurrence by the general equation (1) on probability p

$$p(latency < t_{max} \mid confidence > c_{min}) > threshold \qquad (1)$$

This equation indicates that short latency information must be correlated inside Ego from multiple other sources of that type via a fusion stage as described above. Information that would be provided in anticipation (i.e. further ahead, say, a solid traffic jam 800 meters in front of Ego) with longer latency but stronger confidence could be used more directly by the Decision and Control stages. Hence, the Electronic Horizon would essentially provide means for anticipation to Decision and Control stage while helping classical sensor fusion with quite accurate parameters (e.g. geometry of lanes coming from the high -definition map).

III. A Layered Representation of the World

What was described in the previous section leads to a representation of the world around Ego (mobile agents, infrastructure) constructed by layers. Each layer provides its set of details of that world with an acceptable confidence. The difficulty resides in the fact that the space coverage around Ego may not be fully continuous for all layers because of various constraints coming from implementation and cost. Figure 3.0 gives examples of implementation constraints in a vehicle that need to cope with aesthetics as well as sensing efficiency.

As a result, each sensing layer must deal with overlap in space of similar sensors. Then, over all layers, there must be enough coverage of critical areas around Ego. For safety reasons, it is considered that two to three layers must simultaneously cover those critical regions. We are going to review different implications of the layered structure and constraints added to the quality provided by Perception.

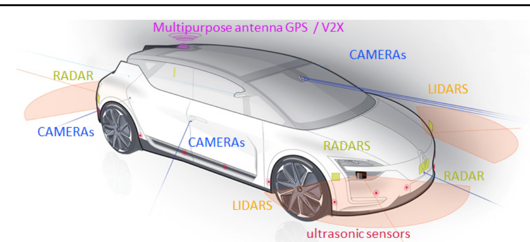

Fig 3.0 Constraints in sensors location

A. *The benefits of a layered sensors structure*

The layered analysis of the World provides key benefits that can be summarized in these categories:

- Redundancy, that contributes to safety.
- Decision/Control validations.
- Low latency paths for Decisions.

Fig. 4.0 illustrates the aspects above. As indicated in the first section, features returned by sensors contribute into the Fusion stage. Failure of one sensor in a layer with other layers still contributing in the corresponding field of view around Ego allows meaningful indications to continue flowing to Decision and Control Stages. This satisfies a safety requirement for the Autonomous Driving system.

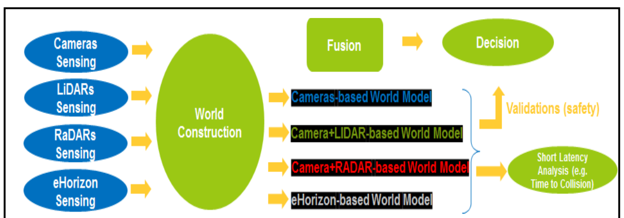

Fig 4.0 Layered view of the World and applications

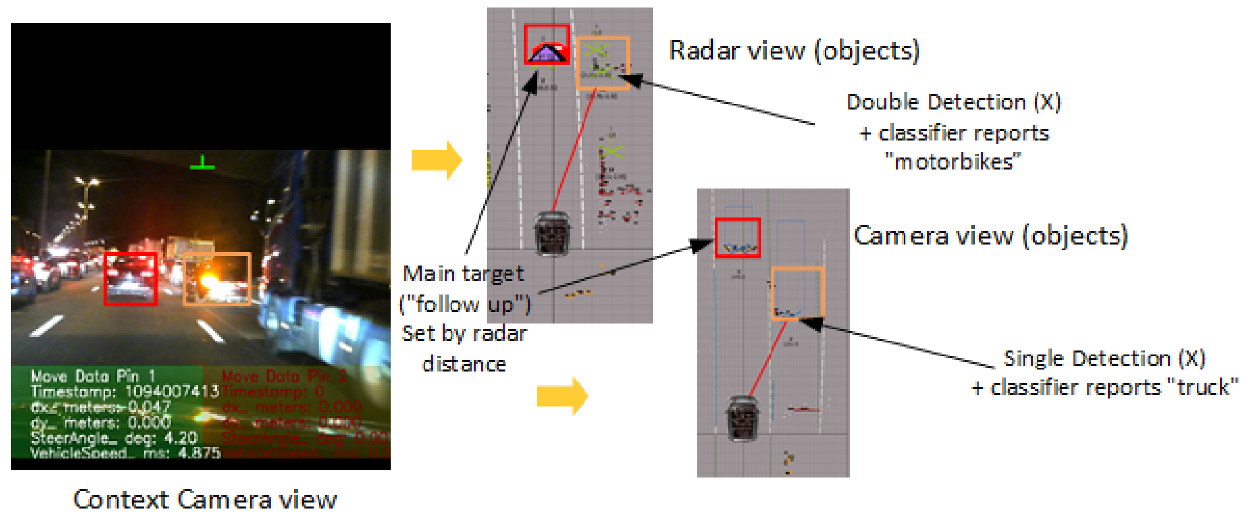

Fig 5.0 Examples of reports (objects – two cars, red and orange) from front radar and camera in a dense traffic situation

Moreover, the Electronic Horizon sensing source is delivering an accurate set of infrastructure features all around Ego that greatly help its positioning accuracy. Similarly, when the Decision stage has constructed a trajectory to be followed by Ego, from fused information, it must be validated for any risk of collision along the way. Having different views of the World around Ego allows to check these risks in different sensing domains (as, for instance, proposed by [4]). This also satisfies a safety requirement of the system. Finally, having layers that provide specific parameters with high accuracy (e.g. lidars with position information) allows to define short paths to decision and control for specific emergency actions like sudden obstacle avoidance.

B. Sensors Complementary behavior illustration

In this part of the paper, we would like to illustrate by mean of a road situation capture, the interest of overlapping multi-layered sensing as well as showing its complexity.

- In Fig 5.0, an image capture is shown, on the left, of the scene ahead of Ego vehicle (heavy traffic, by night). On the right, two data reports are synthesized to indicate what is sensed by the front camera and the front radar. In the lane where Ego drives, the situation is clear and the primary object of attention (in red) is well positioned (in the lane, at the correct distance, by both sensors). The Fusion step can then perform its duty, with such level of information from overlapping sensors. For the car highlighted in orange and regarding its position, objects returned are also reasonably well aligned so that the Fusion could also get useful data out of the information. From the classification standpoint, though, none of the sensors are giving the right answer (a truck for the camera – note that the car is surrounded by trucks in its lane – and two motorbikes for the radar). This is an issue that we will try to tackle in section C.
- In examples like Fig 5.0, we tend to observe that overlapping sensors of different kind complement each other but that overlapping sensors from the same type (e.g. front and corner radars) tend to generate detection ghosts that will potentially mislead the Fusion stage by generating extra tracks to address. To avoid this, it is usually proposed to generate objects from the combined raw signals from overlapping sensors of the same kind. This generates an extra complexity in merging and detecting features but significantly improves the quality of the Fusion stage for agents detections.

C. The need for raw sensor fusion

The examples above have illustrated that, for sensors of the same technology that overlap in space coverage around Ego, it is advantageous from detection quality to start from combined raw signals from that set of overlapping sensors. This is usually called raw sensors fusion. It is a well-known approach in image processing, for instance. In park assist applications, it is now common to use surround cameras to reconstruct a bird eye view of the situation around the car to ease automatic parking trajectory construction. This has also been applied to detect nearby objects with the same sensors ([5], [6]). We would like to further extend this notion of raw sensor fusion to groups of complementary sensors. One of the main objectives of detecting objects with positional, size and other important information is to determine their trajectory over time so that they would eventually be important targets to track for the safety of Ego along its path. To do this, objects delivered by sensors must be tracked before being used to populate states into a motion model (e.g. a Kalman filter). Usually, that tracking is easy if the form of noise on trajectory parameters of an object, as delivered by a sensor, is known or, ideally, low. It would be the case if objects trajectories are sufficiently away from each other. On the contrary, when objects become close in space and time, then the level of uncertainty associated with that vicinity must be better removed to avoid misleading the track to motion association. Fig. 6.0 shows an example of two objects traveling along their trajectories that would lead them to cross paths. When they are far away, the tracker assigns them clear IDs (labeled “0” and “1” – arrows indicate direction of motion, i.e. time increase). But when they come close, the tracker starts to lose these track IDs and begins to assign new ones (“2”, in this case) and, moreover, starts to populate from the new IDs to the trajectory models. This is a quite frequent problem in densely populated driving scenes like the one above.

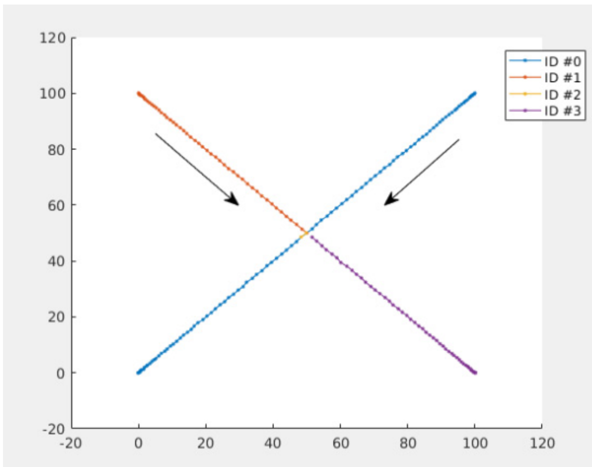

Fig 6.0 Track assignment confusion in close space/time

A solution to this problem is to combine, at sensor-level, some characteristics of objects that would help disambiguate the situation (i.e., the uncertainty) associated with close vicinity in space and time. For instance, adding to the accurate position information delivered by the lidar an accurate indication of color or shape from the camera. And inject these additional characteristics in the motion model. Hence, ID#0 being a sedan, white car and ID#1, being a compact, grey car, it would be easier to assign their IDs even when they get close in space and time.

IV. Software Architecture Implications

In this section we would like to go over some important requirements regarding the software structure that supports the aggregation of multiple sensors.

A. Standardization of interfaces

It is important to understand that, with the number of sensors that are implied in the aggregation described in this paper, it is a requirement that each layer provide features (objects, lanes, etc.) in a normalized way (see Figure 7.0). This is obvious from both the design of the Perception function as well as the construction of the Fusion stage standpoint. And it goes by defining data structure details that support proper descriptions of features. In terms of software implementation in the automotive domain, the AUTOSAR model is now well established to provide efficient implementation and execution of software functions into Electronic Control Units found in modern cars. Sensing functions as reviewed in this paper are usually implemented within ECUs. Recently, the AUTOSAR Consortium has started a normalization initiative for sensors, trying to define common data to be delivered to the rest of the system, and it is gaining traction in the industry. As an example of this, we have been normalizing, in an entire multi-lanes Autonomous Driving system, objects delivered and organized by layers in the form of the following package (and similarly, also, for lanes and other features):

SensorObject structure, containing

- ➔ MoveState, enumerated data structure,
- ➔ ObjBlinkerInfo, enumerated data structure,
- ➔ ObjClass, enumerated data structure,
- ➔ ObjFuncQualifier, enumerated data structure,
- ➔ ObjRefPoint, enumerated data structure,
- ➔ SensorSource, enumerated data structure,
- ➔ ObjCutinInfo, data structure,
- ➔ SensorObjlist, data structure,
- ➔ SensorObject data structure.

In this system, all sensor reporting objects provide results that are grabbed by the autonomous driving software and above data are populated. For instance, all objects have a class being reported in ObjClass, within the enumeration “car”, “pedestrian”, “motorbike”, “truck”, etc. Similarly, in MoveState fields, all objects reported by sensors in the system have the cars motion model ($x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}$) parameters stored. Another field of interest is ObjCutinInfo that carries detections, by those sensors that are capable, of objects intensions (like following a cutting-in trajectory, as studied in [7]). Having these structures properly populated by the software grabbing stage allows Fusion and other functions to be able to work on known elements, independently of their source and prepares for implementation of raw sensing fusion like depicted in the previous section.

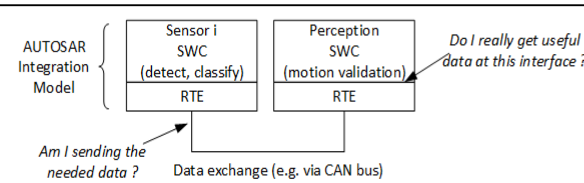

Fig 7.0 Illustrating the need for standard interfaces

B. Software Components organization

Another important aspect of the Perception software architecture is to organize its software components so they can allow efficient parallelization of their tasks on various hardware platforms organizations. Furthermore, those components must be organized so that bridges can be easily established between sensors to support features like raw fusion. This implies that functions inside of the components are organized into services that can be allocated to certain steps of the processing chain. An example of such service is the tracking function that we depicted in the previous section. The benefit of this software construction approach lies in the upfront validation and characterization of key kernels on target hardware platforms so that, later, they can be reliably integrated within the full perception software. This guarantees we can meet expected system performance and favors component re-use to make the overall perception system assembly more reliable.

V. CONCLUSIONS

In this article we tried to highlight the complexity inherently associated to handling a multi-sensing system required to support an Autonomous Driving System. We have depicted the challenges in aggregating sensors features so that we can fulfill safety requirements (redundancy, multi-views of the world surrounding Ego vehicle), improving confidence on features being provided to the Fusion stage (ghosts removal, better qualification for more efficient fusion). Finally, we have given indications of important trends in the construction of the software structure that supports the execution of the Perception function.

ACKNOWLEDGMENT

We would like to acknowledge our colleagues from Renault SW labs and from Renault Technocentre that contributed through reviews and discussions to the ideas browsed in this article.

REFERENCES

[1] SAE International. "Automated Driving Levels of Driving Automation are Defined in New SAE Inetrnational Standard J3016. " http://www.sae.org/autodrive (2014).

[2] Josef Steinbaeck, Christian Steger, Gerald Holweg, Norbert Druml. “Next generation radar sensors in automotive sensor fusion systems.” 2017 Sensor Data Fusion: Trends, Solutions, Applications (October 2017).

[3] Yuki Horita, Ramon S. Schwartz, “Extended electronic horizon for automated driving”, 2015 14th International Conference on ITS Telecommunications (Dec 2015),

[4] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. "On a Formal Model of Safe and Scalable Self-driving Cars. " arXiv preprint arXiv:1708.06374 (October 2018).

[5] Baek, Iljoo, Albert Davies, and Geng Yan. "Real-time Detection, Tracking, and Classification of Moving and Stationary Objects using Multiple Fisheye Images." arXiv preprint arXiv:1803.06077 (March 2018).

[6] Baek, JeongYeol, et al. "Scene Understanding Networks for Autonomous Driving based on Around View Monitoring System." arXiv preprint arXiv:1805.07029 (May 2018).

[7] Il-Hwan Kim, Jae-Hwan Bong, Jooyoung Park, and Shinsuk Park. “Prediction of Driver’s Intention of Lane Change by Augmenting Sensor Information Using Machine Learning Techniques.” Sensors 2017, 17, 1350; doi:10.3390/s17061350, MDPI (June 2017).