Lot6 : Infrastructure intelligente
L6.1-6.2 : Rapport sur les systèmes de communication I2V et les algorithmes de traitement

Programme	FUI23
Référence	6.1 - 6.2
Version	2.0
Date	23 / 02 / 2021
Porteur	Renault
Auteur(s)	S.-S. Ieng, B. Cabon
Contributeurs(s)	S. Masi, Ph. Bonnifait, Ph. Xu, S. Ambellouis

Financé par

bpifrance îledeFrance La Région Occitanie L'EUROPE S'ENGAGE en région Auvergne-Rhône-Alpes UNION EUROPÉENNE La Région Auvergne-Rhône-Alpes Yvelines Le Département

Pôles de labellisation

aerospace valley mov'eo Imagine mobility ViaMéca

Contents

1 Introduction

Connected and autonomous vehicles (CAV) are equipped with numerous sensors and therefore capable today of detecting obstacles and understanding a scene thanks in particular to advanced machine learning and the increasing computing power. However, in many situations, automated vehicles alone are not able to detect and to predict upcoming difficulties, especially when obstacles ares hidden in complex area like junctions, roundabouts or L-turn. In these situations, it is interesting to be able to improve and extend the perception ability of CAV by developing the collaboration between vehicles, other connected users and infrastructure. Roadside sensors can provide valuable information to CAV to predict difficulties and to update the vehicles trajectories planning when unconnected vehicles and vulnerable road users are close.

In order to make collaboration possible between different users, V2X communication systems are being developed and since the last decade, a great number of research activities have been conducted on Cooperative ITS(C-ITS) to improve road safety and traffic efficiency. The deployment of V2X communication systems and large-scale test projects have been launched across Europe (Scoop@F, C-Roads, InDiD, PAC V2X, MAVEN) but also in the United States, Japan [20] or China. Furthermore Standardization organizations such as the European Telecommunication Standards Institute (ETSI) and Car-2-Car Communication Consortium (C2C-CC) now provide specifications of V2X communications protocols and C-ITS services using both connected vehicles and roadside units (RSU) as interacting agent to implement these services. The connected vehicles are able to exchange information about their states such as the position or the velocity or what they locally perceive through collective awareness messages (CAM) or collective perception messages (CPM) and by gathering all these collaborative information, the connected vehicles can construct a Local Dynamic Map (LDM) to have an extended view of their environment.

Tornado project aims to develop a secured autonomous vehicles system by including intelligent roadside perception that is capable of broadcasting in real time local information to the CAVs in low density and peri-urban areas. Within the Tornado project, we are developing the intelligent roadside perception that is used in sparse area where, generally, the roads are narrow and the visibility conditions are poor. For this project, the experiments are carried out in partnership and with the support of *Rambouillet Territoire* and its Mobility Living Lab in the Bel-Air business area. Some important use cases are identified for this area: a narrow passage under the bridge crossing, the roundabout crossing and pedestrians and autonomous shuttle interaction situation.

The outline of this report is the following:

- The Tornado project for autonomous vehicles on sparsely populated and peri-urban area,
- The cooperative driving and ETSI standards
- The camera based perception system for roundabout crossing and pedestrians and autonomous shuttle interaction situation
- The Narrow zone passage

2 Tornado autonomous vehicles on sparsely populated and peri-urban area project

Many experimentation initiatives and projects on connected and autonomous vehicles (CAV) are being conducted in France and all around the world. However, Tornado project differs from other identified so far by proposing for the first time to conduct experiments on CAV that promote the multimodal mobility in sparsely populated, rural and peri-urban areas and to rely on the community of users from the living lab of the Rambouillet territory. The infrastructure in these areas are usually too basic and too poorly maintained to be used by a CAV without any improvement of the road. In the project, it is planned to test

1. CAVs at normal speed (up to 70km/h) connecting the Gazeran train station to the Bel-Air shopping area,
2. shuttles in the shopping center's car park where the shuttle and shopping customers share the same space.

For the CAVs testing, both Renault and the Université Technologie de Compiègne (UTC) have developed their driving control algorithms and their prototype based on the ZOE cars. During the route from the Gazeran train station to the Bel-Air shopping area, the CAVs have to cross narrow rural roads, and some roundabouts where the visibility conditions are very poor. To assist the CAVs to cross the unsignalized intersections, we developed within the project the connected roadside perception systems that broadcast real-time information to the CAVs. Another use case studied in the Tornado project is to improve the mobility of the customers of the shopping area, who left their vehicle in the Gazeran train station car park by using autonomous shuttle operated in a shared space or living street. The Autonomous shuttles will meet pedestrians with or without shopping trolleys, cyclists, cars or delivery trucks along their route. By broadcasting the information on the perceived road users, their categories, and their occupied spaces, the RSU will enhance the shuttles security by extending the perceived area of the shuttles.

The figure 1 shows the roundabout selected for the experimentation and the figure 2 shows the shared space near the drive-through of the Carrefour supermarket

Figure 1: The roundabout selected for the Tornado project's experimentation.

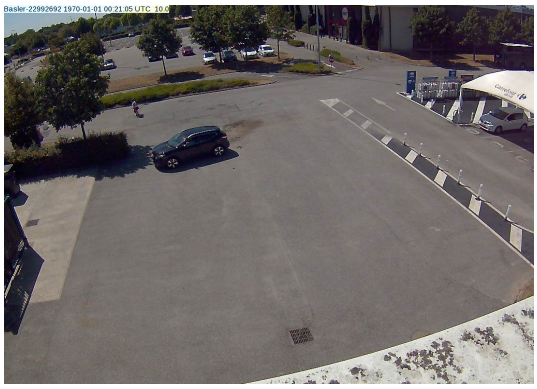


Figure 2: Part of the shared space used by the autonomous shuttle.

3 ETSI Standards for Vehicles Communication

In this document we provide a detailed explanation of the main standards used to achieve vehicular communication in the autonomous vehicles domain. In particular, we focus the attention on the European standard ETSI for vehicle communication. During the document, we detailed the main types of messages contained in this standard and we also provide a brief comparison with the American SAE standard. Moreover, we discuss in detail the format and the contents of the cooperative perception message (CPM) providing an evaluation about the relevance of its different fields to our I2V communication use cases.

3.1 Cooperative driving

Cooperation in autonomous driving is an emerging technology that is looking at to enhancing the performance of autonomous vehicles by means of cooperation in Intelligent Transportation Systems (ITS). In order to implement cooperation, it is required to share information between road users. Vehicular communication is one of the most important means to share such information among several users. Moreover, the communication can be implemented not only between road users, but also between road users and Road Side Units (RSU) in both directions. The aim of this section is to present an overview of the existing standards used in vehicular communications. Furthermore, we analyze also the application of such communication standards to ensure the exchange of information with an intelligent infrastructure in order to ensure safe navigation. In particular, we focus on the necessary information and its format that RSUs and self-driving cars have to exchange to achieve enhanced perception.

There exists in the literature a wide range of scenarios that take advantage of inter-vehicle communication. All these cases can be grouped into three principal subsets:

- **Safety-oriented**: These applications aim to ensure driving safety. This imposes many constraints on the communication standards as for example real time constraints, low message latency and low messages loss rates. In general, the aim of these applications is to make cars exchange data about their current status, their perceived environment and the notification of a perturbing event. Exchanging such information between cars in a road network can provide to single cars a global and enhanced view of the surrounding environment that a driver in general cannot have. Some examples of that are the possibility of seeing cars in bad weather conditions, the advertising of traffic jams and car accidents on the road, enhancing a vehicle perception on cluttered environments and blind spots and the advertising of the presence of an incoming emergency vehicle.

- **Traffic-Control Oriented**: These applications are not related to safety. However, exchanging information about the vehicle's positions can help in individuating the zones with traffic congestion giving a global view about the traffic. This can be used to regulate the traffic in function of the traffic jams. A typical use case is a smart traffic light manager that collects information about the queues of cars that are waiting to pass and regulates the passage minimizing the waiting time.

- **User-Comfort Oriented**: These applications are oriented at providing services that a user can enjoy while he is driving on the car. Some examples are the possibility of downloading movies or music during a trip. All these cases have the basic requirement of having access to the internet.

3.2 ETSI Standard

In Europe, the first experiments to achieve vehicular communication started around 1980. Before that date, several projects had been launched to achieve cooperation among communicating devices. The standards to achieve these tasks have been developed by the European Standardization Union (ESOs), the European Telecommunication Standards Institute (ETSI) and the Comité Européen de Normalisation (CEN). This standardization covers all types of transportation systems and also infrastructure based systems as the tolling systems. The standardization is driven by the Car-2-Car Communication Consortium (C2C-CC), which is an industry consortium of automobile manufacturers that signed an agreement to introduce the standard in Europe since 2015.

The ETSI ITS standard is based on the concept of ITS station. An ITS station can be a connected vehicle, a person with a connected device, or a communicating roadside unit. In the United States, another standard called WAVE has been developed. Figure 3 shows the architecture of the ETSI standard compared with the WAVE architecture. The access technologies layer primarily utilizes a specific set of options of the IEEE 802.11 standard, that is, ITS-G5 (where G5 stands for 5 GHz). In the United States, this set is named Wireless Access in Vehicular Environment (WAVE), formerly referred to as the IEEE 802.11p amendment and now integrated into the IEEE 802.11-2012 standard release. The European variant, ITS-G5, is derived from WAVE and adapted to European requirements.

On the top of the network and transport layer, there are the standards for application-oriented vehicular communication. Among these facilities, there are two services for cooperative communication. The CAM protocol conveys critical vehicle state information in support of safety and traffic efficiency applications. This is useful because receiving vehicles can track other vehicles positions and movements. The DENM protocol disseminates event-driven safety information in

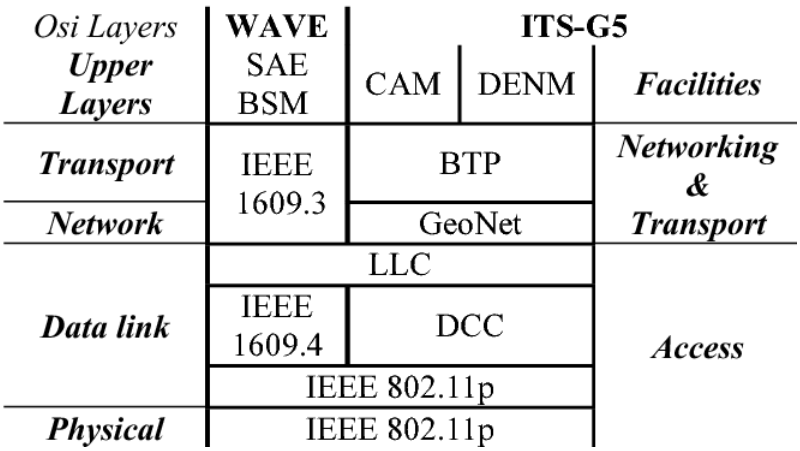


Figure 3: Architecture of ITS ETSI standard compared with the WAVE architecture. Figure from [2]

a geographical region.

Finally, the new Cooperative Perception Message (CPM) protocol (which is not present in Figure 3 because it has been recently introduced) allows vehicles to exchange information about their perceived environment. This is useful to enhance on board sensors' fields of view and to obtain an extended and more consistent representation of the driving environment. It will be presented below.

From Figure 3, one can see that the upper layer facilities are implemented by the CAM and DENM (and CPM) protocols for the ITS-G5 standard, while for the WAVE standard there are the BSM and SAE. Regarding the lowest layers, we can say that the standards are very similar in both versions.

Let us briefly show the contents of the SAE standard. We do not provide details about the lower level architecture of the system, because we are interested in analyzing and comparing only the services provided by higher level application levels. In particular, we are interested in the SAE J2735 standard that specifies the dictionary for the Base Safety Message (BSM) which is used to achieve the main tasks about navigation safety. Figure 4 shows the messages list available on SAE J2735 standard, while Figure 5 displays the contents of a BSM. It is easy to see that there exist similarities between the European and the American Standards. A brief explanation of the most important messages reported in Figure 4 is given hereafter:

- **Basic Safety message**: a message that is constantly exchanged between the neighboring vehicles to inform all the other ITS about the presence of a vehicle.

J2735 Defined Messages

ID	Messages	Typical Use	Status
0	Reserved	N/A	
1	MSG_A_la_Carte	V2X	
2	MSG_BasicSafetyMessage (BSM)	V2V	Used by USDOT program & other ITS industry research
3	MSG_CommonSafetyRequest	V2?	
4	MSG_EmergencyVehicleAlert		
5	MSG_IntersectionCollisionAvoidance	V2X	
6	MSG_MapData	I2V	Based on USDOT/CAMP CICAS-V project. Used by various demo/research program
7	MSG_NMEA_Corrections	I2V	
8	MSG_ProbeDataManagement	I2V	Used by VII Proof of Concept (PoC) project
9	MSG_ProbeVehicleData	V2I	Used by VII PoC project
10	MSG_RoadSideAlert		
11	MSG_RTCM_Corrections	I2V	Based on USDOT/CAMP CICAS-V project. Used by various demo/research program
12	MSG_SignalPhaseAndTiming	I2V	Based on USDOT/CAMP CICAS-V project. Used by various demo/research program
13	MSG_SignalRequestMessage	V2I	
14	MSG_SignalStatusMessage	I2V	
15	MSG_TravelerInformation Message	I2V	Used by VII PoC & will be used in Model Deployment (Curve Speed Warning)

Figure 4: A list of different messages contained into the J2735 standard. Notice that several messages correspond to the same use cases of ETSI. Figure taken from [1].

- **A la carte message**: a message customizable by the user which allows flexibility in data representation.
- **Emergency vehicle alert message**: used to broadcast warnings to the surrounding vehicles of an emergency vehicle operating in the neighborhood.
- **Generic transfer message**: a basic means to exchange data between a vehicle and the roadside unit.
- **Common safety request message**: used when a vehicle that exchanges BSM needs to make specific requests to other vehicles for additional information about safety applications.

Data elements/frames		Description	Remarks
Part I	DSRCmsgID		
Part I: BSM Blob (Octet string)	MsgCnt		
	TemporatyID		
	DSecond		
	Latitude		
	Longitude		
	Elevation		
	PositionalAccuracy		
	TransmissionAndSpeed		
	AccelerationSet4Way		
	BrakeSystemStatus		
	VehicleSize		
Part II	SafetyExtension		Optional
	VehicleStatus		Optional

Figure 5: The payload of a BSM message. Figure taken from [1].

3.3 Cooperative Awareness Messages (CAM)

To provide cooperative awareness between several communicating road entities, a Cooperative Awareness Service has been implemented. The standard is defined into the ETSI EN 302 637-2 document [3]. This standard can be applied to every road vehicle (cars, motorbikes, truck, etc.. . .), to pedestrians and to roadside infrastructure units (intelligent traffic lights, barriers, tolls, etc.). To implement cooperative awareness, the information to be shared is exchanged by means of the Cooperative Awareness Message (CAM). To implement this, every road user has to exchange information about its status. On reception of a CAM, the receiving user becomes aware of the existence of the sending one and its current status. Several use cases for this service are present in literature, as it is explained in [2].

3.3.1 CAM Generation Frequency

To implement the cooperative awareness service, it is important to update periodically the status of the surrounding environment. For safety critical scenarios, the evolution of the driving situation is highly dynamic. To achieve this, the CAM standard proposes some constraints:

- The CAM generation interval should not be inferior to 100 ms (10 Hz).
- The CAM generation interval should not be superior to 1000 ms (1 Hz).

The upper bound limit of 10 Hz has been chosen to avoid network saturation and bottlenecks that can be caused by multiple sending of several road agents.

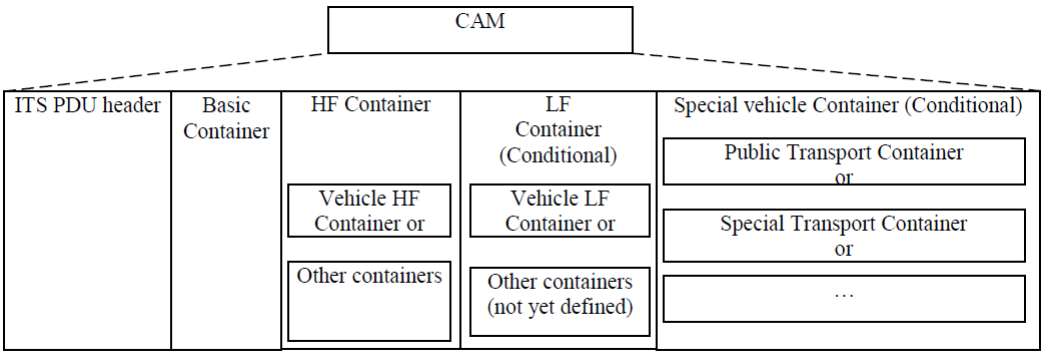


Figure 6: The format of an ETSI CAM message. Figure taken from [2]

3.3.2 CAM Messages Format

CAM is composed of a "ITS-PDU" header and several containers, which together constitutes a CAM. The ITS PDU header is a common header that includes the information of the protocol version, the message type and the user ID of the originating user. Figure 6 illustrates the structure of a CAM.

To successfully implement cooperative awareness, a CAM must include at least one basic container and one high frequency container. It can also have one low frequency container, and one or more special vehicle container:

- The basic container includes basic information related to the originating user.
- The high frequency container contains highly dynamic information of the originating user.
- The low frequency container contains static and not highly dynamic information of the originating user.
- The special vehicle container contains information specific to the vehicle role of the originating vehicle user.

3.3.3 CAM Payload Description

Let us see the information necessary to implement the CAM basic services. In this part, we also decided to assign a color to each field of the payload. Such colors are assigned to fields according to the criteria explained in Table 1. Colors are assigned to each field in order to specify the importance of every piece of information in the context of a safety-critical autonomous vehicle navigation application. The following table 1 explains the meanings of each color.

Red	High priority information	Used to highlight information that is necessary to develop an application for road safety.
Yellow	Middle priority information	Used to have some information that is not considered necessary to implement safety-oriented applications. However, it can be useful to have its knowledge.
Green	Low priority information	This information is not considered relevant for our use case. It can be eventually erased to make space for other more relevant fields.
Blue	High priority information	Used to highlight information that is necessary to develop an application for road safety. However, this information is, in general, hard to compute

Table 1: Classification criteria for messages fields w.r.t. our use-case application.

R	Timestamp	Time corresponding to the time of the reference position in the CAM, considered as time of the CAM generation.

3.3.3.1 CoopAwareness Container This container is the global container that also contains the information of the message generation time.

3.3.3.2 Basic Container This container provides basic information about the station. The following table explains in detail the contents of this container.

Importance	Field Name	Description
R	Station Type	Station type of the originating user.
R	Reference Position	Position and position accuracy measured at the reference point of the originating user. The measurement time shall correspond to generationDeltaTime. The positionConfidenceEllipse provides the accuracy of the measured position with the 95 % confidence level. Otherwise, the positionConfidenceEllipse shall be set to unavailable.

3.3.3.3 High Frequency Container This container provides highly dynamic information about a certain station. This information must be refreshed frequently

in order to keep the state updated. The following table illustrates the container's field in detail.

Importance	Field Name	Description
B	Heading	Heading and heading accuracy of the vehicle movement of the originating user with regards to the true north. The heading accuracy provided in the DE headingConfidence value shall provide the accuracy of the measured vehicle heading with a confidence level of 95 %. Otherwise, the value of the headingConfidence shall be set to unavailable.
R	Speed	Driving speed and speed accuracy of the originating user. The speed accuracy provided in the speedConfidence shall provide the accuracy of the speed value with a confidence level of 95 %. Otherwise, the speedConfidence shall be set to unavailable.
R	Driving Direction	Vehicle drive direction (forward or backward) of the originating user.
Y	Vehicle Length	This DF includes: • vehicleLengthValue: Vehicle length of the vehicle user that originates the CAM. If there are vehicle attachments like a trailer, or overhanging attachments like a crane, which extend the vehicle length to the front and/or rear; then the vehicleLengthValue shall provide the length for the vehicle including the attachments. • vehicleLengthConfidenceIndication: indication of whether the trailer is detected to be present and whether the length of the trailer is known.

Y	Vehicle Width	Vehicle width, measured of the vehicle user that originates the CAM, including side mirrors.
R	Longitudinal Acceleration	Vehicle longitudinal acceleration of the originating user in the center of the mass of the empty vehicle. It shall include the measured vehicle longitudinal acceleration and its accuracy value with the confidence level of 95 %. Otherwise, the longitudinalAccelerationConfidence shall be set to unavailable.
G	Curvature	This DF is related to the actual trajectory of the vehicle. It includes: • curvatureValue denoted as inverse of the vehicle current curve radius and the turning direction of the curve with regards to the driving direction of the vehicle • curvatureConfidence denoted as the accuracy of the provided curvatureValue for a confidence level of 95 %.
G	Curvature Calculation Mode	Flag indicating whether vehicle yaw rate is used in the calculation of the curvature of the vehicle user that originates the CAM.
R	Lane Position	The DE lanePosition of the referencePosition of a vehicle, counted from the outside border of the road, in the direction of the traffic flow. This DE shall be present if the data is available at the originating user. This concept can be computed in a curvilinear framework if a map is available.

B	Steering Wheel angle	This DF includes the steering wheel angle and accuracy as measured at the vehicle user that originates the CAM. It consists of the following DEs: • steeringWheelAngleValue denotes steering wheel angle as measured at the vehicle user that originates the CAM. • steeringWheelAngleConfidence denotes the accuracy of the measured steeringWheelAngleValue for a confidence level of 95 %. Otherwise, the value of steeringWheelAngleValue shall be set to unavailable.
G	Lateral Acceleration	Vehicle lateral acceleration of the originating user in the center of the mass of the empty vehicle. It shall include the measured vehicle lateral acceleration and its accuracy value with the confidence level of 95 %. This DE shall be present if the data is available at the originating user.
G	Vertical Acceleration	Vertical Acceleration of the originating user in the center of the mass of the empty vehicle. This DE shall be present if the data is available at the originating user.
G	Performance Class	The DE performanceClass characterizes the maximum age of the CAM data elements with regard to the generationDeltaTime
G	CenDRSCTollingZone	

G	Yaw Rate	This DF includes: • yawRateValue denotes the vehicle rotation around the center of mass of the empty vehicle. The leading sign denotes the direction of rotation. The value is negative if the motion is clockwise when viewing from the top. • yawRateConfidence denotes the accuracy for the 95 % confidence level for the measured yawRateValue. Otherwise, the value of yawRateConfidence shall be set to unavailable.

3.3.3.4 Low Frequency Container This container provides low dynamic information about a certain station

Importance	Field Name	Description
G	Vehicle Role	The role of the vehicle user that originates the CAM. The setting rules for this value are out of the scope of the present document.
G	Vehicle Light	Status of the most important exterior light switches of the vehicle user that originates the CAM.
G	Path History	This DF represents the vehicle's recent movement over some past time and/or distance. It consists of a list of path points, each represented as DF PathPoint. The list of path points may consist of up to 23 elements.

3.3.3.5 Special Container This container must be filled with another container according to the type of special vehicle and its characteristics we want to describe (e.g. if it is a police car, it is communicated whether the car is on an emergency status or not). This part is not considered relevant to our study and so it is not

described in detail.

3.4 Cooperative Perception Message (CPM)

The main goal of a Cooperative Perception (CP) basis service is to share with other road users the environment perceived by a vehicle with its own sensors. The perceived environment representation can be refined, fused, processed and classified before the broadcast. Final results are stored into CP objects and broadcast to other road users. Moreover, some quality indexes can be added to perceived objects, in order to quantify the information reliability and consistency.

A CP object contains an aggregated and interpreted abstract information perceived by sensors about other road participants and obstacles. Typically objects are represented in a mathematical formalism i.e. a set of variables describing characteristics as their dynamics, their geometry and several other aspects.

In this part, we are also interested in analyzing which information is relevant to exchange perception between several road users. In particular, we focus our attention to safety-critical use cases and we investigate the required level of information to proficiently ensure safety. Obviously, a trade off between detailed information and payload size of the message exists. Such constraint implies that we must investigate the essential information to be shared among road users in order to both provide a compact and complete environment representation and to avoid sending redundant information.

Finally, the object representation is sent to other surrounding vehicles exploiting the V2X communication. With this new received knowledge, other users can enhance their environment representation and complete their knowledge of the ongoing road scenario with information that is not directly accessible.

Some examples where this service can improve the performance are the filling of blind spots and cluttered environments. In our specific case, we exploit a remote intelligent infrastructure to provide the autonomous vehicle additional sources of information via I2V communication. For this reason, in the following part, we investigate the contents of a CPM message to select information that is necessary to broadcast to ensure safe navigation in a roundabout.

3.4.1 Cooperative Perception versus Cooperative Awareness

Cooperative Perception is the concept of sharing perceived environment of a road user to others. This perception is based on information obtained from sensors. The main difference between cooperative perception (CP) and the cooperative awareness (CA) is that, in the first case, the broadcast information is about the vehicle current environment, rather than about the vehicle current status.

However, it is mandatory to include in a CP basic service information about the sending vehicle in order to reference the perceived objects in other vehicles frames.

3.4.2 Messages Transmission and Generation

The road user is supposed to send a CPM whenever it detects an object with a sufficient level of confidence. It is possible not to send a detected object because the confidence level on the detection is low. However, even if the object is rejected, the user should send a CPM (at least empty, if no objects are reputed to have good confidence) at minimum sending frequency. The empty container must have inside it the information about the sending vehicle. This can help other road users in knowing the following things:

- A road user is present on the scenario
- A potential additional source of information is present on the scenario

Transmission rate of a CPM is computed according to the following criterion:

- Broadcast information should be as detailed as possible and provided as frequently as possible
- The utilization of the channel should be minimized

According to this, the CP basic services define the limits of the interval between two consecutive CPMs (and the corresponding sending frequencies) as follows:

- $T_{genCpm} > T_{genCpmMin}$, with $T_{genCpmMin} = 200$ ms (CPM generation rate of 5 Hz)
- $T_{genCpm} < T_{genCpmMax}$, with $T_{genCpmMax} = 1000$ ms(CPM generation rate of 1 Hz)

A CPM message needs to be generated when:

- A new object is detected
- A change in position of a previously declared static object is detected
- A change in velocity or position of a previously declared dynamic object is detected.

3.4.2.1 Object Confidence Transmitted data should be as close as possible to the original data. In such situation, one idea could be to send directly sensors raw data. This has the advantage that no information loss occurs. In facts, perceived information is transmitted in its raw data form without introducing any kind of treatment on data. However, it is not possible to send raw data directly because it is not feasible in terms of channel load and there is no guarantee that the receiving user has the necessary means to process them. For this reason, it is often preferred to send a compact representation of the perceived environment. If the channel constraints allow it, it is also possible to attach into the CPM some raw data representation.

In order to fulfill the whole requirements of CPM standard, some processing at low levels is needed. In particular, it is required to send an object and to provide an index that quantifies the confidence level of the provided information. This has to be done to provide the receiving user a means to evaluate the quality of a detected piece of information. Such confidence needs to be computed in a way that it can have the same meaning for every user that has access to the shared object. However, there exist some cases in the literature where this value depends on the particular method that has been used to compute it. Confidence needs to be computed considering coherency with previously sent CAMs. This can help in tracking objects and in associating the new detection to the previously existing ones.

3.4.2.2 Objects localization and HD maps CPM deals only with objects that move or have the ability to move. This assumption corresponds to driving scenarios. This imposes that objects must be located on the driving lanes or pedestrian walks. If a map-based representation of the driving environment is available, it can be useful to have map-matching procedures to localize objects on the scene at lane-level. Such phase has to be included into the CPM pre-processing part. Moreover, the map matching results should be sent to other users, in order to provide also the vehicle location inside a high-definition (HD) map and also their occupancy at lane level. However, to exploit such information in a proficient way, one needs to assume that the HD map is the same for every agent, which is, in general, not true. In our case, we assume that both the remote intelligent infrastructure and the autonomous vehicle share the same HD map representation of the driving environment.

3.4.2.3 CPM Message Format The CPM format is made of several containers, as the usual structure of ETSI standard messages. In Figure 7, we illustrate the general structure of a CPM message.

On Figure 7, the ID of the sender is contained into the ITS PDU header. As

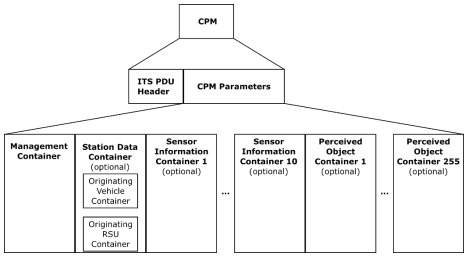


Figure 7: General structure of a CPM message. Figure courtesy of [2]

we did for the CAM before, we need to ensure that every sender (Vehicles and RSU) has a unique identifier. This time, we also need to specify if the sender is a vehicle or a road side unit (RSU).

If the sending entity is a vehicle, it is strongly advised to specify into the Originating Vehicle Container the information about the dynamic of the vehicle (if it is available). On the other hand, if the message is generated by an RSU, containers need to provide references to identify the infrastructure on the HD road map. This is useful to localize information in the correct working frame. In our work, we consider the map frame as the world frame. As a consequence, information perceived by both the intelligent infrastructure and the autonomous vehicle on board sensors is converted into such frame to be taken into account during navigation.

3.4.2.4 CPM Payload Description As we did previously, we have decided to assign a color to each field of the payload. Such colors are assigned to fields according to the criteria explained in Table 1. Colors are assigned to each field in order to specify the importance of every piece of information in the context of a safety-critical autonomous vehicle navigation application. Contrary to the scenario taken into account in section 3.3.3, we consider now a use case where there is only one AD vehicle in a driving scenario with only MD vehicles. Direct communication exists only between the infrastructure and the AD vehicle and, of course, information about other road agents needs to be estimated by the AD vehicle perception system. Table 1 explains the meaning of each color.

Importance	Field Name	Description
R	Station Identifier	It allows to identify the source of a certain CAM. It also permits to distinguish between several sources of information.
Y	Station Type	Tells us if the sending user is a vehicle or an infrastructure (RSU)
R	Reference position	It provides a position to reference perceived objects relatively to a global provided position. Detected objects are referenced into the vehicle's body frame. Once a CPM is shared, the receiving user should be capable of converting received data in their own frames.
R	Timestamp	A timestamp that indicates the time at which the cam has been sent by the user. It is important to distinguish between the sending timestamp of a CAM message and a timestamp used to date perceived objects.
Management Container Information.		

Importance	Field Name	Description
B	Heading	Value of the vehicle's heading w.r.t. to the true north with a 95% confidence level. This data can help in knowing the vehicle intentions in terms of trajectory. We need to clarify the difference between vehicle heading and vehicle orientation.
R	Speed	Driving speed of the sending vehicle. This measure should be provided with a 95% confidence level.
B	Vehicle Orientation	Angle and angle accuracy of the disseminating vehicle absolute orientation. This value is not equal to the heading, that is computed considering the speed value. An accuracy with a confidence level of 95% should be provided.
R	Driving Direction	Vehicle driving direction (Forward or Backward)
R	Longitudinal Acceleration	Vehicle longitudinal acceleration of the originating user at the reference point of the vehicle. Accuracy value with the confidence level of 95% should be included.
R	Lateral Acceleration	Vehicle lateral acceleration of the originating user at the reference point of the vehicle. Accuracy value with the confidence level of 95% should be included.
R	Vertical Acceleration	Vehicle vertical acceleration of the originating user at the reference point of the vehicle. Accuracy value with the confidence level of 95% should be included.
R	Yaw Rate	Rotation of the vehicle around its center of mass with its 95% confidence level
R	Path	The nominal trajectory of a vehicle. In a map-based approach, Path is represented as an ordered list of identifiers of the road links.
R	Pitch Angle	Vehicle pitch angle with 95% confidence level.
R	Roll Angle	Vehicle roll angle with 95% confidence level.
R	Vehicle Width	Width of the sending vehicle with 95% confidence level
R	Vehicle Length	Length of the sending vehicle with 95% confidence level
R	Vehicle Height	Height of the sending vehicle with 95% confidence level
R	Trailer Details	Details on eventual vehicle trails
Originating Vehicle container information		

Importance	Field Name	Description
R	Intersection ID	Allows to link a CPM perceived from a given intersection to an existing intersection on the HD road map
Originating RSU Container Information		

Importance	Field Name	Description
R	Sensor ID	An identifier of a sensor. This pseudonym is used to relate sensor measurements to the sensor that perceived the measurements. A correspondence between the perceived objects and the corresponding sensor id should be instantiated.
R	Sensor type	Type of sensor. (Enumerated value). This field can indicate information not only from a single sensor, but also information fused from several sensors.
R	Vehicle Sensor	Specifies if the sensor is mounted on a vehicle, other characteristics are provided in [tab vehicle sensor]
R	Stationary Sensor	Specifies if the sensor is mounted on a roadside infrastructure, other characteristics are provided in [tab infrastructure]
Sensor Information Container		

Importance	Field Name	Description
R	Ref. Point Id	Identification of a reference point in the case the sensor is mounted on the trailer
R	Sensor Position X offset	Mounting position of the sensor in the x position w.r.t. the reference point of the vehicle
R	Sensor Position Y offset	Mounting position of the sensor in the y position w.r.t. the reference point of the vehicle
R	Sensor Position Z offset	Mounting position of the sensor in the z position w.r.t. the reference point of the vehicle
R	Range	Value of the sensor range
R	Horizontal opening angle start	Start of the horizontal opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Horizontal opening angle end	End of the horizontal opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle start	Start of the vertical opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle end	End of the vertical opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
Vehicle Sensor Container		

Importance	Field Name	Description
R	Sensor Position X offset	Mounting position of the sensor in the x position w.r.t. the reference point of the infrastructure
R	Sensor Position Y offset	Mounting position of the sensor in the y position w.r.t. the reference point of the infrastructure
R	Sensor Position Z offset	Mounting position of the sensor in the z position w.r.t. the reference point of the infrastructure
R	Range	Value of the sensor range
R	Horizontal opening angle start	Start of the horizontal opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Horizontal opening angle end	End of the horizontal opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle start	Start of the vertical opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle end	End of the vertical opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
Stationary Sensor Container		

Importance	Field Name	Description
R	Circular	Sensor with a circular view. It provides the radius of the sensor field of view and the center point w.r.t. the vehicle body frame
R	Polygon	This can be used to provide a detection polygonal area. This area can be associate to one sensor or considered as the union of several sensors characteristics. In the latter case, the sensor type should be set to “fusion”. The field PolyPoint provides the geometry of this detection area
R	Ellipse	This field can be used to provide a description of an elliptic detection area. The required information is only the geometry of the ellipse and its orientation in the frame of the infrastructure.
R	Rectangle	This field can be used to provide a description of a rectangular detection area. The required information is only the geometry of the rectangle and its orientation in the frame of the infrastructure.
Detecting characteristics of a sensor		

Importance	Field Name	Description
R	Object ID	Identifier of the detected object. This id is unique for every object from the same user. This id should help in identifying different objects. Before labeling with this id, detected objects should be refined via data fusion and tracking procedures in order to have a consistent estimation of objects motion. It is recommended to use the same id for the same objects in subsequent CPMs to facilitate the association.
R	Sensor ID	Id of the sensor that detected the perceived object
R	Time of Measurement	A timestamp that states the exact time at which the measurements from the detected object have been taken. This must not be confused with the message timestamp. It is possible to express this time relatively to the message timestamp. Information for synchronization should be provided.
R	Object Age	Provides the age of the detected object. In order to have this field, several data association procedures need to be taken into account before sending the perceived objects
R	Object Confidence	The confidence associated to an object. This confidence should be computed in a way that it is equal for every user, i.e. every road entity can have the same information from this value. Objects with confidence under a certain threshold value should not be sent.
R	X, Y, Z Distances	Absolute distance from the detected object to the user reference point in the three coordinates x, y, z at the time of measurement. This distance is expressed in the detecting user reference frame. A confidence level of 95% should be provided.
R	X, Y, Z Speed	Relative speed of the detected object from the user reference point in x, y, z directions at the time of measurement. This parameter should be estimated as well as possible in order to track the object. A confidence level of 95% should be provided.
B	X, Y, Z Acceleration	Relative acceleration of the detected object from the user reference point in the x, y, z directions at the time of the measurement. A confidence level of 95% should be provided.
B	Yaw angle	Relative yaw angle of the object from the user reference point. This angle is computed w.r.t. the x direction of the detecting user body frame. A confidence level of 95% should be provided.
B	Object Bounding Box	A bounding box representing the detected object. This object can be modeled as a parallelepiped, with the 3 dimensions Length, Width and Height. Some different and more detailed shapes of the road entity as a mesh or a surface estimation can be considered to be included in this field. We also need to associate a level of uncertainty relative to these 3 measures, in order to quantify risks in estimation of the detected user boundaries.
B	Object reference point	The reference point relative to the perceived object. Provided measurements are computed w.r.t. this point.
B	Object dynamic status	Classification of a perceived object towards the capability to move. Three statuses are possible:Dynamic Has been dynamic Static
R	Classification	Provides the classification of an object in several pre-defined categories.
Perceived Object Container		

3.4.2.5 CPM Reference Position For vehicles, we consider the reference position (i.e. the origin of the vehicle body frame) as the center of the front side (i.e. the width) of the bounding box of the vehicle, according to the CPM standards. However, there exists several models in literature that consider the origin of the vehicle body frame placed on the middle of the rear wheels axis. Other implementations also suggest putting it on the middle of the back side of the vehicle bounding box. It is mandatory to define a unique standard for this

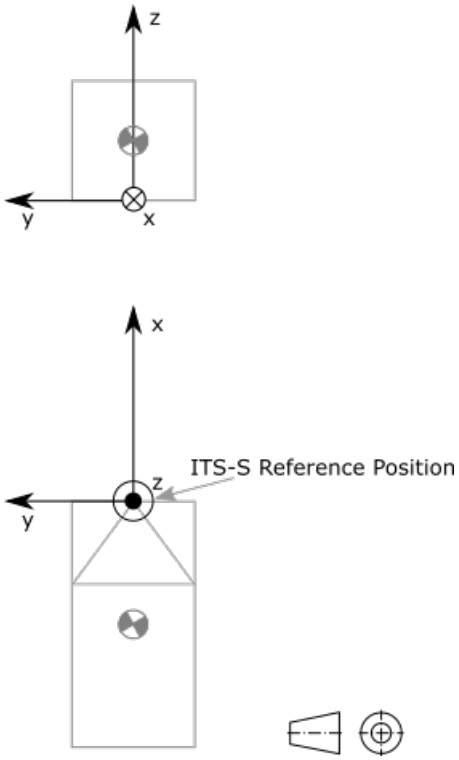


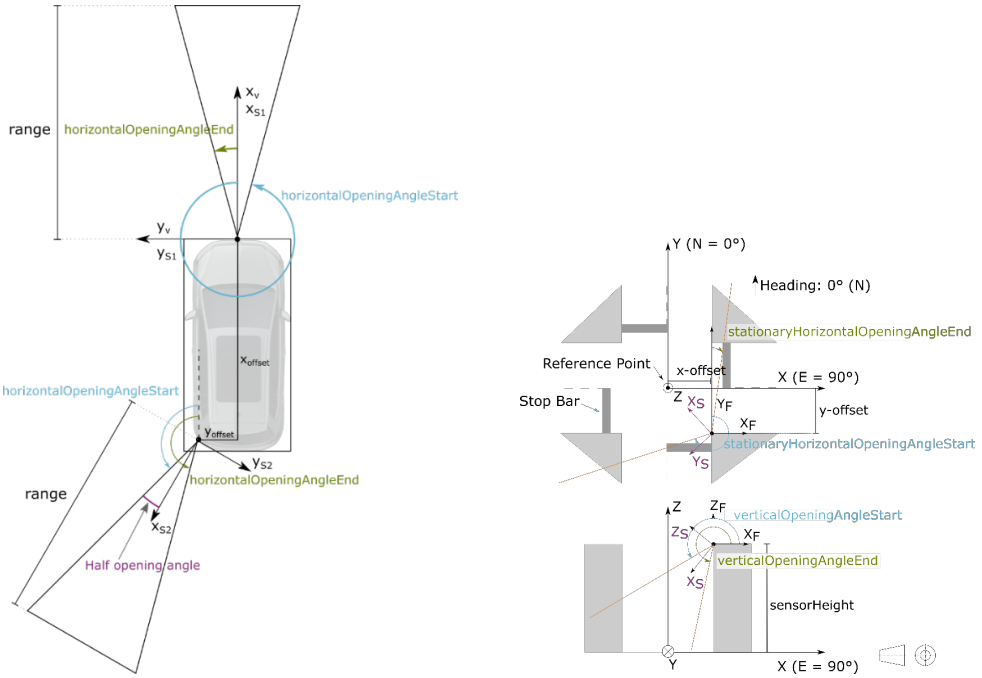
Figure 8: Different standards for the body frame.

field. If this is not possible, transformations to pass from the alternative vehicle body frame to the standard one must be provided.

If the user is a RSU, the origin of the local frame should be defined as a point of the infrastructure (e.g. the point in which a camera is mounted).

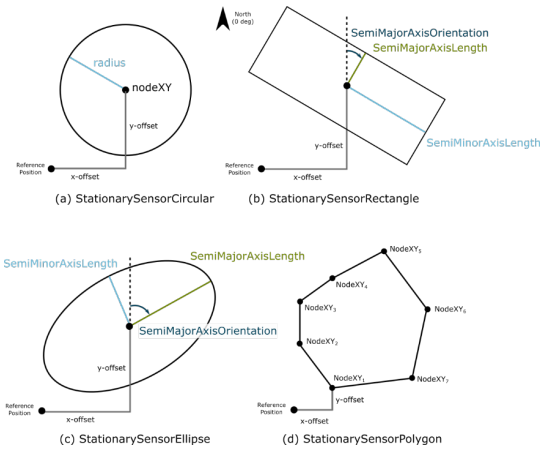
3.4.2.6 Sensor mounting specifications In the following figure (Fig. 10), we can see an example of the sensors parameters that can be described in the Sensor Information container.

3.4.2.7 Sensors field of view description It is possible to describe the field of view of a sensor according to its characteristics. It is possible also to fuse several fields of view obtaining a polygon describing a more complex covering zone.



(a) (b)

Figure 9: Sensors mounted on a vehicle and on an intelligent infrastructure with parameters description.



(a) StationarySensorCircular (b) StationarySensorRectangle

(c) StationarySensorEllipse (d) StationarySensorPolygon

Figure 10: Illustration of the different sensors fields of view encoded in CPM.

3.5 MAP Message (MAP)

The main goal of the MAP message is to announce the topology of the roads, crossroads, roundabout that a vehicle may follow. The topology is described as lanes for e.g. vehicles, bicycles, public transportation that connect with each other and the allowed maneuvers, and signal group id for intersections equipped with Traffic lights. Figure 11 shows an example of the topology of an intersection. Basically each ingress lane is connected with one or more egress lanes which define the allowed maneuvers in the intersection. This "connection" includes the signal group identifier, which is the link for signalization between the topology and the corresponding signalling.

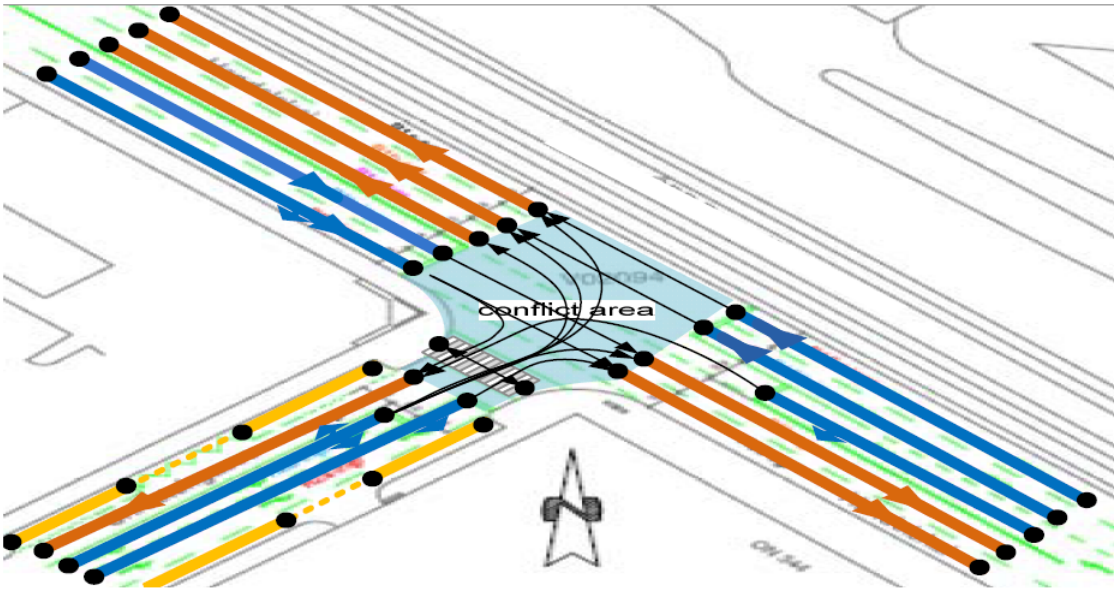


Figure 11: Intersection topology.

3.5.1 MAPEM payload description

MAP Extended Message (MAPEM) is the extension of MAP message for Europe. It is the MAP message as defined in SAE J2735 wrapped in the European ITS PDU format. MAPEM is defined in ETSI TS 103 301 document [4]. It mainly contains an Intersection Geometry List which is mainly composed of

- an intersection reference id, that identifies uniquely the intersection
- a reference point from which data points of lane set are offset until a new point is used.

- a lane list that describes all the ingress and egress lanes

The lane list is a set of generic lanes that are mainly made up of

- a lane id, that is unique within the intersection
- a descriptive name
- lane attributes that provides information about the basic selected lane type, directions of use, Geometric co-sharing and type specific attributes
- the allowed maneuvers for this lane
- a node list that is lane spatial path information as well as various Attribute information along the node path
- the connections to other lanes with tha associated signal Group Id, which is mainly used to correlate with the information sent in the SPAT message.

3.5.2 MAPEM generation

For Tornado project, MAPEM was sent at 1Hz, which was far enough as the topology of the intersections is static. The vehicle must just receive the MAPEM message at least once before crossing the intersection.

3.6 Signal Phase and Timing Message (SPAT)

The main goal of the SPAT is to announce the current states of an intersection managed by traffic lights. It includes safety-related information for supporting vehicles to execute safe maneuvers in an intersection area. The goal is to enter and exit an intersection "conflict area" in a controlled way. It announces in real-time about the operational states of the traffic light controller, the current signal state, the date of next phase change. Figure 12 shows an example of the description of a crossroad managed by traffic lights.

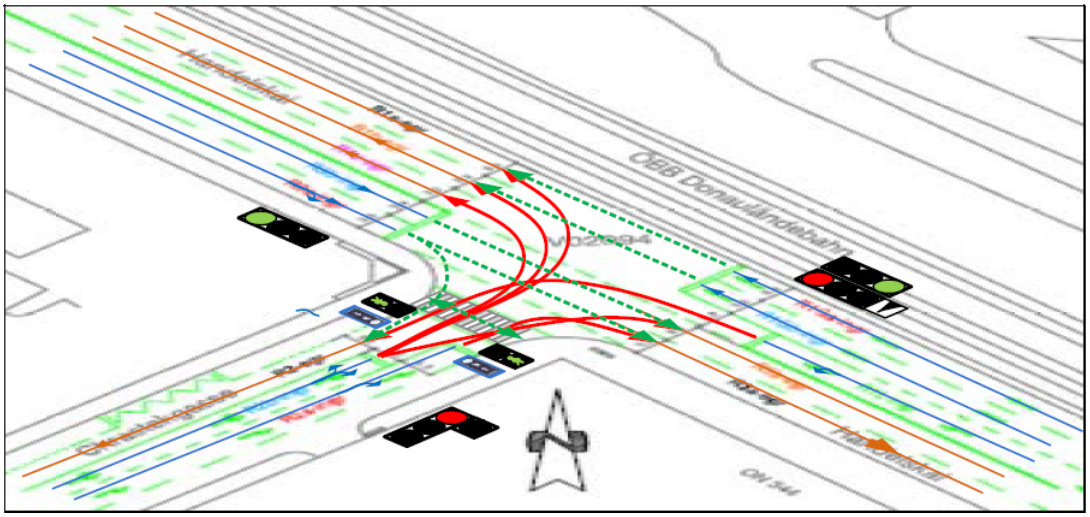


Figure 12: Intersection signaling status

3.6.1 SPATEM payload description

SPAT Extended Message (SPATEM) is the extension of SPAT message for Europe. It is the SPAT message as defined in SAE J2735 wrapped in the European ITS PDU format. SPATEM is defined in ETSI TS 103 301 document [4]. It mainly contains an Intersection State List which is mainly composed of

- an intersection reference id, that identifies uniquely the intersection.
- a movement list. Each Movement is given in turn and contains its signal phase state, mapping to the lanes it applies to, and point in time it will end, and it may contain both active and future states.

The movement list is a set of movement state that are mainly made up of

- a Signal Group Id, which is mainly used to correlate with the information sent in the MAP message
- a movement event list that mainly contains the dates at which the current and next phases change will occur.

3.6.2 SPATEM generation

For Tornado project, SPATEM was sent at 1Hz, which was enough as there was no pre-emption or prioritization requests sent to a traffic light controller. It is important to note that SPAT announces a date of changement and not a remaining time (which allows to bypass latency problem) and as there is no prioritization

request, the date of current phase change is static. The vehicle must just receive the SPATEM message at least once before crossing the intersection.

4 The camera based perception system

The camera based perception system is designed to provide, in real time, important information to autonomous vehicles in poor visibility junction such as a roundabout or in a strong interaction area between vehicles and vulnerable road users :

- targets' identities: vehicles, motorcycles, trucks or pedestrians
- targets' accurate positions and poses
- the directions of the targets

4.1 Camera latency

An efficient roadside perception system must provide information in real time with the lowest latency. During the experimentation, we notice that cameras have non-negligible shuttle lag for our application. The shutter lag is the delay between triggering the shutter and when the photograph is actually recorded. This is a common problem in the photography of fast-moving objects or people in motion. In the section 4.5, we will show that it is critical for a real-time application to have a camera with a shutter lag as short as possible. In our project only one model was used. It is the Basler BIP2 1300C.

4.2 Camera calibration

The camera calibration is an important part of the system installed on the roadside. This step is necessary to extract real world 3 dimensional information from the two-dimensional image data. The procedure can be equated with determining intrinsic and extrinsic camera parameters. Intrinsic parameters deal with the camera's internal characteristics, such as its focal length, skew, distortion, and image center. Extrinsic parameters describe its position and orientation in the world. Knowing intrinsic parameters is an essential first step for extracting real-world information, as it allows you to estimate the scene's structure in Euclidean space and removes lens distortion, which degrades accuracy. Our approach is based on the Zhengyou Zhang's 1999 paper [30]. The main geometric principle of the camera calibration and the understanding of real world scene in computer vision is covered by the Hartley and Zisserman's book [13]. The calibration of the camera includes intrinsic and extrinsic calibration. The intrinsic calibration is done only once and the extrinsic calibration must be done each time the camera is moved.

4.2.1 Intrinsic calibration

The intrinsic calibration is achieved using the OpenCV library on Python. We also use the OpenCV chessboard. The figure 13 is showing an undistorted result for one camera.

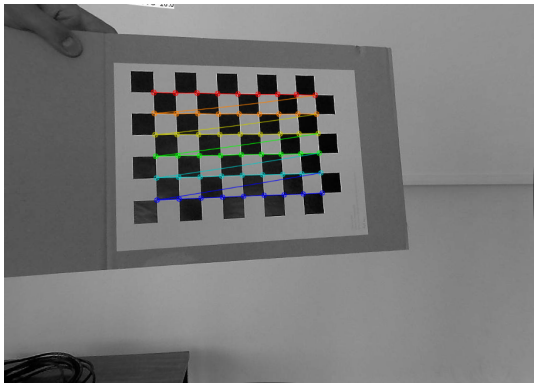


Figure 13: The undistorted result of the chessboard image by the OpenvCV intrinsic calibration algorithm

The useful result is the camera matrix (1) and the distortion vector (2) that are the input of the perception system.

$$\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$d = (k_1, k_2, k_3) \tag{2}$$

where (f_x, f_y) is focal length and (c_x, c_y) is optical center. The distortion coefficients are parameters of the radial distortion model given by the equation (3)

$$\begin{cases} x_u ndist &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_u ndist &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases} \tag{3}$$

The intrinsic camera calibration by OpenCV can be found on: https://opencv-python-tutroals.readthedocs.io

4.2.2 Extrinsic calibration

The extrinsic calibration is carried out on the field. The final goal is to be able provide detected and tracked vehicle's position in a well-known ground frame of reference. The generally used reference to describe the land vehicle trajectories is the East, North, Up reference (ENU). In our project, the CAVs and the perception

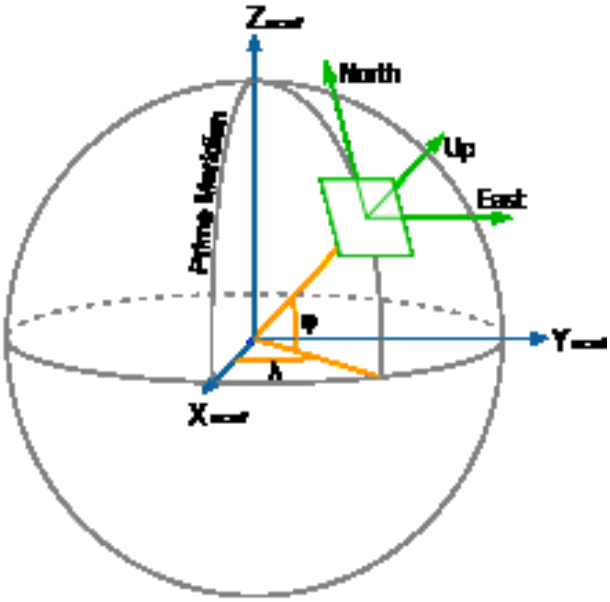


Figure 14: ENU and ECEF reference. (Wikipedia https://en.wikipedia.org/wiki/Axes_conventions)

system will used this frame of reference. The figure 14 is showing the ENU and the Earth Centred, Earth Fixed (ECEF, used by GPS) references.

We want to install the system on several sites including four in Rambouillet's Bel-air shopping center where the project demonstrate the secured CAV system and the roundabout in front of the Université Technologique de Compiègne (UTC). The extrinsic calibration is to estimate the position and orientation of the camera. To do that, we choose in the image of the scene remarkable points such as roadside panels, roadside public lights and lane markings. The positions of these points in the real world must be available. We use the Renault's confidential map with accurate positions from GPS. The first step of our the procedure is to associate manually the chosen points to the numerical Map's data. The figure 15 is showing our method.

The second step is the estimation of the rotation matrix and the translation vector of the camera using Nelder-Mead's Simplex approach. The translation matrix is the camera's position in the ENU reference and the rotation matrix provides the camera's orientation in the same reference. We first verify that the chosen points can be back projected into the image and the projection is not too far from the clicked points, the errors of the back projection can be an indicator of the accuracy of the calibration. For the two Rambouillet's roundabouts the mean errors back projection are 2.5 and 3.2 pixels, and for the UTC's roundabout, the error is 2.7 pixels. We finally provide the visual result of the calibration

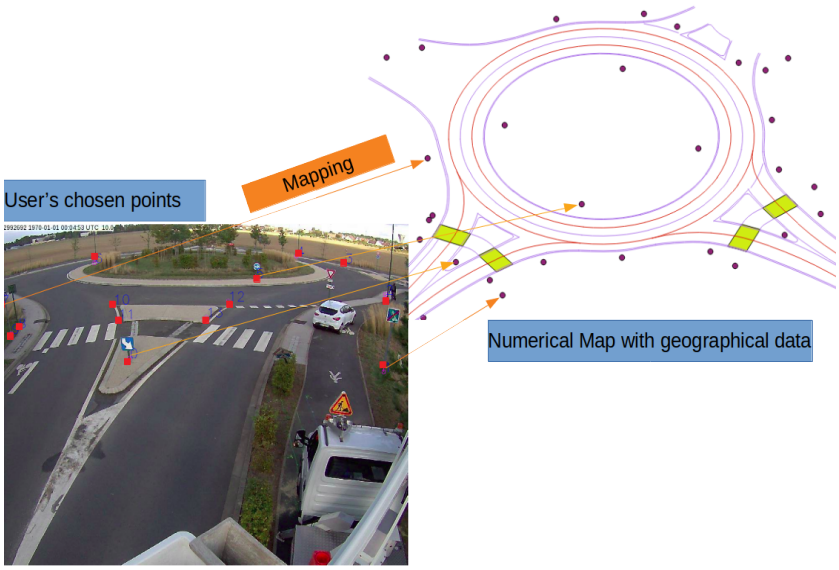


Figure 15: The data association between the image and the numerical map is done manually.

by choosing 21 points in an image roundabout in "rue d'orphin" and map them into the real-world ENU coordinates. Later on, we will validate the calibration by comparing the GPS embedded in the UTC autonomous vehicle using the detection and tracking algorithm presented in the following section 4.3.

4.3 detection and tracking algorithms

To assist the CAV in trajectories planning, the perception system should provide every moving road users' information (position, speed) nearby the vehicle to predict the upcoming interactions between the CAV and the other users. The positions and speed estimations are achieved by using computer vision's detection and tracking algorithms. The target's classification and the dynamic information must then be broadcast to every CAV in and nearby the intersection area.

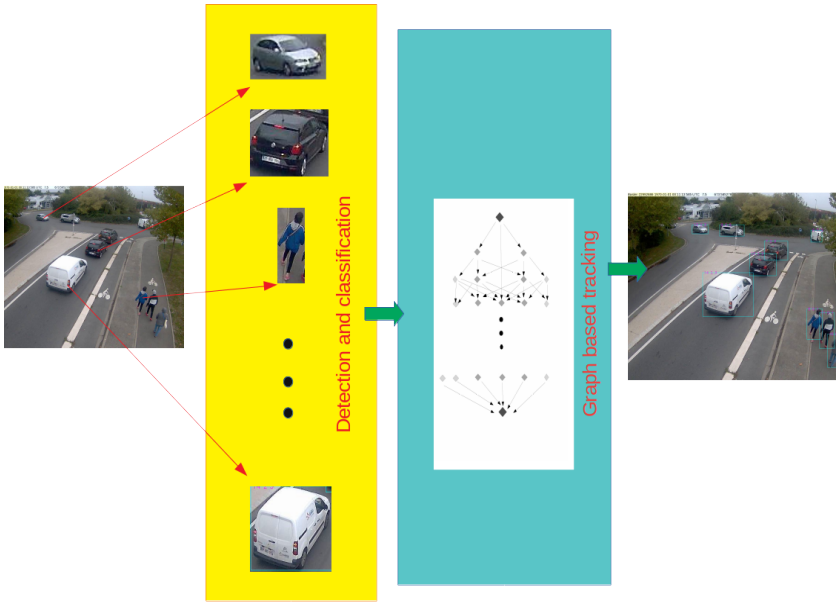


Figure 16: The data association between the image and the numerical map is done manually.

The proposed camera's perception system is based on two main algorithms: the convolutional neural network (ConvNet) [17] based YOLO (You Only Look Once) [15, 22, 23] classification system and a tracking algorithm based on a graph optimization proposed by [8]. Our method is first to detect every road users (the targets) such as vehicles and vulnerable users and classify them into different classes. The targets are represented by a vector $v_i = (d_i, x_i, y_i, v_{x,i}, v_{y,i})$ where, d_i is the timestamp (every target detected in the same image frame have the same date), $(x_i, y_i, 0)$ are the coordinates in the ENU reference frame and $v_i = (v_{x,i}, v_{y,i})$ is the speed in the ENU reference frame. We will first present the classification part then the tracking part.

4.3.1 Road users detection by deep network

In our system, we want to detect the moving road users that may have interaction with the CAV and may modify the CAV's planned trajectory. The road users are motorized vehicles like personal car, trucks, motorcycles but also the vulnerable users like pedestrians or cyclists. This section is dedicated to the description of the object detector and more specifically of the CNN based algorithms.

4.3.1.1 Generalities A deep detection algorithm consists in classifying objects and in localizing objects in images by using deep convolution networks (CNN). Because deep network can efficiently learn feature space within images, many algorithms have been proposed to detect, classify and localize objects in the scene.

On the one hand, some methods proceed through a two stages algorithm: the first stage is proposing some regions of interest that are removed or confirmed during a second stage (R-CNN[11] and their variants Fast R-CNN [12] and Faster R-CNN [24]). During this last stage, the objects are confirmed (i.e. detected), classified and localized. On the other hand, other methods do not require to pre-detect any objects of interest. In these cases, objects are labeled and localized directly from the image content (Single Shot detectors- SSD [19] and YOLO version 1, 2, 3 [15, 22, 23].

R-CNNs are one of the first deep learning-based object detectors and are an example of a two-stage detector. In the first step of the algorithm, bounding boxes that could contain objects are proposed. In the first R-CNN version [11], the object detector requires an algorithm such as Selective Search [26]. In the second step, the region in the bounding boxes were then passed into a CNN for classification, ultimately leading to one of the first deep learning-based object detectors. The problem with the standard R-CNN method was that it was painfully slow and needs another algorithm such as Selective Search to provides region of interest (ROI) as input.

Girshick et al. published a second paper in 2015 [12] about the Fast R-CNN algorithm made considerable improvements to the original R-CNN, namely increasing accuracy and reducing the time it took to perform a forward pass. However, the model still relied on an external region proposal algorithm. All the proposals of the Selective Search are then passed into the R-CNN component for final classification and labeling but it has been modified to not require to feed all the region proposals to the convolutional neural network every time. Instead, the complete feature space is computed once per image and the feature subspace of each proposal is generated by projecting the ROI of each proposal into the feature space volume and by applying a ROI pooling.

In the same year, [24] published the faster R-CNN, third version of the R-CNN. This last version of R-CNN became a true end-to-end deep learning object

detector by removing the Selective Search requirement and instead relying on a Region Proposal Network (RPN) that is fully convolutional and it can predict the object bounding boxes and the objectness scores (i.e., a score quantifying how likely it is a region of an image may contain an image).

While R-CNNs tend to very accurate, the biggest problem with the R-CNN family of networks is their speed they were incredibly slow, obtaining only 5 to 7 FPS on a GPU for the last improved version of it.

The two-stage detectors appear clearly not fast enough except when using powerful GPU materials. As alternatives, algorithms that compute both boundary boxes and classes directly from feature maps in one step (i.e. without RPN or Selective Search steps) were developed. The most poplular are SSD and YOLO. These algorithms consider object detection as a regression problem, taking a given input image and simultaneously learning bounding box coordinates and corresponding class label probabilities. In TORNADO project, we need an algorithm that treats data as fast as possible with a good accuracy and without requiring a high performance computing hardware. The table 4 are extracted from [24]: it appears clearly that YOLO V3 yields the best rate for the best frame rate. This method has been used in the project. In this table, the second column shows the mean average precision (mAP) that is a popular metric in measuring the accuracy of object detectors when the Intersection of Union (IoU) between the predicted bounding box and the ground truth is greater than 50%. This metric is used in COCO challenges (http://cocodataset.org/#detection-eval) to evaluate detectors. With the resolutions 320×320 and 416×416, YOLO V3 is the faster detector and the mAPs are higher than the SSD's. The other detectors are too slow compared to YOLO V3.

4.3.1.2 YOLOv3 algorithm description YOLOv3 is an improvement of the YOLO network. We first describe YOLOv1 and v2 architectures to finally present the different modifications of YOLOv3.

YOLO divides the input image into an $N \times N$ grid and this algorithm is made to predict that only one object is contained in each grid element (objectness). A fixed number B of boundary boxes is predicted for each grid element. A confidence value is estimated for each boundary and a fixed number of conditional class probabilities C are estimated for this single object i.e the probability that the detected object belongs to one of the set of classes. Finally, YOLO outputs a tensor which shape is: (N,N,X). X is defined by $X = B * 5 + C$.

YOLO architecture (cf. Figure 17) is inspired from GooLeNet architecture and it is based on 24 convolutional layers followed by 2 fully connected layers (FC). Some convolution layers use 1×1 reduction layers alternatively to reduce the depth of the features maps as proposed in the Inception block of Google net. The output

Method	mean Average Precision(mAP)-50	time (ms)
SSD321	45.4	61
DSSD321	46.1	85
R-FCN	51.9	85
SSD513	50.4	125
DSSD513	53.3	156
FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOV3-320	51.5	**22**
YOLOV3-416	55.3	29
YOLOV3-608	57.9	51

Table 4: Comparison of CNN based detector.

tensor before the FCs has a shape (7, 7, 1024). The 2 fully connected layers as a form of linear regression, and outputs a (7,7,30) shape layer parameters that contents $B = 2$ boundary box predictions per location for $C = 20$ classes to be train and evaluation on PASCAL VOC dataset.

The learning step is based on the optimization of the loss function defined by the sum of the 3 following loss functions: the classification loss, the localization loss (errors between the predicted boundary box and the ground truth) and the confidence loss (the objectness of the box). To compute the loss for the true positive bounding boxes that are relevant to the single detected object, the bounding box with the highest IoU (intersection over union) with the ground truth is selected.

YOLOv2 is the first upgraded version of YOLO. It keeps its general strategy and add some improvements. For example, it accepts multi-resolution as input images (we have to make sure that width and height are a multiple of 32). To avoid the problem of vanishing or unstable gradient and the difficult convergence during the training, YOLOv2 proposes to predict several bounding boxes from a set of 5 anchors from which they are derived. To reduce the effect of shallow feature map, reshaped layers are concatenated from low to high resolution to obtained fine-grained feature map. This approach is able to better detect small objects. At 288×288 YOLOv2 runs at more than 90 FPS with mAP almost as good as Fast R-CNN. At high-resolution YOLO achieves 78.6 mAP on PASCAL VOC dataset.

YOLOv2 has been designd with different CNN backbone network. First VGG-16 has been chosen to reach better results on PASCAL VOC dataset and a top-5 rank on Imagenet dataset. If replaced by GoogLeNet architecture, the performance decreases. The backbone has been finally simplify as described on the figure 18.

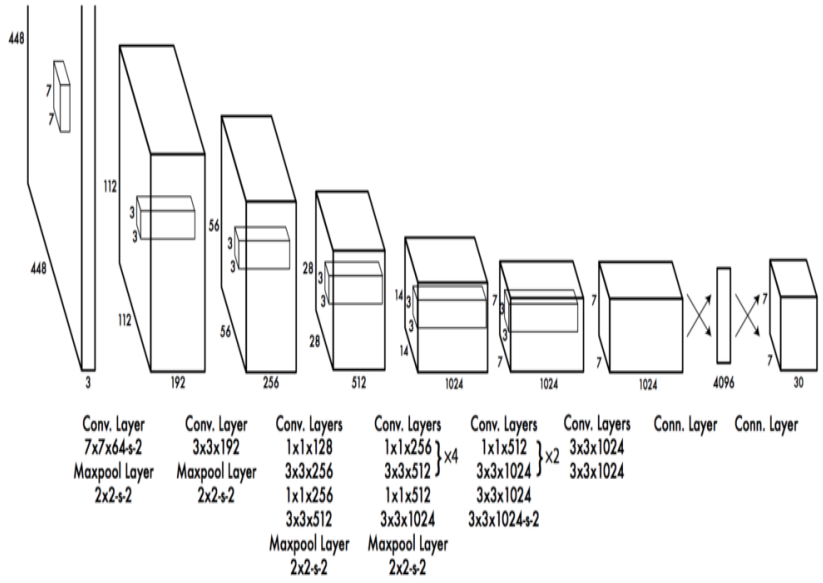


Figure 17: Yolo architecture: 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduces the features space from preceding layers.

As for YOLOv2 the two FC are removed and replaced by three 3×3 convolutional layers each outputting 1024 output channels (cf 19. A final 1×1 convolutional layer is following to convert the $7 \times 7 \times 1024$ outputs into $7 \times 7 \times 125$ ($N = 7$ cells, $B = 5$ boundary boxes each with 4 parameters for the box, 1 objectness score and 20 conditional class probabilities).

YOLOv3 is the network used for TORNADO detection system. It is based on YOLOv2 with 4 main new contributions. The backbone is a new darknet-53 network described in the figure 20 as the feature extractor. Darknet-53 mainly compose of 3×3 and 1×1 filters with skip connections like the residual network in ResNet. Darknet-53 has less billion floating point operations than ResNet-152, but achieves the same classification accuracy at $2\times$ faster that is a very good property for our real-time TORNADO application.

YOLOv3 uses a multi-label approach for classification. Softmax is not used as for YOLOv1 and v2. Independent logistic classifiers is used to perform good performance. Because of this choice, binary cross-entropy loss is used during training. Moreover, this formulation helps when there are many overlapping labels (i.e. Woman, Person, pedestrian) in the dataset. Using a softmax constraints that classes are mutually exclusive.

YOLO v3 makes prediction at three scales, which are precisely given by downsampling the dimensions of the input image by 32, 16 and 8 respectively. Thus it makes 3

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Figure 18: Darknet architecture on which the last convolution layer of VGG-16 based YOLO has been removed and replaces.

predictions per grid elements. Each prediction composes of a boundary box, an objectness and C class scores, i.e. $N \times N \times [3 \times (4 + 1 + C]$ predictions. Because of this Feature Pyramid Networks (FPN) like approach, the final architecture contents more than 53 layers as presented in the figure 21

4.3.1.3 YOLOv3 training The latest version of YOLOv3 has been trained on COCO dataset that content 80 object categories of labeled and segmented images. The learned weights are available for downloading by following this link: https://pjreddie.com/media/files/yolov3.weights

The training of YOLOv3 network is carried out using the built-in functionality of the Darknet framework (https://pjreddie.com/darknet/yolo/). This framework allows users to set the network structure using configuration files and specify the hyper parameters for the network and its training. After a training step, the user can apply the neural network defined by the learned weights to process images or videos and check the quality of the training on set of test samples.

Neural network training takes place over several epochs. During this process, at

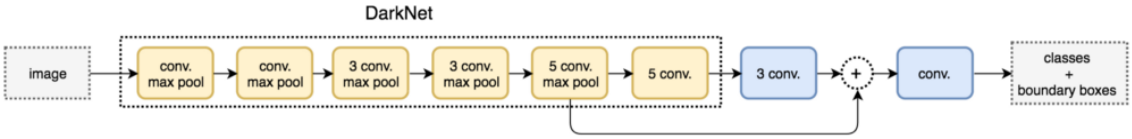


Figure 19: The two FC are removed and replaced by three 3x3 convolutional layers each outputting 1024 output channels. A final 1×1 convolutional layer is following.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 20: Architecture of the Darknet-53.

each stage, the weights are optimized using gradient descent based method and a back propagation process. When training the network âfrom scratch,â the weights are initialized by randomly non zero values distributed to avoid error propagation.

In TORNADO project we use the "fine tuning" approach. This approach uses a pre-trained model, i.e. a set of pre-trained weights, that performs a quite similar recognition function. The objectif are (1) to reduce the set of classes to the objects the TORNADO system can perceive, (2) to improve the bounding boxes accuracy and (3) to remove the false positive detection. YOLOv3 has been trained on PASCAL VOC, Imagenet and COCO datasets. The best model obtained for

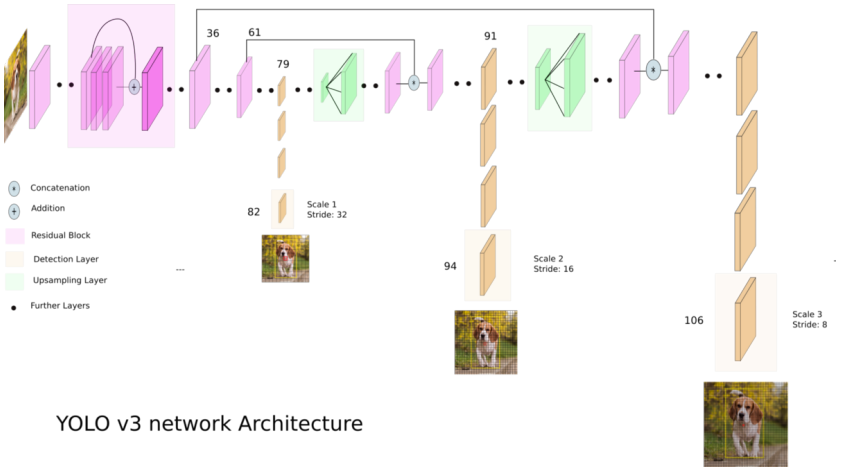


Figure 21: Architecture of the backbone YOLOv3 with FPN.

YOLOv3 has been trained on COCO dataset. We retrieve all the object classes we aim at detecting in the 8 first classes of the COCO labels: person, bicycle, car, motorcycle, bus, truck. The YOLOv3 network has been initialized with the COCO trained weights and trained again by using our TORNADO dataset. TO ovoid overfitting the learning rate (LR) is set lower than the LR used for a training from scratch.

While a training YOLO from scratch needs a high number of images for each class, "fine-tuning" needs less sample per class. The TORNADO dataset that has been use to fine-tune the COCO YOLOv3 model has been extracted from the stream of a video camera installed on the infrastructure near the roundabout which was on the route of the autonomous shuttle. The images were selected taking care not to have the same moving objects several times in the scene, i.e. avoiding taking successive images. Moreover, the object class appear at different positions on the road and roundabout.

The images have been annotated using the Yolo_mark tool available at: https://github.com/AlexeyAB/Yolo_mark. Other tools can be used but each annotated object has to be defined by one line of the form:

$$< object - class >< x >< y >< width >< height >$$

where $< x >, < y >$ are the coordinates of the center of the Bounding Box and $< width >, < height >$ are the size of the bounding box and $< object - class >$ is the class id of the object: person(0), bicycle(1), car(2), motorcycle(3), bus(4), utilitaire(5), truck(6). The 4 values related to the bounding box are normalized by the width and the height of the image. For the training, the TORNADO dataset was cut in two subsets of images: the training and the evaluation subsets representing respectively 80% and 20% of the dataset. The training has been

executed on a NVIDIA P5000 GPU. The convergence ended after 5 days. The figure 22 illustrates what is obtained by the new model. It appears clearly that all objects are detected with the good label and the bounding boxes are accurately estimated that is required for the next part of the algorithm that estimate the 3D location of the detected objects (cf. 4.3.2).

Figure 22: Detection results obtained by applying the fine-tuned darknet model.

4.3.2 Graph based tracker

Tracking multiple targets is important in many computer vision application including Autonomous vehicles trajectories planning, event recognition etc. This problem received many attention during the last decade. In crowded environments, the road users can be occluded or may be not detected or misclassified. Thus, the tracking is often a difficult task.

First, we will review the popular trackers. Kalman Filtering (KF) is an efficient way to address multi-targets tracking when the number of targets remains limited [14, 9]. However, when this number increases, errors and missed detection become more frequent and the these errors are difficult to handle due to the recursive

nature of method if the KF is applied without modification. Particle Filtering (PF) [25] is another popular tracking method and it can avoid the KF problem by exploring many hypothesis. Another method to associate the detection over time is the Dynamic Programming but the computational complexity can be very high [28, 10]. The graph-based tracker is another interesting method for multi-target tracking [29, 6, 21, 16]. Unlike the other method mentioned above, this method is able to handle a huge and varying number of targets. The problem to solve is a min flow cost problem; the cost can be minimized by using linear programming. We will use this approach to track vehicles and vulnerable users in Tornado project, especially the method proposed by [29] that is also used by [21, 16]. Furthermore, the usual graph-based trackers use *the tracking by detection* paradigm: first, object ares detected in each frame of the sequence and in the second step, each detected object in the frame f is associated to a detected object of the frame $f+1$. This carried out for each frame of the sequence minimizing the flow cost of the graph to construct complete trajectories. These applications are initially not used in on-line applications.

Let us first provide a fast explanation of the approach. The data association is based on the observation that there is an analogy between finding non-overlapping object trajectories and finding edge-disjoint paths in a graph; the latter can be solved efficiently by network flow algorithms. In the detection step, each detected target is presented by a vector $d = (x_i, b_i, t_i)$, where x_i is the position of the target, b_i is the size of the target that can be the size of the bounding box and t_i is the time stamp of the target. In the application, the time stamp is the time stamp of the acquired frame. Let $X = d_i$ be a set of targets observations. A trajectory τ_k is defined as an ordered list of the detection vectors $\tau_k = d_{k_1}, d_{k_2}, \cdots, d_{k_n}$. Zhang et al. define the data tracking problem as a MAP problem with the non overlapping constraint:

$$\begin{array}{c} \tau_{op} = argmax_{\tau} \prod_i P(d_i|T) \prod_{\tau_k \in T} P(\tau_k) \\ \tau_k \bigcap \tau_l = \emptyset, \forall k \neq l \end{array} \tag{4}$$

where:

$$P(d_i|T) = \begin{cases} 1 - \beta_i \ \exists \tau_k \in T, d_i \in \tau_k \\ \beta_i, \ otherwise \end{cases}$$

$$P(\tau_k) = P(\{d_{k_0}, d_{k_1}, \ldots, d_{k_n}\}) = P_{\sigma}(d_{k_0}) P_{\lambda}(d_{k_1}|d_{k_0}) P_{\lambda}(d_{k_2}|d_{k_1}) \ldots P_{\lambda}(d_{k_{n-1}}|d_{k_n}) P_{\kappa}(d_{k_n}) \tag{5}$$

β_i is the probability for d_i to be a wrong target or false detection, P_{σ} is the probability for a detection to be the first point in the trajectory, $P_{\lambda}$is the transition probability between two detections of two frames and $P_{\kappa}$is the termination probability. The non overlapping constraints of the optimization problem can be modelized by the 0-1 indicator variables :

$$
\begin{aligned}
f_{\sigma,i} &= \begin{cases} 1 \; \exists \tau_k \in T, d_i \; is \; the \; first \; point \; in \; \tau_k \\ 0, \; otherwise \end{cases} \\
f_{\kappa,i} &= \begin{cases} 1 \; \exists \tau_k \in T, d_i \; is \; the \; last \; point \; in \; \tau_k \\ 0, \; otherwise \end{cases} \\
f_{i,j} &= \begin{cases} 1 \; \exists \tau_k \in T, d_j \; is \; right \; after \; d_i \; in \; \tau_k \\ 0, \; otherwise \end{cases} \\
f_i &= \begin{cases} 1 \; \exists \tau_k \in T, d_i \; is \; in \; \tau_k \\ 0, \; otherwise \end{cases}
\end{aligned} \tag{6}
$$

With the equation6, the trajectory T is non-overlap if and only if :

$$\sum_{i,j} f_{j,i} + f_{\kappa,i} = f_i = f_{\sigma,i} + \sum_{i,j} f_{i,j} \tag{7}$$

The equation 7 is the constraint of the optimization problem 4 and the -log of the objective function can be written with the 0-1 indicators:

$$T = argmin_T \sum_i C_{\sigma,i} f_{\sigma,i} + \sum_{i,j} C_{i,j} f_{i,j} + \sum_i C_{\kappa,i} f_{\kappa,i} + \sum_i C_i f_i \tag{8}$$

where $C_{\sigma,i} = -Ln(P_\sigma(d_i))$, $C_{\kappa,i} = -Ln(P_\kappa(d_i))$, $C_{i,j} = -Ln(P_\lambda(d_j|d_i))$ and $C_i = Ln(\frac{\beta_i}{1-\beta_i})$.

The output of the tracker is the optimal list of trajectories $T = \tau_k$ that maximizes of the following probability.

This optimization problem can be mapped into a cost-flow network with source σ and sink κ. Looking carefully at the costs in the equation 8, there are the transition cost $C_{i,j}$ and the cost C_i of the detection d_i in one of the trajectories of T .

This means that in the cost-flow network, one detection d_i is represented by two nodes (u_i, v_i), u_i being the transition node that can be seen as the predicted position of the previously detected target and v_i being the observation node where the cost depends on the current detected target's data and the previous one's.

The nodes (u_i, v_i)can be grouped by layer. Each layer contains every detections at the same time t.

For instance, let us say, we have a sequence of 3 frames. In the first frame, taken at time t_0, there are 2 detections, in the second frame taken at time t_1, there are 3 detections and in the last frame at time t_2, there are 2 detections. Each node pair (u_i, v_i) represents a detection.

4.3.3 Graph based tracker for the Tornado perception system

In the section 4.3.2, we described graph based tracking that is designed for a off-line application and the algorithm is applied to the entire sequence with usually a large

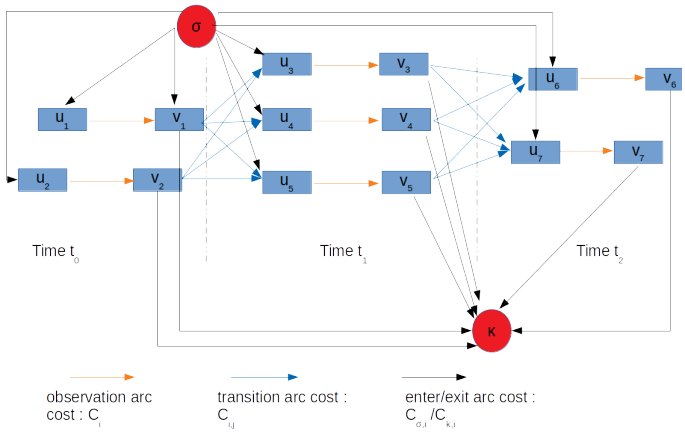


Figure 23: An example of cost-flow network with a sequence of 3 frames. Each node pair (u_i, v_i) represents a detection

amount of frames. The goal of this approach is to be robust to long term occlusions. In our application, the tracker must track vehicles in real time. Recent works show that the multi-targets tracking can be carried out on-line, the well-known on-line tracker are SORT [7]and Deep SORT[27] that only track the pedestrians. These algorithms are based on the tracking-by-detection principle like the graph based tracking but instead of solving occlusion problem, the main goal is to track objects in real time as accurate as possible. The long term occlusions are no more a the main issue of the tracking problem. In SORT, the tracking is carried out with two steps: the prediction step using linear Kalman filtering and the second step is to associate the new detections to existing target using the Hungarian algorithm. For the data association, SORT only assumed that the bounding boxes have a constant shape. If this assumption is true for the pedestrians it is no longer valid for the vehicles because the vehicle's shape vary significantly when it turns right or left. To solve both the short term occlusions issue and the real time constraint issue, we modified the network flow method by [29]. We proceed with a sequence of 3 to 5 image frames. As an example, at timet_0, we have 3 frames with time stamp (t_{-2}, t_{-1}, t_0) as input of the tracker, at time t_1, the frame with the time stamp t_{-2} exits and a the new frame with time stamp t_1 enters in the new sequence of images with time stamp (t_{-1}, t_0, t_1) as the new input of the tracker (cf figure 24 a) Each newly detected target is assigned with a label and it keeps this label until it leave the field of view (FOV) of the camera. However, a target can be occluded temporarily by another tagret and it does not exit the FOV. In [29], the occlusion model is solved using the Explicit Occlusion Model (EOM) for the short term occlusion but this method is only efficient when the number of the

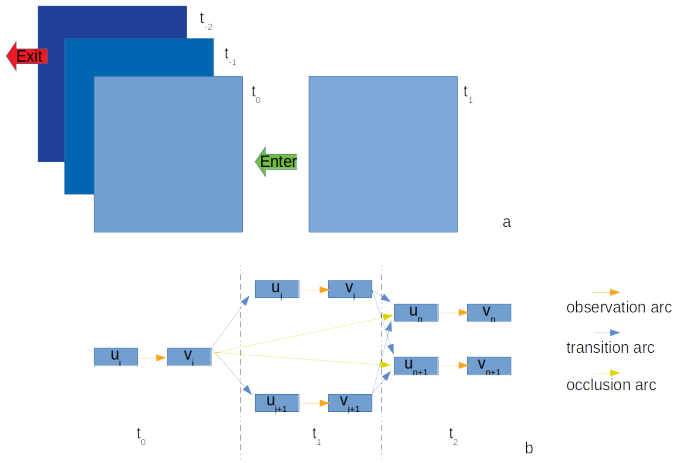


Figure 24: a: The new frame at time $t_1$enters in the sequence of 3 frames and the oldest frame exits the sequence. b: Additional arcs for occlusion ares added in the graph

sequence is important. We use a simpler solution by adding occlusion arcs that are transition arcs connecting nodes from the i^{th} frame to nodes of the $(i+k)^{th}$ frame with $k > 1$.(cf figure 24 b). Like SORT, we also use the Kalman filter to predict the target's new position and to smooth the vehicle's speed. The Kalman filter predicts the future positions of the targets once the data association is carried out using the linear model 9:

$$\begin{cases} x_{n+1} = x_n + \dot{x_n}\triangle t \\ y_{n+1} = y_n + \dot{y_n}\triangle t \\ \dot{x}_{n+1} = \dot{x}_n \\ \dot{y}_{n+1} = \dot{y}_n \\ \varphi_{n+1} = \varphi_n \end{cases} \tag{9}$$

4.3.3.1 Target localization In the perception system, it is important to locate the moving road user on the road that is assumed to be flat. The position on the road surface is $z = 0$. The main function of the calibration described in the sections 4.2.1 and 4.2.2is to map the image reference frame into the ENU frame. The YOLO detector's output is the target's bounding boxes in the image reference frame and the objectness that quantifies how likely it is for a bounding box to contain an object, as first defined by [5]. If the detected target has a objectness higher than a threshold, we used the mid point in image coordinates frame of the bottom segment of the bounding box as the target's location (u, v) and map it into the ENU frame $(x, y, z = 0)$ as shown in figure25. We can notice here

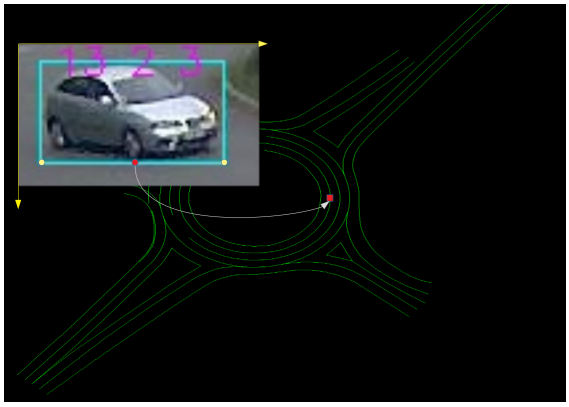


Figure 25: The mid point of the bounding box bottom segment is mapped into the ENU coordinates frame, $(x, y, z = 0)$.

that the position estimation is sensitive to the bounding box and how it surrounds the moving object. However, during how test, the mid point of the bottom of the bounding box was the best point to estimate the object's position.

4.3.3.2 Target classification The classification of the target is an important information for the autonomous vehicles. Indeed, the identity of the road user wil provide important information about approximately how large it is, how long it is and how fast it can move without any measurement. In the perception system, we used Yolo v3 that learnt the COCO database [18]. In this data base , more than 80 classes are learned and the useful classes for our applications are:

- person as pedestrian with the label 0
- bicyle with the label 1
- car with the label 2
- motorbike withe the label 3
- bus with the label 5
- truck with the label 7

We can also extend this list with traffic lights(label 9) and stop signs(label 11) or dogs (label 16) that can be detected in a road scene.

model	length/width
Renault Koleos	4535mm/1900mm
Renault Zoe 2020	4087mm/1730mm
BMW coupe serie 2	4432mm/1774mm
Volvo S60	4761mm/1850mm
Volkswagen Golf SW	4567mm/1799mm

Truck's volume	length/width/height
$10m^3$ to$12m^3$	5900mm/1900mm/2400mm
$20m^3$ to $23m^3$	7000mm/2200mm/2700mm
$30m^3$ to $40m^3$	8000mm/2300mm/3200mm
$50m^3$	10000mm/2500mm/3800mm
$80m^3$ to $100m^3$	18500mm/2600mm/3800mm

Table 5: Table: Dimension of personal car and truck

4.3.3.3 Target's yaw angle The vehicle yaw angle is another important information that can help the CAV to make decisions in a junction. The yaw angle provides information to predict future trajectory of the vehicle. In a roundabout, it is important to predict if a vehicle is about to exit the roundabout. With a mono camera system, the yaw angle is difficult to be estimated because the 3D information is not available. To estimate the yaw angle, we proposed two approaches:

- the first one, can be used the a very general situation where the bounding box of the vehicle and the vehicle class is available.
- The second one, is a more faster and it is dedicated to the track vehicle in a roundabout.

The first approach is based on the idea that the same kind of vehicle have approximately the same dimension. The following tables is showing the dimension for personal car and trucks. The dimensions of the personal car are quite homogeneous where the

length is in the majority less than 5m and the width is approximately 1.8m. The approximation made here is quite good for the a camera system whose accuracy is more than 10cm. The trucks dimensions show more disparity in length and in width. It will be difficult to classify the smallest trucks and the biggest personal cars. The buses is a vehicle class that usually has well-defined dimension in France. The classical model has a length of 12m and the width is 2.50m. The articulated version is longer with 18m.

Assuming the length and the width is constant for a given vehicle category, the width of the bounding box can be used to estimate the yaw angle. We project

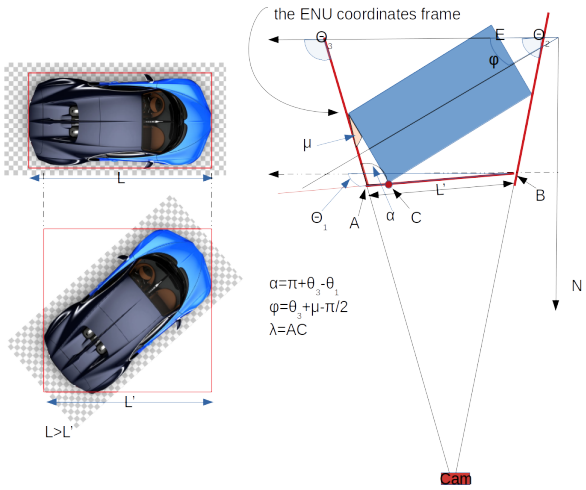


Figure 26: When the vehicle is seen in its side view, the bounding box's width is the vehicle's length. When the vehicle is seen with other angle , the bounding box is shorter. The yaw angle can be computed if the vehicle's length is given and if the angles $\alpha, \mu, \theta_1, \theta_2, \theta_3$,shown in the figure, are available.

the bounding box into the ENU coordinates frame. The bottom border of the bounding box in the ENU coordinates frame is the location of one point of the vehicle that is the closest point to the camera. Let φ be the yaw angle. We are measuring the yaw angle as the angle between the E axis and the heading of the vehicle. The estimation of φ is a geometrical problem as shown in the figure 26.

Once the bounding box is mapped into the ENU coordinates frame, the position of the points $A = (x_a, y_a)$ and $B = (x_b, y_b)$ shown in the figure 26 is known and with the camera position $cam = (x_c, y_c)$, we have the three equations of the lines $(cam, A), (cam, B)$ and (A, B):

$$\begin{array}{ll} y = a_1 x + b_1 & (A, B) \\ y = a_2 x + b_2 & (cam, B) \\ y = a_3 x + b_3 & (cam, A) \end{array} \tag{10}$$

Thus,

$$\begin{array}{l} \theta_1 = atan(a_1) \\ \theta_2 = atan(a_2) \\ \theta_3 = atan(a_3) \end{array} \tag{11}$$

Let $AB = L', \lambda = AC$. Looking at the figure 26, we claim that$\alpha = \pi + \theta_3 - \theta_1$. We also assumed that the vehicle width w is known as soon as the vehicle is well classified. Following the sines law, $\mu = arcsin(\lambda \frac{sin(\alpha)}{w})$ and $\varphi = \theta_3 +$

$arcsin(\lambda\frac{sin(\alpha)}{w}) - \frac{\pi}{2}$. We estimate the yaw angle by solving the simple optimization problem:

$$\begin{aligned}\lambda^* = argmin_{\lambda\in\left[0,\frac{w}{sin(\alpha)}\right]} - y_a + (a_2x_a + b_2) - \lambda\left(sin(\theta_1) - a_2cos\left(\theta_1\right)\right)\\ -L\left(a_2sin(\theta_3 + arcsin\left(\frac{\lambda sin(\alpha)}{w}\right)) + cos(\theta_3 + arcsin\left(\frac{\lambda sin(\alpha)}{w}\right))\right)\end{aligned} \tag{12}$$

The optimization is a convex problem and λ is defined in a bounded interval. Thus, the global minimum may not exist in the interval when the assumed vehicle length is wrong. The difficulty in this problem is to keep λ inside the interval while we need a fast convergence. One fast algorithm is the Newton-Raphson's (NR) method but it is difficult to keep λ inside the interval. We combine NR method with a gradient descent by checking the loss function: if the value of the loss function is small enough with a high derivative or if λ is close to $\frac{w}{sin(\alpha)}$, we switch the NR algorithm to the gradient descent instead of NR with a small step size. If the gradient descent does not converge to 0 and λ is close to $\frac{w}{sin(\alpha)}$, we stop the search because no solution exists. The length assumption is wrong in that case. The loss function in the equation12is different if the closest point to the camera C is close to B instead of A. But this is not the problem because C's position depends on the position of the vehicle on the roundabout. In the figure27, we show the map we use to determine if C is near the corner A or the corner B.

The second approach is based on the vehicle position(x, y)in the roundabout and the vehicle is far from an exit of roundabout. We also assume that the roundabout is circular with the center $cen = (x_c, y_c)$. At this position, the vehicle is tangent to the cercle of cen with the radius $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$

In the Tornado perception system, both approaches are used. When the vehicle is in the blue area where it cannot exit and won't change its trajectory, the second approach is used because it is faster. Otherwise, the first approach is used . In the figure 27, in purple area, the vehicle has choice to exit or to stay in the roundabout. In this area, the first approach is better.

4.3.3.4 Target's speed The vehicle speed estimation is one hard task for the mono-camera based perception system developed in the Tornado project because the detection and tracking algorithm must run very fast to broadcast the information in real time. The speed computation is based on the vehicle detection and tracking. The better is the target modeling the more accurate will be the speed estimation. Currently, the speed estimation is based on the tracking of two image frame with a Kalman filter as it is mentioned in beginning of this section. The speed is first estimated by using positions of the targets measured in two consecutive frames: $\dot{x}_n = \frac{x_n - x_{n-1}}{\triangle t}$ and $\dot{y}_n = \frac{y_n - y_{n-1}}{\triangle t}$. As expected, the result is very noisy with very significant differences between two consecutive values. Kalman filtering allows the

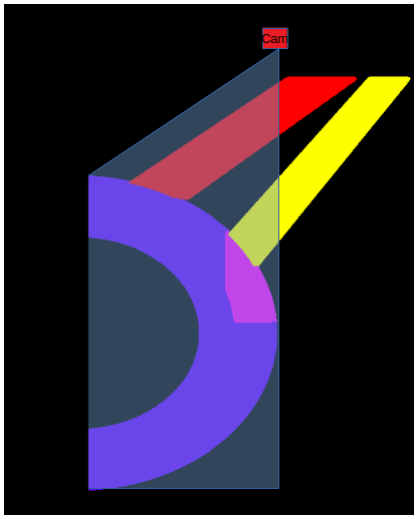


Figure 27: Roundabout in Rambouillet. In the blue area, the vehicle's yaw angle is deduced from the only information of the position. In the red and yellow area, the point C is close to the corner A.(fig 26). When the vehicle is in the purple area, it can exit or stay in the roundabout. The point C is close to the corner B.

calculations to be smoothed and to obtain more realistic values. The result before and after the Kalman filter processing is shown in figure 28. In this figure, we compare the result with the RTK GPS measurement acquired during the on field experimentation as it is presented in the section 4.4.

4.3.3.5 Target's information broadcast by the Road Side Unit The traffic information is broadcast using the RSU by LACROIX City. The RSU allows 255 perceived object and perceived targets information are sent to the RSU using the Collective Perception Message's (CPM) perceived object container. As the technical report [2] suggests, the position and the occupied space of the target are derived from the so-called object reference point, yaw angle, vehicle's length and vehicle's width. We show the object reference point in the figure 29.

4.4 experimentation

The roadside camera based perception system is evaluated using an instrumented vehicle equipped with a RTK GPS installed near the rear axle of the vehicle in the roundabout of the experimentation and demonstration site of the Tornado project at the commercial area of Rambouillet. The vehicle provides accurate positioning and yaw angle. Our evaluation aims to evaluate the accuracy of the estimated target's position, the vehicle yaw angle and target's speed in the field of view

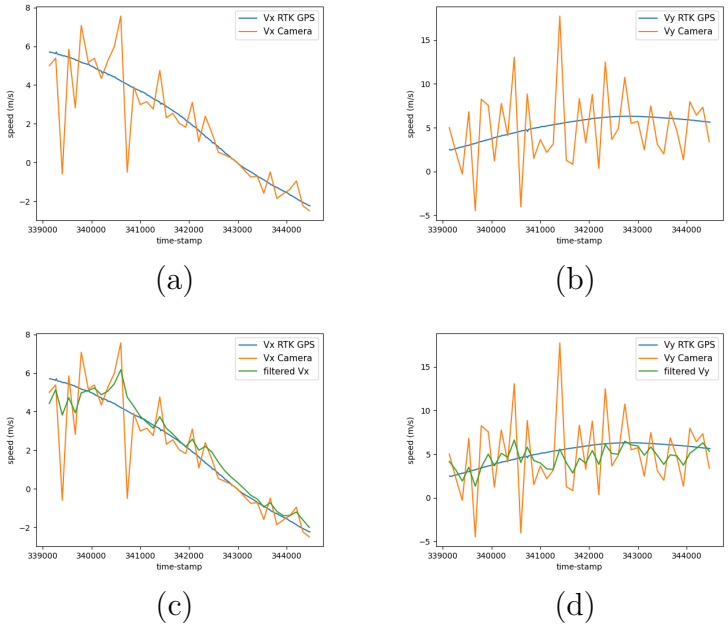


Figure 28: The raw speed components provide quite good global variation of the vehicle's speed (a) is the E axis component and (b) is the N axis component. The estimation is improved by the Kalman filter as shown in (c) and (d)

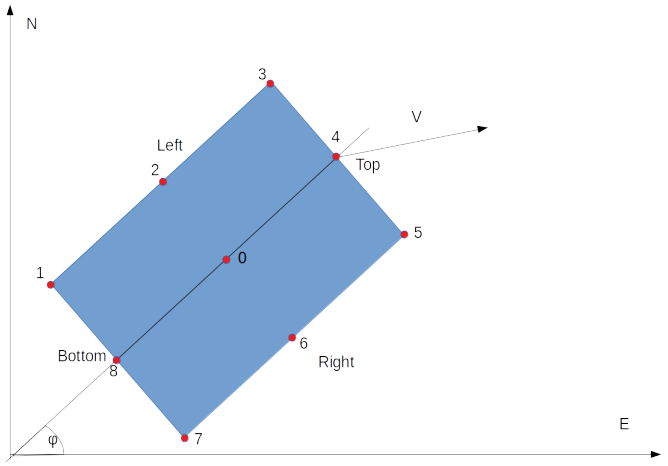


Figure 29: The object reference point.

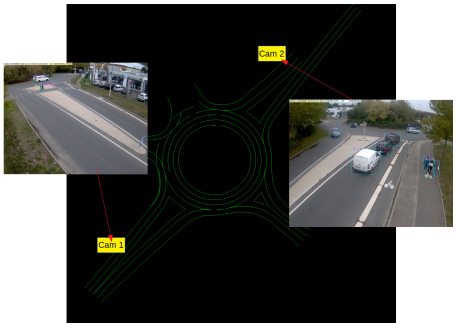


Figure 30: The experimentation site is a large roundabout. We need two perception systems to cover the roundabout. CAM1 is at 54m of the center and CAM2 is at 55m.

of the camera and finally we measure the duration from the image acquisition to the information broadcasting. The roundabout is a large infrastructure. Its central island's diameter is 26.7m. To cover entirely the roundabout, we install two systems: the first, called CAM1, is installed in the north side at 54m of the center of the roundabout and the second, CAM2, is installed in the south side at 55m of the center. The figure 30 is the illustration of the experimentation site. To evaluate the algorithms of the system, the measurements were acquired in the north of the roundabout where CAM2 is located. The instrumented vehicle made several passages in the roundabout and in the avenue which enters into this roundabout in the north side of the roundabout. The evaluation of each component of our perception system (estimation of vehicle position, estimation of the yaw angle and vehicle speed) was carried out as follows:

1. The sequences of images acquired during the experiments with the instrumented vehicle are processed by the perception system.

2. In the processed sequences, the instrumented vehicle was perfectly detected and tracked. By hand, we identified the labels of the targets corresponding to the instrumented vehicle. There are several labels, because for two different passes in the field of view of the camera, we have two different labels because the system did not re-identify targets.

3. The results concerning the instrumented vehicle are then adjusted temporally with the measurements of the RTK GPS installed inside the vehicle.

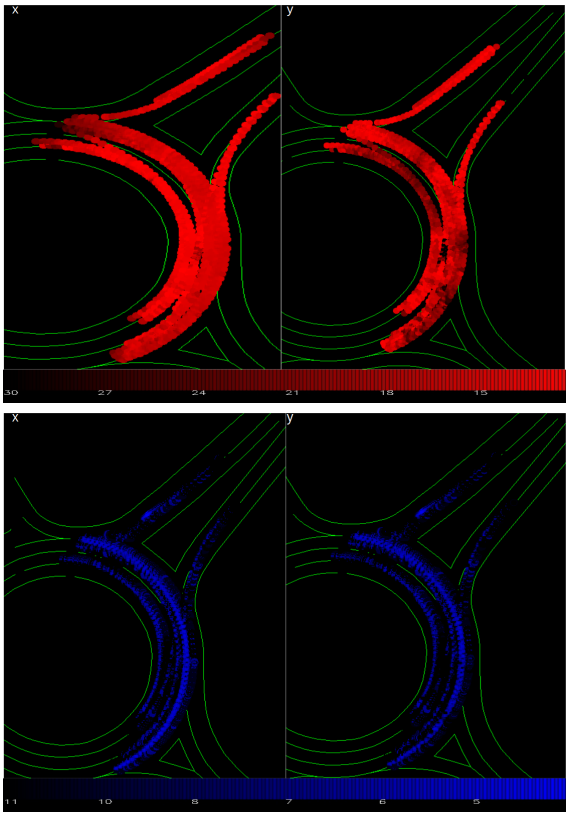
4. Finally, we calculate the mean errors between the results of the perception system with the ground truth, the standard deviation for every position of the vehicle in the roundabout.

4.4.0.1 Target's position and yaw angle

The GPS receiver installed inside the instrumented vehicle is not visible by the camera, we must estimate its position once the yaw angle and the object reference point's position are calculated. The estimated position of the GPS receiver is compared to RTK GPS measurement. The instrumented vehicle is moving in the roundabout several time and data are acquired by both the system CAM2 and the GPS receiver in the vehicle. As we are expected that the accuracy of the measurement depends on the position of the vehicle, we computed locally the accuracy mean and standard deviation in different area of the roundabout: for every position (x, y) in the field of view of the CAM2, we compute the mean and the standard deviation in the square area from $(x - \triangle, y - \triangle)$ to $(x + \triangle, y + \triangle)$. The mean position error on the Est axis and on the N axis is shown in the figure 31. As we can notice the position in East axis is more accurate than the position in the North axis. In the East axis, the mean error is close to $15cm$ to more than $1m$ in the worse cases mainly due to the occlusions of tree's leaves. And for the North axis, the measurements are less accurate. We explain this by the fact that the instrumented vehicle are often seen behind another vehicles, the y position can not be measured accurately because the BBox is a smaller. We can notice that usually the error is less than $30cm$. However, some error larger than $50cm$ are observed when the instrumented car is occluded The yaw angle provided by the perception system is in radian. In our system, the yaw angle is computed using the approach presented in the section 4.3.3. The figure 32 shows the result of the comparison between yaw angle computed by our system and the measurement provided by the instrumented car. The error of the yaw angle estimation by the camera is less than $0.4rad$ in most case except in a few localized spots near the tree or near the roundabout's exit where one can observe slowing down vehicles.

4.5 Real-time evaluation

As we already mentioned, a automatic driving application must provide information with only a very short delay. In the Tornado project, the objective is to broadcast information with delay as short as $0.1s$. During the evaluation, the processing time (Detection and tracking and CPM sending) is 0.0769 second (13hz) when only few vehicles are in the roundabout. When the roundabout is busy, the processing time is can increase to 0.09 second (11Hz). However, we notice that there is a significant delay that comes from the images acquisition. To measure the time needed to produce the image, we took images of the GPS clock and we compare the date shown in the image and the timestamp of the image provided by the camera taht is also synchronized to the GPS clock using NTP server. As it is illustrated in the figure 33, there is a difference close to 0.1 second between the time shown in the image and the timestamp by the camera. In the example shown



(a)

(b)

Figure 31: (a) The mean error of the vehicle's position depends on the position of the vehicle. We notice that the mean error is more significant in the North axis (y). (b) The standard deviation is also more significant in the same axis. For the sake readability, Te unit of the scale is the centimeter(cm) for the mean error and the decimeter(dm) for the standard deviation.

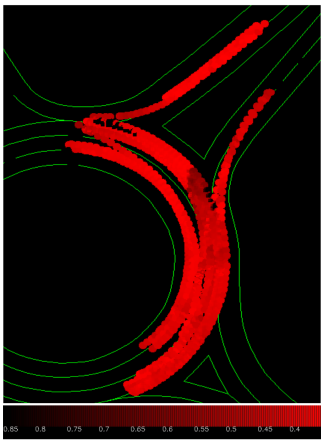


Figure 32: The error of the yaw angle estimation by the camera is less than $0.4rad$ in most case except in a few localized spots near the tree or near the roundabout exit.

in the figure 33, the timestamp is $2020 - 09 - 28at14h53m26.254s$. Thus, we are facing a hardware problem. We have to use a suitable camera with a delay much shorter than 0.1 second. Unfortunately, this issue was noticed late in our research work and we still not test other camera models to find the suitable camera for our application.

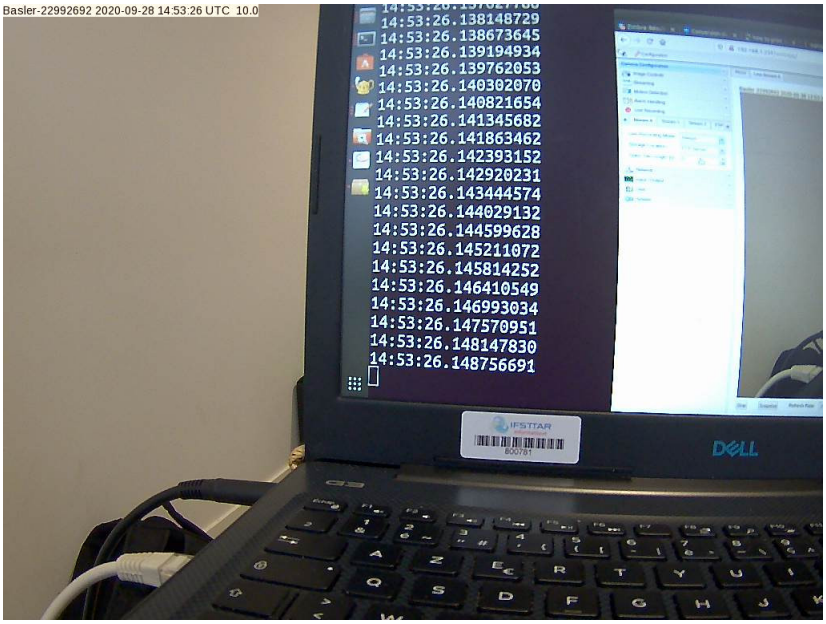


Figure 33: The error of the yaw angle estimation by the camera is less than $0.4rad$ in most case except in a few localized spots near the tree or near the roundabout exit.

5 Narrow zone passage use case

5.1 Use Case Description

The main goal of the "narrow zone passage" use case is to help the autonomous vehicle to cross a narrow zone with no visibility of the vehicles coming in the opposite direction. The use case takes place under a railway bridge on Route de Bray in Rambouillet, see Figures 34 and 35.

Figure 34: Narrow Zone - View 1

Figure 35: Narrow Zone - View 2

5.2 Equipments

5.2.1 Connected Millenium Traffic Lights

For this use case, we have used two LACROIX City Millenium Portable Traffic Lights, placed on both sides of the bridge. It is an optimal portable traffic sign solution for worksite traffic control. The Millennium worksite traffic light is entirely designed to facilitate setting up a worksite. On this Traffic Lights we have added a RoadSide Unit (RSU), which allows to send MAPEM and SPATEM messages (see paragraph 3.5 and paragraph 3.6) that announces the topology of the zone and the traffic light status (current phase and date of next phase change). This information are sent using the ITS-G5 technology, see paragraph 3.2.

5.2.2 OBU embedded in a Zoe Car

For this use case, we have also used a LACROIX City V2X On-Board Unit (OBU), embedded in a Renault Zoe vehicle. The OBU receives the MAPEM and SPATEM messages (see paragraph 3.5 and paragraph 3.6) using ITS-G5 technology (see paragraph 3.2). The content of this message is made available through an API provided by LACROIX City, based on Websocket or HTTP REST. The information of topology and traffic lights status is an input for an algorithm embedded in Renault Zoe vehicle, that allows the autonomous vehicle to decide whether or not it can cross the narrow zone passage.

Figure 36: Renault Zoe in front of a Millenium connected traffic light

6 Conclusion

This document reports the works on collaborative roadside perception system carried out in work package 6 of the Tornado project. In this project, several identified use cases need collaborative roadside perception: a narrow passage under a bridge with poor visibility, a crossing of a roundabout and a perception assistance to an autonomous shuttle in a shared space. The roadside collaborative is based on two technologies: the sensor and the V2X technology. For the V2X communication, the ETSI recommendations on the collaborative perception is used. The information are broadcast at 10Hz for CPM and 1Hz for CAM, SPATEM and MAPEM using the RSU and OBU by LACROIX City, The proposed roadside perception system is based on a camera and images are processed by YOLOV3 for the road users detection and a multi-object tracking system to track targets, to estimate yaw angle and target's speed. The perception system is evaluated on the experimentation site in the Bel-Air business area. To evaluate the perception system, an instrumented vehicle with RTK GPS passed several time on the roundabout. And ground truth are acquired with the RTK-GPS. The measurement are: the vehicle position, the vehicle yaw angle and the speed. In the section 4.4, we show that vehicle's positions are globally estimated with mean error less than $20cm$ (East axis). The few part of the roundabout with mean error more than $25cm$ up to $1m$ is due to occlusions by trees or cars (North axis). Depending on the position of the vehicle, the yaw angle is estimated with an average error ranging from $0.4rad$ to less than $0.8rad$. The vehicle's speed estimation is really difficult only using mono-camera. However, the Kalman filter improved the estimation by smoothing the values. Finally, we assess the system's ability to deliver results in real time by measuring the time from image creation to data broadcasting. The

average duration observed on the processing is $0.17s$ per image with $0.1s$ used for the creation of the image by the camera and only $0.7s$ for the target detection and the tracking.

References

[1] V2x communications message set dictionary j2735. 03 2016.

[2] Intelligent transport systems (its) vehicular communications basic set of applications part 2: Informative report for the collective perception service. 05 2018.

[3] Intelligent transport systems (its) vehicular communications basic set of applications part 2: Specification of cooperative awareness basic service. 04 2019.

[4] Intelligent transport systems (its) vehicular communications basic set of applications part 2: Facilities layer protocols and communication requirements for infrastructure services. 02 2020.

[5] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34, 01 2012.

[6] Jerome Berclaz, Francois Fleuret, and Pascal Fua. Multiple object tracking using flow linear programming. pages 1 – 8, 01 2010.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[8] Kap Luk Chan Bing Wang, Gang Wang and Li Wang. Tracklet association with online target-specific metric learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[9] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *IEEE Workshop on Motion and Video Computing*, pages 169–174, 2002.

[10] Francois Fleuret, Jérôme Berclaz, Richard Lengagne, and Pascal Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30:267–82, 03 2008.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 580–587, June 2014.

[12] Ross Girshick. Fast r-cnn. *CoRR*, abs/1504.08083, 2015.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[14] Sachiko Iwase and Hideo Saito. Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *Proceedings of the 17th International Conference on Pattern Recognition, Volume 4*, pages 751–754. IEEE Computer Society, 2004.

[15] Santosh Divvala Joseph Redmon and Ali Farhadi. You only look once: Unified, real-time object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[16] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. pages 120–127, 11 2011.

[17] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context, 2014.

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, pages 21–37, 2016.

[20] Atsushi Nakagawa, Tsuyoshi Nakano, and Yasukazu Okamoto. Demonstration experiments of driving safety support systems using vehicle-to-infrastructure communications systems. *Toshiba Review (Janpanese)*, 64(4):19–22, 20109.

[21] Hamed Pirsiavash, Deva Ramanan, and Charless Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. pages 1201 – 1208, 07 2011.

[22] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[23] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[24] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *NIPS*, pages 91–99, 2015.

[25] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[26] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[27] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017.

[28] J.K. Wolf, A.M. Viterbi, and G.S. Dixon. Finding the best set of k paths through a trellis with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, pages 287–296, 1989.

[29] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. 06 2008.

[30] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:1330 – 1334, 12 2000.