MAVEN

Managing Automated Vehicles Enhances Network

Deliverable 3.2:
Cooperative environment perception algorithms

Version	Date	Release	Approval
1.0	28.02.2019	Reza Dariani	Ondrej Pribyl

Document Log

Version	Date	Comments	Name and Organisation
01	03.12.2018	Initial version- document structure	Andreas Leich, Reza Dariani, DLR
02	25.02.2019	Contents on Cooperative Environment Perception Algorithms DLR added	Andreas Leich, DLR
03	27.02.2019	HMETC contributions on architecture and algorithms added	Michele Rondinone, HMETC
04	28.02.2019	Conclusions and Executive Summary	Andreas Leich, DLR
05	28.02.2019	Finalization	Julian Schindler, DLR

Authors

Beneficiary nº (1 to 5)	Name and Organisation	Country
1	Reza Dariani (DLR)	Germany
1	Andreas Leich (DLR)	Germany
1	Julian Schindler (DLR)	Germany
3	Michele Rondinone (HMETC)	Germany
3	Thomas Walter (HMETC)	Germany
3	Dominik Matheis (HMETC)	Germany

Table of Contents

List of Figures

Abbreviations and definitions

Abbreviation	Definition
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
CI	Covariance Intersection
CAM	Cooperative Awareness Message
CAV	Cooperative Automated Vehicle
CPM	Collective Perception Message
DLR	German Aerospace Center
DMM	Decision Making Module
FCI	Fast Covariance Intersection
GNC	Guidance, Navigation and Control
GNSS	Global Navigation Satellite System
HMETC	Hyundai Motor Europe Technical Center
I-FCI	Improved Fast Covariance Intersection
IMU	Intertial Measurement Unit
LAM	Lane Advice Message
LIDAR	Laser Imaging, Detection and Ranging
MAP	V2X message containing map data
MP	Motion Planner
OBU	On-Board Unit
PDC	Park Distance Control
PP	Path Planner
ROS	Robot Operating System
RSU	Road Side Unit
SPAN	Synchronous Position, Attitude and Navigation
SPAT	Signal Phase and Timing
VC	Vehicle Control
V2X	Vehicle to Any (e.g. to Vehicle, Infrastructure)

	communication
VRU	Vulnerable Road User

Executive summary

This document describes the project achievements in cooperative perception algorithm development and testing.
An overview of the building blocks, vehicle setups and embeddings of the perception algorithms into the test vehicle architectures used within the experiments at DLR and HMETC is given. A review of the state-of-the-art outlines the different levels of sophistication that are known from literature for the association and data fusion steps of a cooperative environment perception algorithm and reflects recent developments in the field. Real time algorithms are explained that were used in the experiments and demonstrations at the HMETC test site with both HMETC and DLR vehicles involved. Fusion experiments demonstrate detection and perception of a pedestrian not in sight of the ego vehicle sensors via a collective perception (CPM) message and combining of position estimates received via a cooperative awareness message (CAM) with the sensor detections of a test vehicle.
Based on the results of the state-of-the-art review, an advanced fusion algorithm is proposed and its performance in urban traffic scenarios as well as Monte Carlo simulation scenarios evaluated. Conclusions are drawn for cooperative perception algorithm development and testing.
This deliverable serves (in combination with D3.3) as basis for the testing done in WP6 with its upcoming deliverable D6.4 and the evaluation done in WP7 with its upcoming deliverable D7.2.

The developments, experiments and evaluations presented in this deliverable have been executed by the MAVEN project partners HMETC and DLR in tight cooperation.

1 Introduction

Highly and fully automated vehicles, especially when connected to the C-ITS infrastructure, can significantly contribute to meeting the EU objective of effectively accommodating growing mobility demands while still ensuring lower environmental impacts and increased road safety (Horizon 2020 Work Programme 2016-2017 - Smart, green and integrated transport, 2019).

In this context, the MAVEN project (Managing Automated Vehicles Enhances Network) will deliver C-ITS-assisted solutions for managing Cooperative Automated Vehicles (CAVs) at signalised intersections and intersection corridors with the aim of increasing traffic efficiency and safety. A key technology for increasing safety is cooperative perception. Cooperative perception means, that a CAV can utilize information that is provided from a road side unit (RSU) or from another CAV for improving its own model of the environment. As a result, a gain in precision and anticipation can be achieved and the abilities of assisted and highly automated driving systems can be improved. Especially information about Vulnerable Road Users (VRUs, e.g. pedestrians and/or cyclists) and/or other obstacles will be used to better drive vehicle manoeuvre algorithms. Extending advanced driver assistance systems (ADAS) functionalities with the cooperative approach is one of the core objectives envisioned by MAVEN.

1.1 Purpose of this document

This document describes in detail the MAVEN developments, experiments and findings in the field of cooperative collective perception. This document contains:

- an overview of the architecture of the in-vehicle architecture used for cooperative perception experiments,
- a literature review on the state-of-the-art in the data association and fusion domain,
- a description of the on-line algorithms used for real-time/real-world experiments,
- a presentation of the advanced algorithms that were only tested offline on recorded sensor data,
- test and evaluation results of real-time/real-world experiments, and
- test and evaluation results of simulation experiments.

This report presents different algorithmic approaches to implement the cooperative perception. These approaches vary in their level of computational complexity and sophistication. The properties and performance of the algorithms are outlined and compared and conclusions are drawn.

The work described in this deliverable serves as basis for the implementations and testing described in upcoming deliverables D6.4 and D7.2.

1.2 Document structure

The remainder of this document is organized as follows:

Section 2 gives an overview of the building blocks, vehicle setups and embeddings of the perception algorithms into the test vehicle architectures used within the experiments at DLR and HMETC.

Section 3.2 contains a review of the state-of-the-art. It outlines the different levels of sophistication that are known from literature for the association and data fusion steps of a cooperative environment perception algorithm, reflects recent developments in the field.
Based in the results of the review of the state-of-the-art an advanced fusion algorithm is proposed in section 3.3.1. The performance of this algorithm is evaluated in section 3.3.2.
Section 3.4 presents the real-time algorithms that were used in the experiments and demonstrations at the HMETC test site with both HMETC and DLR vehicles involved. Fusion experiments demonstrate perception of a hidden pedestrian via a CPM-message and fusion of position estimates of DLR’s ViewCar2 vehicle with the sensor detections of HMETC’s test vehicle. The ego position of ViewCar2 is transmitted via a CPM message.
This deliverable closes with conclusions in section 4.

2 In-Vehicle architecture

2.1 DLR

At DLR, two automated vehicles were used for the cooperative perception experiments in MAVEN: FASCarE and ViewCar2. FASCarE is a VW e-Golf series car, ViewCar2 is a VW Passat GTE Plug-In Hybrid series vehicle. Both test vehicles are equipped with sensors and actors that are under full control of the DLR research facility's developers.

The FASCarE sensor equipment includes (Figure 2):

- SMS Radar UMRR-P-0801
- Localization unit Novatel SPAN-CPT with inertial measurement unit (IMU) and global navigation unit (GNSS)
- 5 Ibeo Scala Laserscanners integrated into the vehicle body
- Cohda V2X communication unit
- Linkbird communication unit
- Vehicle series equipment:
  - Ultrasonic Park Distance Control (PDC) sensors
  - lane keeping assistance front camera
  - long range Adaptive Cruise Control (ACC) Radar

The ViewCar2 sensor equipment includes:

- BOSCH MRR4 long range ACC Radar
- Hella Compact Radar A0 short range Radar
- Novatel GNSS 800 with IMU and wheel odometry
- 6 Ibeo Scala Laserscanners integrated into the vehicle body (Figure 1)

The software system of both vehicles consists of three parts:

- dSPACE vehicle interface (CAN-UDP)
- DLR self-tailored Dominion middleware
- ROS (short for Robot Operating System)

The sensors and sensor fusion functions are implemented as ROS nodes. The cooperative perception experiments within MAVEN are explained in more detail in section 3.3.2. For these experiments, only the LIDAR vehicle sensors were used, as shown in the building blocks diagram in Figure 4, which shows all the steps from sensor to objects to object fusion to vehicle automation and V2X transmission. The algorithms presented in section 1 can, however, be applied to all sensors within both vehicles. Cooperative perception seamlessly fits in the general multi-sensor fusion pipeline of the vehicle.
Figure 3, taken from D3.1 (Dariani, et al., 2018), shows the embedding of the sensor data fusion part into the overall DLR vehicle software architecture for the MAVEN project.

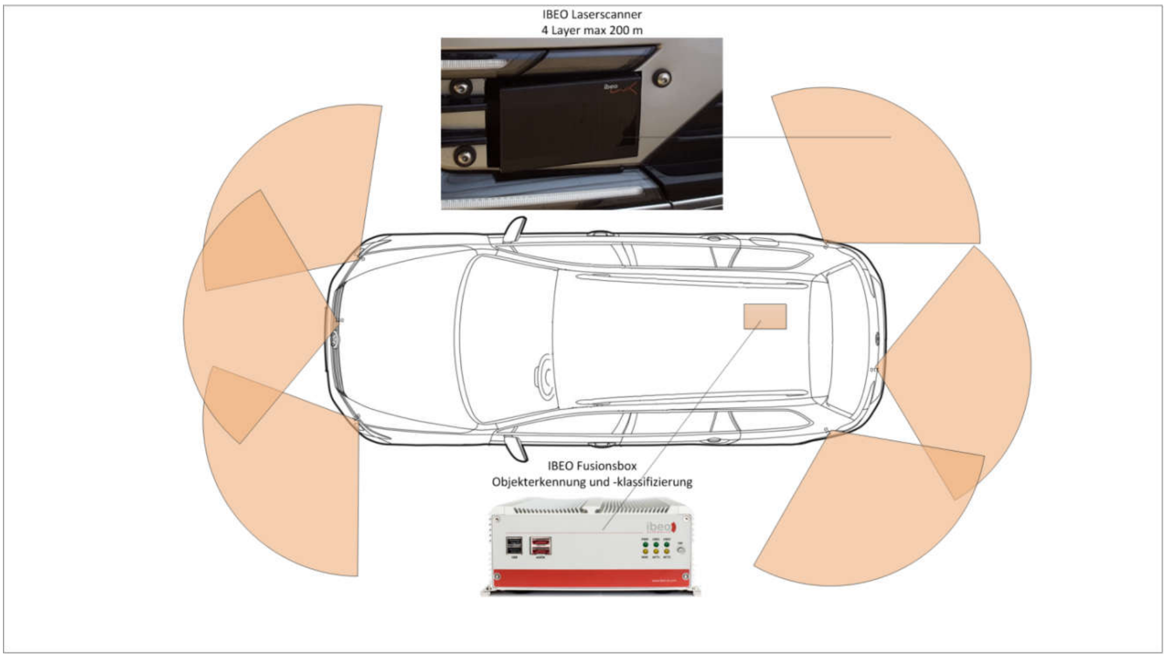

Figure 1: LIDAR sensor equipment of the ViewCar2 automated vehicle development platform

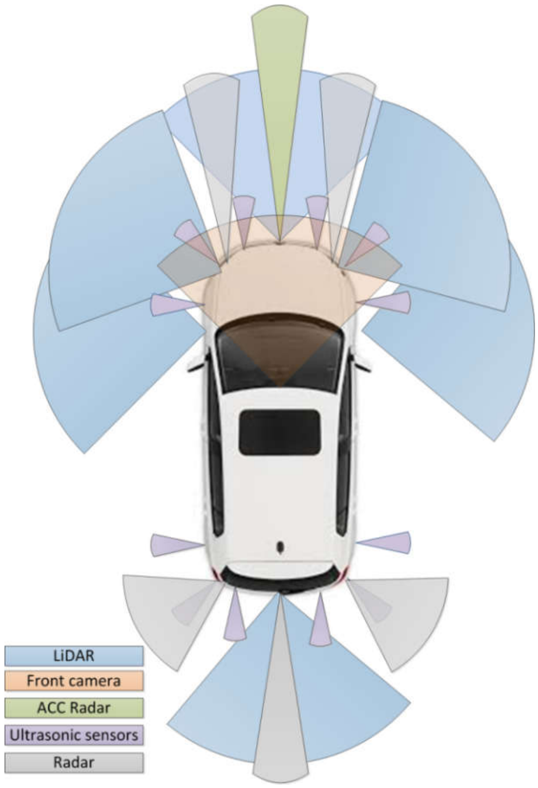

Figure 2: Overview of the sensor equipment of the FASCarE automated vehicle development platform

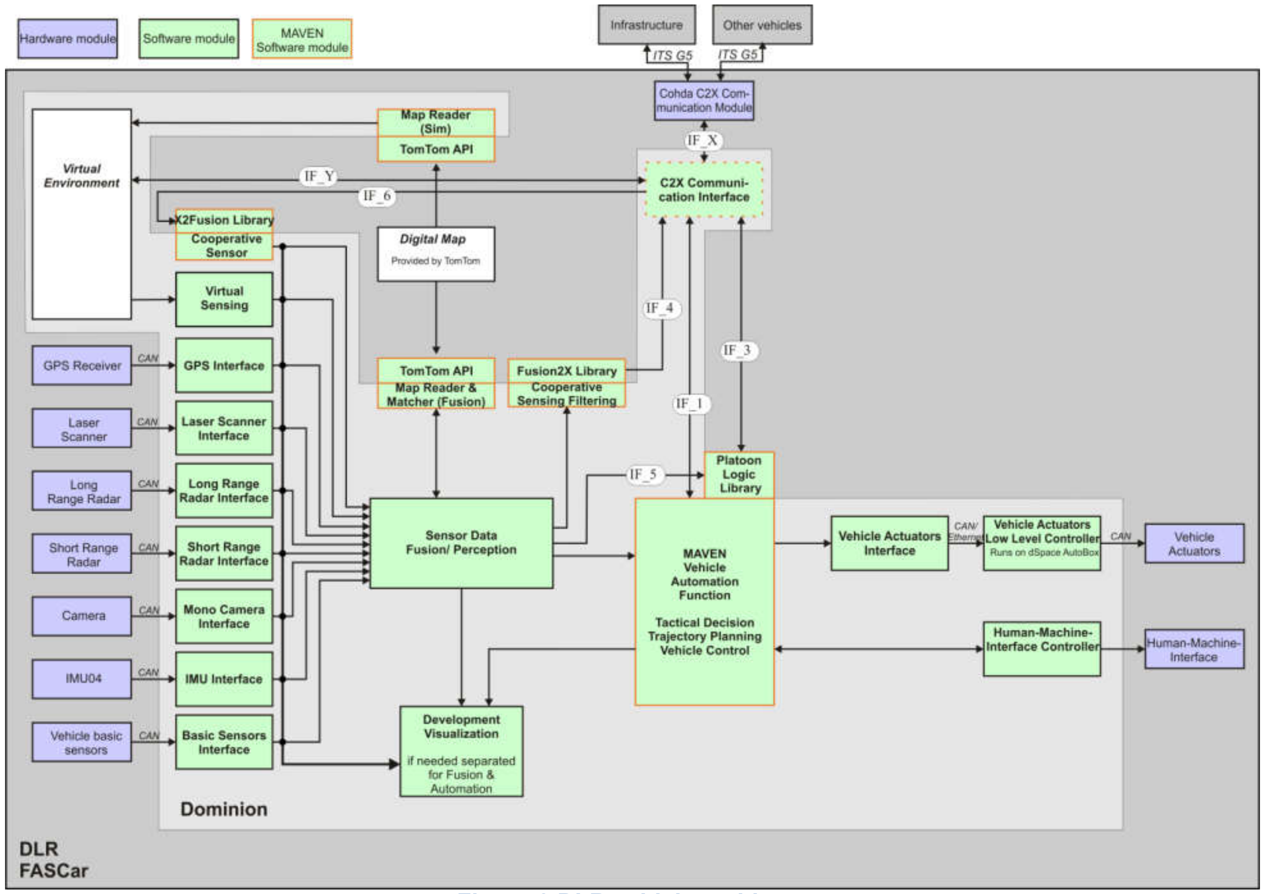

Figure 3 DLR vehicle architecture

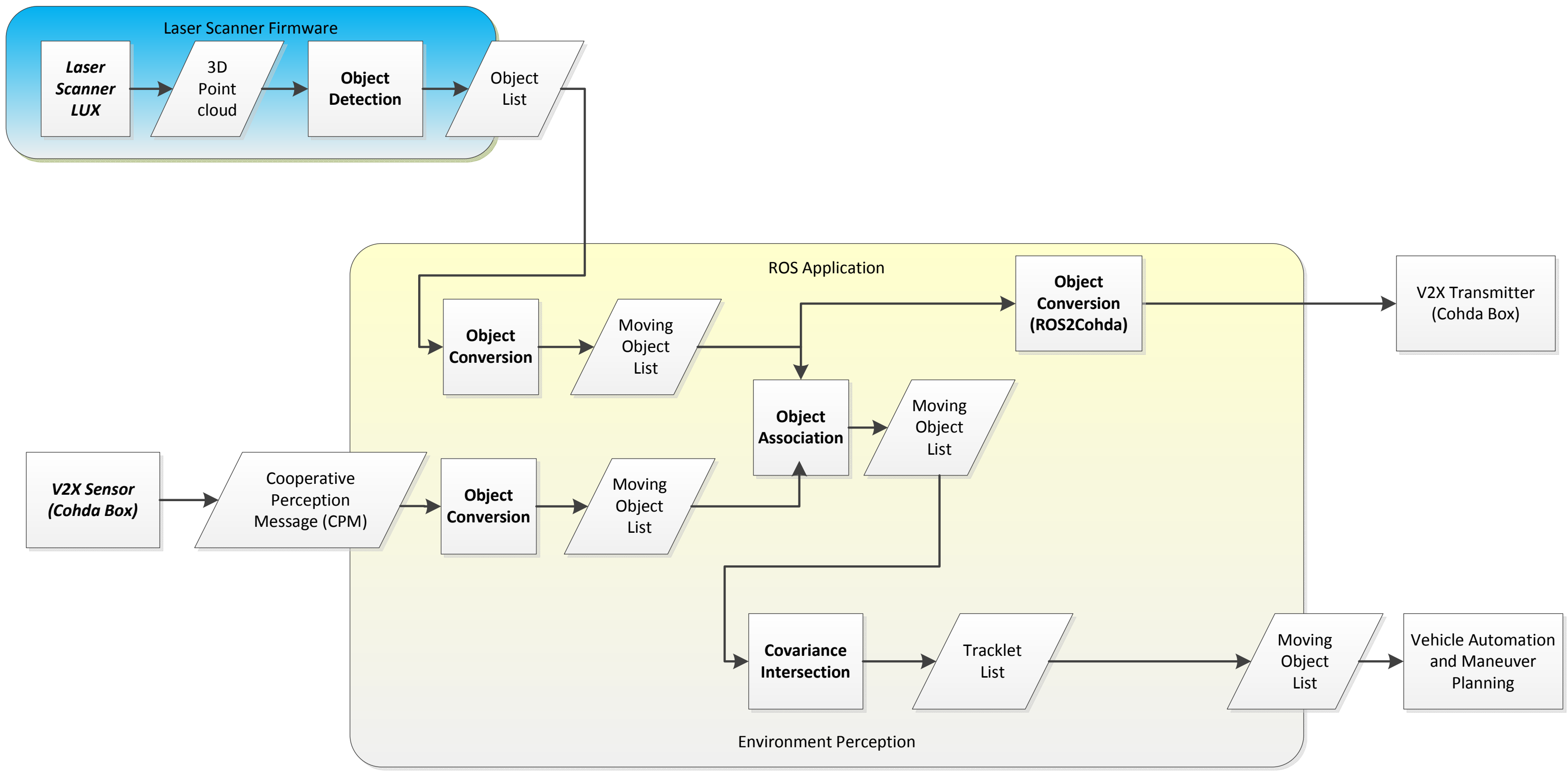

Figure 4: Object level fusion building blocks

2.2 HMETC

The Hyundai vehicle used for MAVEN is a series Ioniq vehicle enhanced with sensors and computing equipment that are not experimental samples but either already installed in series vehicles, or are close to be. Hyundai's main goal in MAVEN is the research of cost effective solutions for automated driving vehicle architecture. This means for example the usage of series or near-series components and the integration of required computing platforms to run the MAVEN control logic. Figure 5 shows Hyundai's Software architecture for the MAVEN project which doesn't differ much from DLR's approach. This is for two reasons:

1. Enabling compatibility to perform MAVEN use cases (especially by using the same set of V2X communication messages)
2. Using common software architecture shared and used by several OEMs, research institutes and universities (common also in the robotic sector) → "sense, act and control". (See Figure 6)

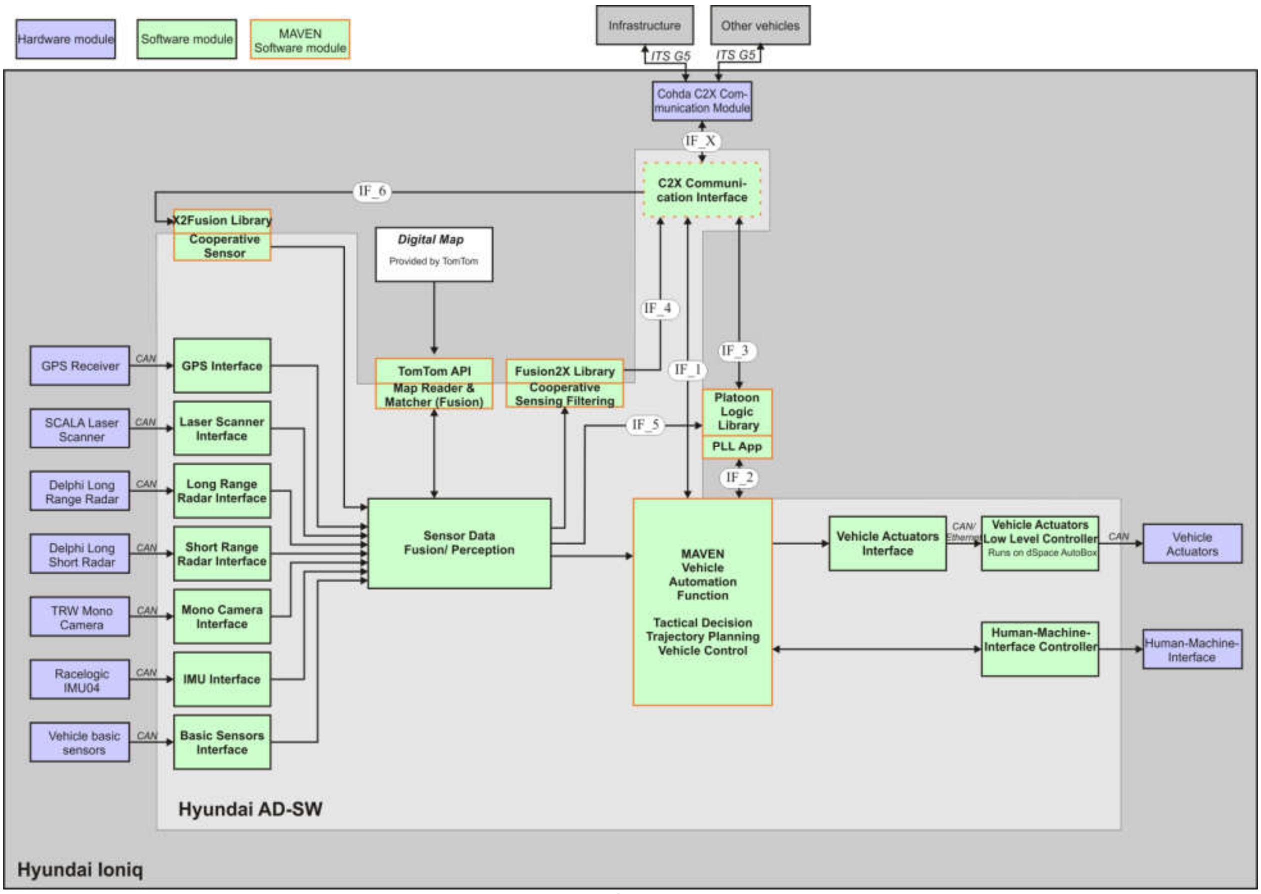

Figure 5: HMETC vehicle architecture

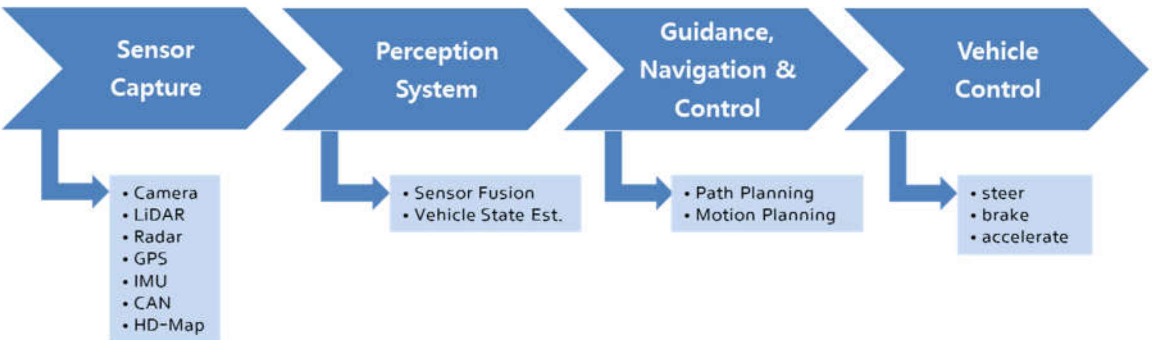

Figure 6: HMETC data processing approach "sense, act & control"

Figure 5 illustrates the vehicle architecture of the HMETC vehicle. The architecture contains different sensors, software and interfaces. Some of the main modules are explained briefly.

- AD_SW: Hyundai's test vehicle framework running the MAVEN functions is based on the commonly used ROS which is taking care of the communication between several so called nodes (sending and/or receiving endpoints/control units/functions), scheduling and maintenance tasks. On top of this base framework, control logic and specific MAVEN functions are integrated to fulfil the required MAVEN use cases like manoeuvring control upon road infrastructure advisories, handling of collective perception information or platooning.

- Sensors: the following On-board sensors are used (see Figure Figure 7):

    - 1x Ibeo Scala front + 1x Ibeo rear LiDAR,
    - 1X TRW/Mobileye front Camera,
    - 4 x Delphi/Aptiv SRR4 corner radars,
    - 1 x Delphi/Aptiv MRR3 front radar,
    - 1 x Oxford OXTS DGPS.

  Also, the following Cooperative Sensors are used:

    - Cohda MK5 OBU (V2X communication module)

Figure 7: HMETC vehicle sensor setup

- Sensor Data Fusion / Perception: This module receives the information from different sensors, information from the Highly Automated Driving (HAD) map and also information collected via V2X communication. Then it fuses the information and delivers the fused information to vehicle automation.

- V2X communication interface: This module is used to transmit and receive the required information needed to improve the environmental perception and run platooning algorithms (such as CAM messages and CPM messages), as well to receive information from the infrastructure needed for I2V advices (SPAT/MAP and LAMs).

- Vehicle automation (also called Guidance, Navigation and Control module (GNC) module in the following): This module consists of various sub-modules: Decision Making Module (DMM), Path Planner (PP), Motion Planner (MP), and Vehicle Control (VC). The DMM analyses the intender route, the outputs of the sensor fusion module about the vehicle surrounding and the information received via V2Xcommunication. It uses this information to take decisions regarding possible manoeuvres and threats as inputs to the path planner based on some priority mechanisms. Based on these, the path planner plans a trajectory and provides inputs for the motion planning. Vehicle control consists of several feedback and feedforward controllers to guarantee that vehicle follows the planned motion.

- Platoon logic: The same platoon logic implementation as for the DLR car is used.

In section 3.4, the HMETC sensor data fusion module will be explained in more detail to highlight the cooperative environmental perception algorithms adopted in MAVEN.

3 Cooperative perception algorithms

3.1 Introduction

This section presents the results of the implementation and test of algorithms for information fusion in the context of cooperative perception. Cooperative perception means that the environment model derived from in-vehicle sensors is combined with object detections that the vehicle receives from other road users from a road side unit, as shown in Figure 4. The standardized message types include the cooperative perception message (CPM) and the cooperative awareness message (CAM). While CAM messages are transmitted by other road users only, CPM messages can be received from road side units as well. A CPM message was introduced by the project Ko-FAS (Forschungsinitiative Ko-FAS. Ko-PER - Kooperative Perzeption.). The CPM is by now closed to be standardized and contains information on:

- the ego-position of the sender entity,
- its sensors and their field of view, and
- the objects detected, including their state parameters and confidence values.

Cooperative perception algorithms solve the task of combining the model of the environment derived from the in-vehicle sensors with the information received via V2X so that a consolidated environment model is computed. The consolidated environment model can have improved data quality with regard to the following criteria:

1) the detection range and field of view of the vehicle is extended,
2) the confidence of detected objects in the consolidated environment model is higher,
3) the precision of the state variable measured is higher, measurement errors are averaged out or identified and corrected.

A cooperative perception algorithm solves a task that is very similar to a multisensory fusion algorithm. For an overview of multisensory fusion algorithms in automotive environments, see (Houenou, Bonnifait, Cherfaoui, & Boissou, 2012). A multisensory fusion algorithm is designed for the slightly different case that the fields of view of the different sensors have more overlap and are located on the same vehicle. The only difference is that a cooperative perception algorithm must be able to process sensor measurements for sensors with different or even disjunct fields of view which are located at different positions and move in different directions with different speed.

The building blocks of a multisensory fusion algorithm are:

a) detection,
b) association, and
c) tracking,

while it is reasonable to further distinguish:

d) a temporal alignment phase, and
e) a spatial transformation procedure (Seeliger & Dietmayer, Inter-vehicle information-fusion with shared perception information, 2014)

before the association is executed. This is caused by the fact that the detections come from a different time frame and from sensors at different locations. The following sections present algorithms implementation and testing within the domains b), c) and e) (from the list on previous page). Temporal alignment procedures were not investigated and are subject of further research.

3.2 Related work

In (Houenou, Bonnifait, Cherfaoui, & Boissou, 2012), a complete algorithm for multi-sensor data fusion is presented. Its architecture consists of a sensor layer and a fusion layer, which is inspired by the earlier work of (Darms, Rybski, Baker, & Urmson, 2009). The sensor layer includes feature extraction (Detection), spatial and temporal alignment and data validation. The fusion layer contains track to track association and track merging. Track merging is performed by a fusion operator that is derived from the Kalman Filter equations in (Smith & Cheeseman, 1986). It consists of equations for computing the output state vector X and its covariance matrix P of the tracked object in the merged track. Equations (1) and (2) are a fusion operator that can be applied for pairwise fusion of tracks a and b. The operator can be applied repeatedly if a cluster of more than two tracks needs to be processed.

$$X = P_b(P_a + P_b)^{-1}X_a + P_a(P_a + P_b)^{-1}X_b \quad (1)$$

$$P = P_b(P_a + P_b)^{-1}P_a \quad (2)$$

For association, (Houenou, Bonnifait, Cherfaoui, & Boissou, 2012) use a distance metrics introduced in (Müller, Meuter, Ghosh, & Müller-Schneiders, 2011), that accounts as well for the current state as for the history of the trajectories by averaging (equation (3)) the distance, $D_{a,b}$, of the measurements (equation (4)) for the last n observations.

$$D_{a,b}(k) = \frac{1}{n}\sum_{i=0}^{n-1} d_{a,b}\,(k - i) \quad (3)$$

$$d_{a,b}(k) = \Delta X^T(P_a(k) + P_b(k))^{-1}\Delta X_k \quad (4)$$

The distance metrics (4) is very similar to the square of the sum of the Mahalanobis-distances between the states X^a and X^b and the probability distribution of a random variable equal to their sum. It is therefore a reasonable distance metrics. (Houenou, Bonnifait, Cherfaoui, & Boissou, 2012) extend the distance measurement formulation by (Müller, Meuter, Ghosh, & Müller-Schneiders, 2011) while introducing an added term $\ln(|P_a(k) + P_b(k)|)$. This term penalizes trajectories with high uncertainty that otherwise could be randomly fused by chance causing clutter.

While the algorithm of (Houenou, Bonnifait, Cherfaoui, & Boissou, 2012) relies on position and velocity measurements, (Duraisamy, Schwarz, & Wöhler, 2015) introduce a slightly extended approach that utilizes additional non-kinematic information such as target width and road user class (vehicle, pedestrian, truck) into the distance calculation. This improved the results. The work of (Duraisamy, Schwarz, & Wöhler, 2015) was not intended for cooperative perception, but can be applied to it.

In (Sawade, Schäufele, Buttgereit, & Radusch, 2014) a cooperative blind spot detection system is presented where the field of view of a given blind spot detection system is extended with object detections received via V2X. For extending the field of view of the in-vehicle sensor, no sophisticated algorithms are required, as long as the detection ranges of the ego-vehicle sensors and the remote V2X sensors to not intersect or at least do not give contradicting detections.

Equation (2) computes the estimate of the covariance matrix of the track that is merged from two other tracks. In (Seelinger & Dietmayer, 2014) a more sophisticated approach is applied to cooperative perception: covariance intersection (Julier & Uhlmann, 1997). Equation (2) is only correct in the situation that the observations of the different data sources are independent, which is not necessarily true. Therefore, an algorithm that uses equations (1) and (2) is called Simple Convex Combination (SCC). The Covariance Intersection Algorithm is more general in that it gives correct estimates when there is, to an unknown extent, cross correlation between the measurements a and b. (Julier & Uhlmann, 1997) propose a fusion algorithm, where the covariances of the different measurements are weighted using a variable $w \in [0,1]$. In the convex combination algorithm, equations (5) and (6) are used instead of (1) and (2).

$$X = P\left(wP_a^{-1}X_a + (1-w)P_b^{-1}X_b\right) \tag{5}$$

$$P^{-1} = wP_a^{-1} + (1-w)P_b^{-1} \tag{6}$$

The idea of the convex combination algorithm is to search for an optimal value for w with regard to some optimization criterion, such as minimizing the trace of the resulting covariance matrix P or the determinant of P.

In (Seeliger, Fahrzeugübergreifende Informationsfusion für ein Kreuzungsassistenzsystem, 2017), different possible algorithms for covariance intersection are studied in the context of cooperative perception for road users. According to Seeliger, Simple Convex Combination (SCC) and Fast Covariance Intersection (FCI) are the practically relevant options for the case that apart of the state vector and its covariance no additional information exists. Since SCC is reported to give inconsistent results and to be error prone, FCI was selected for implementation. The following section presents details of the implementation.

Fast covariance intersection was first proposed by (Niehsen, 2002). It gives a good estimate of the optimal w at low computational cost.

$$w_{FCI} = \frac{\det(P_2)}{\det(P_1) + \det(P_2)} \tag{7}$$

This formula gives erroneous results in special conditions, especially, when $(P_1 - P_2) \approx 0$ or $|P_1 - P_2| \gg 0$. Therefore, (Franken & Huepper, 2005) propose an improved version (I-FCI), that proved to handle situations well, where FCI fails.

$$w_{I-FCI} = \frac{\det(P_2^{-1} + P_1^{-1}) - \det(P_2^{-1}) + \det(P_1^{-1})}{2\det(P_1 + P_2)} \tag{8}$$

In Figure 8, an example for covariance intersection is presented. Vehicle1 is equipped with ACC. The ACC radar (yellow) gives good distance and velocity measurements, but has high uncertainty in lateral direction. The camera (black) gives good lateral position estimates, but has higher uncertainty in longitudinal direction because of the impact of perspective. As a consequence, the two error ellipses, corresponding to the covariance matrices P_1 and P_2 have different shape. Properly combined, the combined covariance P gives a more accurate estimate of the vehicle's true position.

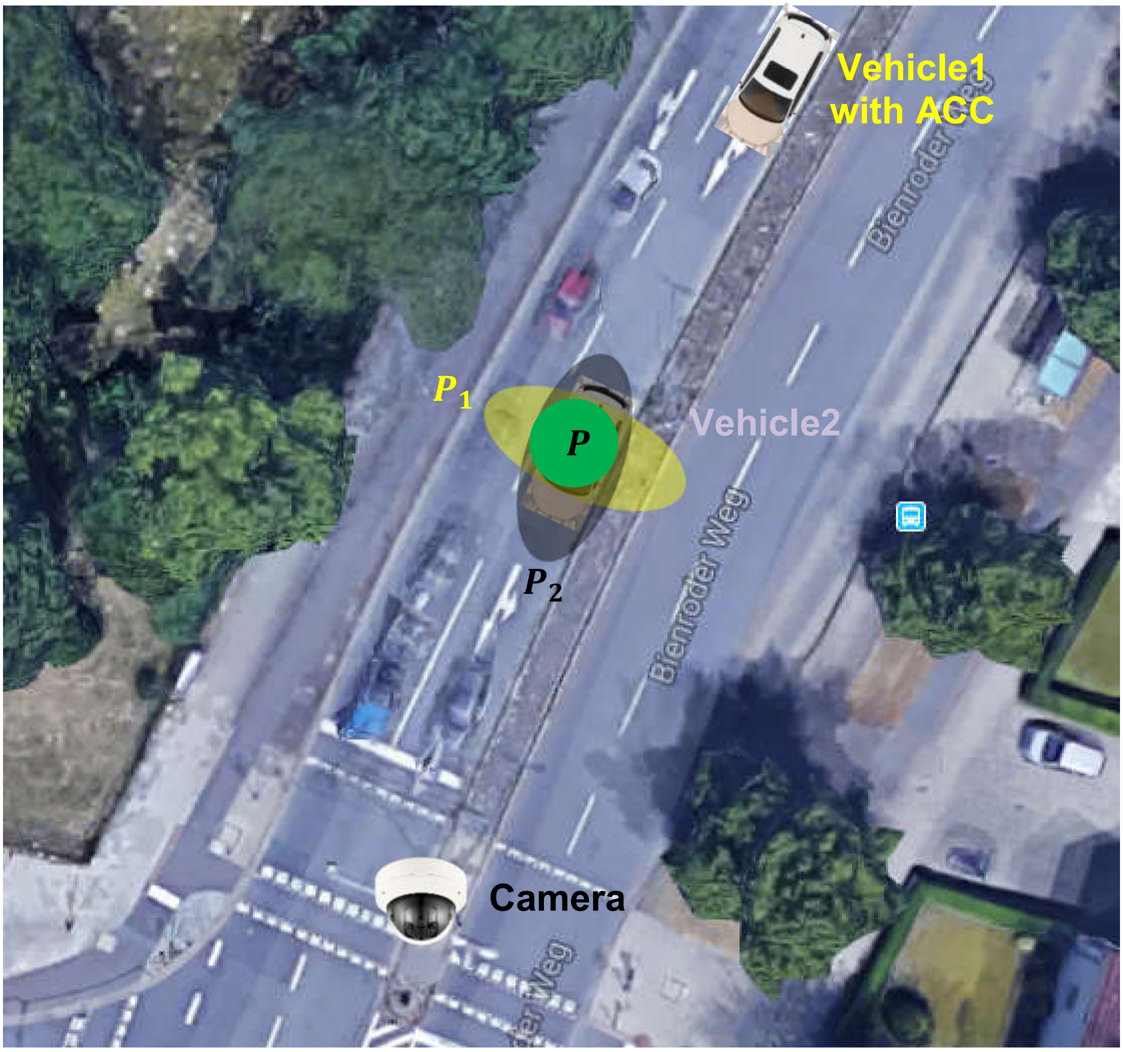

Figure 8: Covariance intersection example

3.3 DLR

3.3.1. Implementation of Association and Fusion

This section gives an outline of the association algorithm implemented for cooperative perception at DLR. More details on algorithm implementation can be found in (Steuernagel, 2019). The software building blocks scheme in Figure 4 illustrates how the V2X fusion algorithm is integrated into the vehicles FASCarE and ViewCar2.

The association algorithm is mostly based on the approach proposed by (Houenou, Bonnifait, Cherfaoui, & Boissou, 2012). Consequently, Equation (4) is extended by a term that penalizes trajectories with high uncertainty.

$$d_{a,b}(k) = \Delta X^T\big(P_a(k) + P_b(k)\big)^{-1}\Delta X_k + \ln(|P_a(k) + P_b(k)|) \tag{9}$$

The distances are arranged in a matrix M, where

$$M_{i,j} = \begin{cases} MaxVal & if\ i = j \\ MaxVal & if\ sensor(i) = sensor(j) \\ MaxVal & if\ D_{i,j}(k) > G \\ D_{a,b}(k) & else \end{cases} \tag{10}$$

The function $sensor(x)$ gives a unique index for every sensor involved in the process. The gating G is the largest possible distance allowed for joining two tracks and $MaxVal$ denotes a number that is greater than any potential measurement. The constraints raised in equation (10) assure that

- track self-associations receive $MaxVal$ cost and do not influence the overall result,
- track associations of the same sensor are penalized with $MaxVal,$ and
- tracks cannot be associated when their distance is above the gating threshold.

In the clustering step, the following procedure is executed repeatedly, to compute a set of clusters where each cluster corresponds to a real world object:

1. select the smallest value from M
2. if the smallest value is $MaxValue$, terminate the procedure
3. add the corresponding tracks, characterized by the indices i, j a cluster. This can either
   a. make a new cluster with both tracks added or
   b. make an update to an existing cluster if one of the tracks i, j is already included in a cluster
4. set all cells in column i and row j to $MaxVal$

In contrast to the nearest neighbor approach outlined in Figure 15, the more generalized approach by (Houenou, Bonnifait, Cherfaoui, & Boissou, 2012) gives better results in a real world scenario such as a crowded urban intersection. However, this more sophisticated association algorithm requires higher computational cost.

If N_{track} is the number of tracks, the computational complexity of the association algorithm is $O(N_{track}^3)$. Therefore, the number of objects is essential for real-time operation. Filtering out static objects and clutter from the laser scan becomes a plays an important role in this context.

3.3.2. Results of Association and Fusion experiments

The output of the self-tailored laser-tracking algorithm by DLR contains a high number of static objects. Two techniques were implemented to reduce the number of objects:

- map based filtering, and
- velocity based filtering (see Figure 10).

Both techniques were implemented and help to reduce the required computation time. The association algorithm itself was implemented and tested on recorded data in ROSBAG format. Target positions between measurements were estimated using a constant velocity model. This was necessary because the update rates of V2X messages were 1…10Hz while the laser scanner updates had an update frequency of 25Hz. For spatial alignment, the built in *tf* package of ROS “that lets the user keep track of multiple coordinate frames over time” was used (ROS.org - tf, 2019).

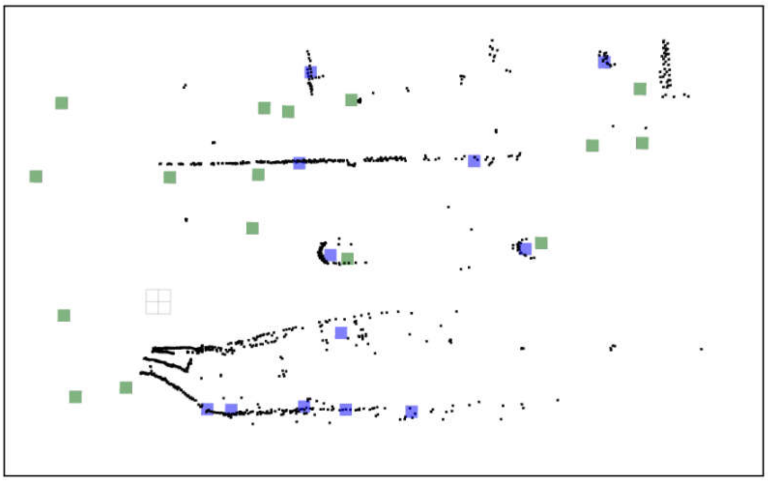

Figure 9: Overlay of the Laserscans and the detected targets for FASCarE (blue) and ViewCar2 (green).

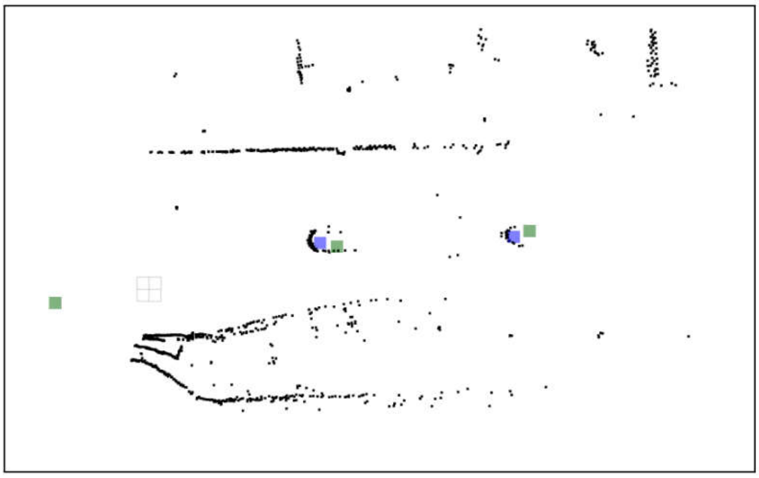

Figure 10: Scene from Figure 9 after filtering out all objects with v<5 km/h

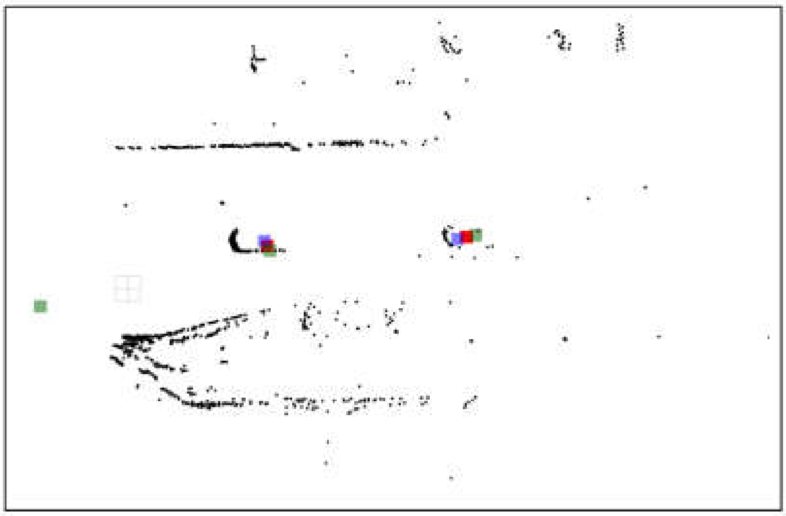

Figure 11: Covariance intersection fusion (red) result for the scene in Figure 10

Testing of a new version of the laser scanner manufacturer's (Ibeo) tracking firmware showed, that the results of the new firmware version outperform the self-tailored tracking software. Therefore, an interface with the new built in detection and tracking software was implemented and integrated into the environment perception pipeline of ViewCar2. An update of FASCarE was not possible, because this would have required upgrading the whole hardware including the Ibeo sensors and Ibeo fusion ECU which was not possible by the time this report was issued. Therefore, only the ViewCar2 was collecting laser scanner data in the complex urban traffic at the Tostmannplatz area, where also all other test runs of the urban platooning and the AGLOSA tests done in the MAVEN project are taking place (see (Schindler J. , Blokpoel, Rondinone, Walter, & Přibyl, 2018), (Schindler, Dariani, Rondinone, & Walter, Dynamic and Flexible Platooning in Urban Areas, 2018), and (Schindler, Dariani, Rondinone, & Walter, Implementation and Testing of Dynamic and Flexible Platoons in Urban Areas, 2019)). Further experiments with two vehicles that deliver CPM messages and experiments with CPMs transmitted by a road side unit at Tostmannplatz will give more insight in the potential improvements through the use of covariance intersection.

To analyse the effect of the history length $N_{history}$ were carried out with real data, in order to investigate the tradeoff between computational complexity and quality. It was found, that simple nearest neighbor association only incorporating the current state sometimes fails in real situations.

Monte Carlo simulation runs were performed to evaluate the added value of the both covariance intersection algorithms FCI and I-FCI. It was found, that a reduction of the RMSE by 9% to 30%, compared with the precision of the more accurate of both sensors, could be achieved by fusion. This demonstrates that the cooperative approach has high potential to improve environment perception performance.

The results of the tests were as follows:

- There is a partial lack of covariance estimates from the DLR self-tailored laser tracking code. In case covariance estimates are missing, the solution of equations (7) and (8) is $w_{FCI} = w_{I-FCI} = 0{,}5$. When possible, covariance intersection did significantly improve the precision of the state vector.
- The computation time of the association algorithm was near real time but still too high on the used Intel Core i5-6600 CPU. Therefore, the algorithm could be used offline only.
- I-FCI, in all cases, computed better results than FCI. This and the fact that FCI is less stable, leads to the clear recommendation to use I-FCI instead of FCI.
- Covariance intersection gives higher yield in a situation where two precise sensor measurements are fused. With higher uncertainties in the sensor measurements, the precision gain tends to be less.
- In a controlled environment where the traffic situation does not become too crowded and complex and errors keep being small, using a history does not pay. However, in complex urban scenarios, with a high density of detected targets and possible ego-positioning errors, as shown in Figure 12, mismatches do occur. It can be concluded, that accounting for the history guarantees safer operation of the algorithm.

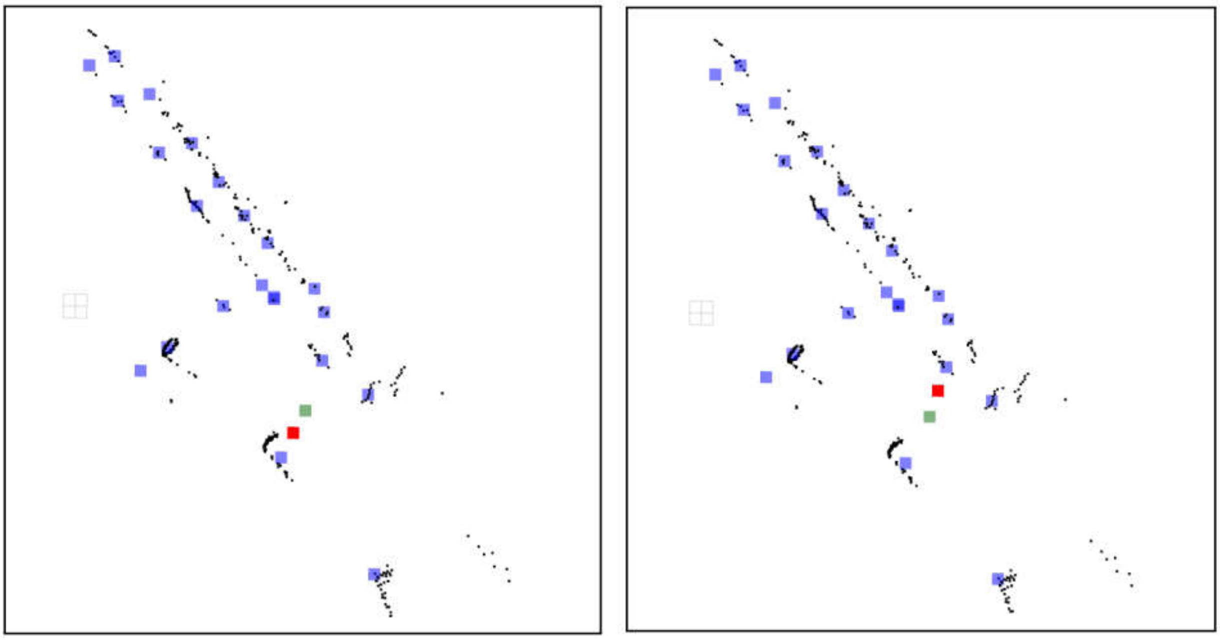

Figure 12: Association error when fusing objects detected by ViewCar2 (blue) with the ego position of FASCarE received via a CAM message (green). Left: correct association with a history length of 2s. The fusion result (red) corrects the ego-positioning error in the CAM message. Right: incorrect association with a history length of 0s. In this case, fusion (red) completely spoils the estimation results.

3.4 HMETC

3.4.1. Sensor interfaces and environmental perception overview

Environmental perception sensors like Radar, LiDAR and front camera are used for object detection and classification as well as for vehicle localization/positioning tasks ((D)GPS receiver and IMU). The sensors used by HMETC are "smart" in the sense that they offer outputs of pre-processed sensor data (e.g. object detection, classification and tracking). More specifically, dedicated ECUs (Electronic Control Unit) perform a data fusion of data captured respectively by single LiDAR or Radar units, as well as an object tracking (360° around the vehicle). Pre-processed data (classified objects) are fed into the sensor data fusion / perception module to combine objects detections from different sources with vehicle dynamics sensor data (vehicle velocity, acceleration, yaw rate). The data set is completed with V2X inputs (extracted from received CAM and CPM messages), positioning information generated by (D)GPS receivers and an Highly Automated Driving (HAD) map database. Sensor data capturing and data conversion is done by a sensor capture node which distributes filtered and compiled data to the different post-processing nodes as shown in Figure 13. The sensor fusion module collects inputs from the individual sensors (including the V2X communication module) and provides a consolidated representation of the environment to the Guidance, Navigation and Control module (GNC).

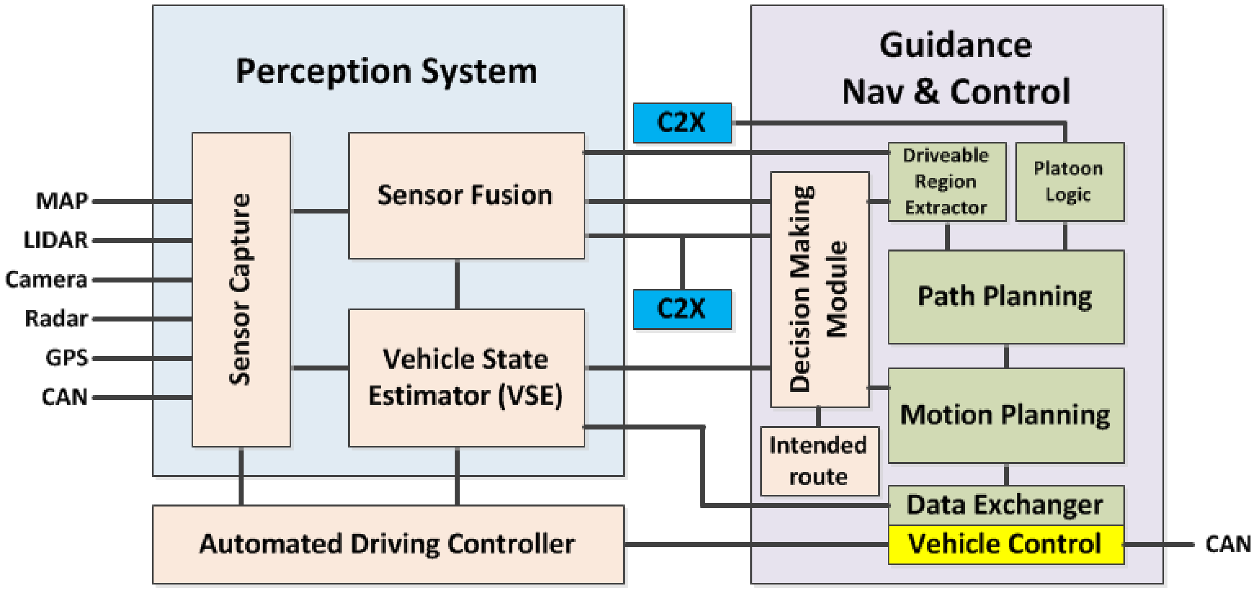

Figure 13: HMETC AD_SW building blocks (Perception and Guidance, Navigation and Control)

To support V2X communications, the HMETC vehicle supports a V2X communication module, whose interfacing with the general AD_SW framework is handled via a dedicated V2X interface node. The dedicated V2X interface node is in charge of collecting from the other AD_SW modules (sensor fusion and GNC) the needed information to be transmitted by the V2X communication module. On the reception side, the V2X interface node receives the data extracted by the V2X communication modules out of received V2X messages. The interface node then distributes this data to the different AD SW modules that reuse it. For the communication between the V2X communication unit and the V2X interface node, UDP sockets are used. The data exchange is organized in well-defined data structures that have been defined according to the SW modules that are providing and reusing the V2X exchanged information (see Figure 2 and MAVEN deliverable D5.1 (Schindler J. , Blokpoel, Rondinone, & Vreeswijk, 2018)), as well as the V2X messages employed. As an example, the V2X interface node receives dedicated subset of data from the GNC block (IF_1) , which is then reused to populate MAVEN CAM messages to inform other cooperative road user about the presence, position and dynamics of the vehicle. Similarly, the SF and GNC modules provide data structures to populate CPMs at the V2X communication module over another UDP interface (IF1 and IF4). CPMs are used to inform cooperative other road users about non-cooperative objects locally detected by the vehicle. When the V2X modules receives CAM and CPM messages, similar UDP interfaces are used to provide data structures to the SF modules that reuse it (IF_1 and IF6).
The rest of the section highlitghs the cooperative object fusion and tracking functionalities implemented in the HMETC vehicle SF module, especially when additionally handling V2X message receptions. In addition, a description of the approach used for creating a cooperative perception environment via V2X information sharing at the transmission side is given.

3.4.2. Cooperative object fusion and tracking

The objective of the Object Fusion (OF) functionality is to spatially combine redundant and complementary information of objects that are sensed by distinct sensor types. Information is redundant when describing an object via detections done by multiple sensor types, ex. Position. On the contrary, information is complementary when it enriches the object description thanks to detections done by certain sensor types only. The OF generates a more precise description of object states that can be used for further tracking and decision making tasks. In the HMETC AD SW implementation, the OF functionality is implemented by the OF module (Figure 14) hosted by the SF block. The scope of the OF module is to continuously execute an algorithm to fuse the objects list received from the different sensor types listed above. In particular:

1. It receives objects lists (object detections) from the LiDAR and Radar ECUs, Camera and V2X module
2. It fuses the objects list from different sensor types using the position, shape, object class etc., to spatially associate each detection.
3. It also generates a fused representation of each object by fusing the redundant information and appending the complementary information from distinct sensor types.

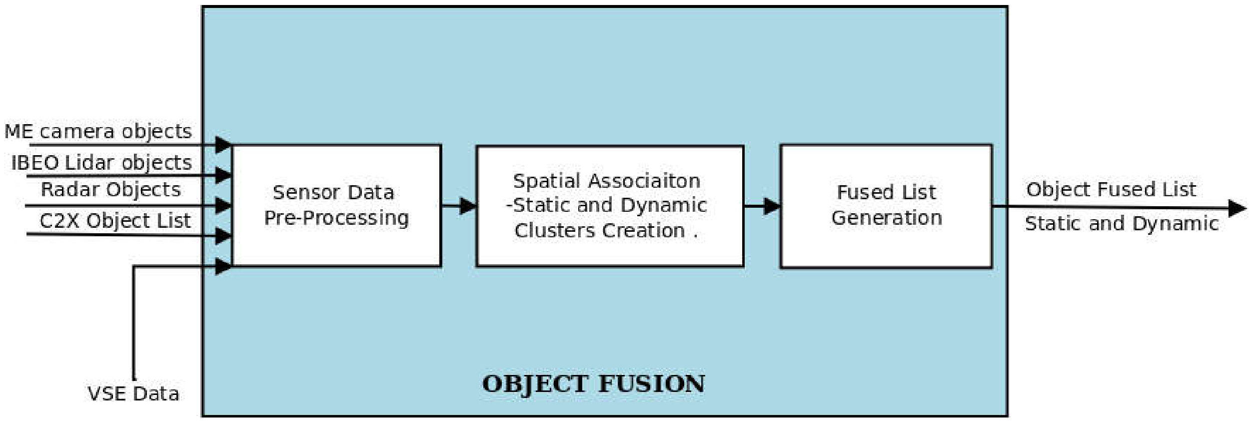

Figure 14: HMETC AD_SW OF module representation

From an algorithmic point of view, the following operations are executed:

1. The OF module receives the objects list as input from different on board sensor types that are mapped to the 3D ego vehicle coordinate system with a common time base. The expected necessary input sensor message format includes object position as the center of the object, relative velocity, length (in object's coordinate system), width (in object's coordinate system), orientation and object class, contour points of the object whichever applicable according to sensors characteristics.
2. Data synchronization: the OF module considers the most recent sensors data available every 40 msec for fusion. The synchronized data at the fusion rate will be considered further to generate a fused representation of the objects list.
3. Preprocessing steps:
   i. Removal of invalid sensor data
   ii. Consideration of Static and Dynamic objects
   iii. Limitation on number of fused dynamic object list generation
4. Static and Dynamic detections from different sensors, are then clustered using single Linkage Hierarchical Clustering Algorithm. Detections belonging to same object are identified, by testing their proximity. Detections grouped into a cluster are assigned a combined representation. This applies for each static and dynamic object.
5. Probabilistic-based Bayesian fusion and/or fusion based on sensor's priority is performed to generate the combined representation of the objects properties that are

grouped into a cluster. The redundant information from various sensors types is fused according to Bayesian rule/priority, whereas the other complimentary data are appended to the fused object representation. In this context, the priority rule follows considerations based on the measured quality of a certain sensor type to estimate a given object property. More precisely, priority for a given object property description is given to the sensor type that can better estimate it.

6. For the current fused objects, a proposal for motion model is generated.

Multi-sensor-based multiple moving Object Tracking (OT) form an essential part of the HMETC perception system. The critical aspect of such system is accurate tracking of moving objects which enable correct modelling of the environment. Most of the objects tracking approaches depend on the accuracy of the fused objects list. The HMETC OT functionality focuses on the “measurement to track” association, “track management” and tracking itself. In the following, a description of the high level design of multiple objects tracking functionality applied by the HMETC SF is given. This design is in accordance with the requirements posed by the MAVEN use cases, in particular the platooning ones.
The object tracking functionality is implemented by the OT module. The scope of this module is to run an algorithm to track the dynamic objects retrieved from the OF module and generate their estimated and predicted states for further decision making tasks. For this purpose the following steps are executed by the OT module:

1. It continuously receives the fused objects list as input from the OF module as well as a feedback of predicted states from the OT module itself.
2. For the dynamic objects’ list, it computes measurements to track association, track management followed by objects’ state update and prediction for next time instant.
3. It generates the estimated and predicted objects states with unique track ID along with the corresponding covariance values. The static objects list will get appended and published.

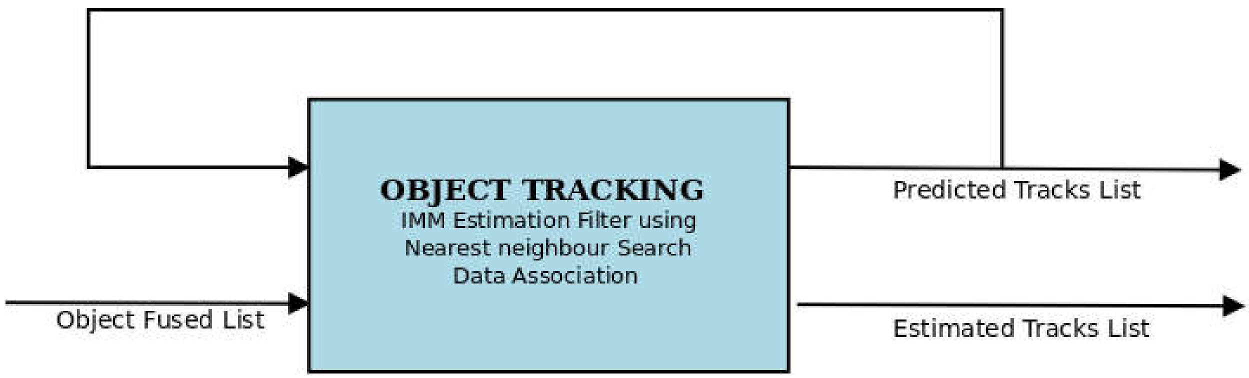

Figure 15: HMETC AD_SW OT module representation

The algorithmic approach to execute the above mentioned steps is as follows:

1. The OT module takes the fused objects list from OF module and the predicted state parameters of objects as inputs.
2. Preprocessing: the static objects list received from the OF module is filtered out, the OT module considers only the dynamic objects list for tracking.
3. Track to measurement association: A previous tracked object is associated to a current measurement using a global nearest neighbour data association algorithm.
4. Track Management and Tracks Update: After associating the track and the corresponding measurement, updating the existing track information, deletion of previous tracks and creation of new tracks is performed.
5. The current tracks will be predicted for next time instant. If a given object is described by received C2X data only, the prediction holds for the next second, which resembles, in the worst case scenario the time interval between two CAM or CPM receptions.
6. The estimated and predicted object lists along with track ID and the corresponding process noise covariance matrix is published along with static objects list. If there is no input from OT module for predefined time, the prediction cycle will be initiated with the existing tracks and the outputs will be published at the predefined rate.

When fusing and tracking objects detected concurrently by on-board sensors and via V2X receptions (CAMs or CPMs) some challenges arise. The detection rate of on-board sensors (every tens of msecs) is much higher than that achieved via V2X message receptions (here a variable reception period in the range [100-1000msec] is observed). As a consequence, the clustering of objects' measurements done via on-board sensor with those done via V2X receptions is successful only at instants close to V2X reception. As the ego-vehicle and the object move, the object states received via V2X become outdated and do not match with "fresher" measurements done by the on-board sensor until the next reception.

For coping with this issue, the OF module can run an internal prediction algorithm that looks at the content of the received V2X message to continuously estimate the current states of the object. In fact, the V2X messages contain dynamic properties of the object (position, heading, speed, acceleration, etc.) as well as the exact time at which those properties are "read". Nevertheless, less complex, yet effective approach is implemented by the HMETC OF module. Whenever, the OF module receives V2X measurements, the output list of dynamic objects is divided in two sub-lists: one that fuses measurements performed by on-board sensors only, the other one that only includes measurements deriving from V2X receptions. In the latter sub-list, the object properties (position, Speed, etc.) are static for the period between two V2X receptions. When these sub-lists are received by the OT module, they are fused in such a way to merge fast-changing properties of the objects detected by the on-board sensors, into the sub-list of V2X detections.

The following figures show a representation of the OF/OT results onto a visualization tool used in the HMETC vehicl . The snapshots depicts a situation on the Griesheim Test track in which the DLR vehicle drives, at variable speed, behind the HMETC vehicle while sending CAM messages and being detected by the HMETC car's rear Lidar. As it can be appreciated, the pink box 30m behind the ego-car in the visualization tool indicates the DLR car detected via CAM receptions. This box overlaps perfectly to the purple box indicating the fusing and tracking results of the OT module when relying on on-board sensors only. As a result, it can be stated that the fusion and tracking of V2X data in the HMETC perception system provides a correct behaviour.

Figure 16: HMETC AD_SW OF/OT performance in presence of CAM receptions

Similarly, Figure 17 shows the OF/OT results in the visualization tool when the HMETC vehicle receive CPMs including information about a pedestrian crossing the road in a section where it is not still detectable to the ego-vehicle on-board sensors (experiments done on the Griesheim test track). The snapshots depicts a situation in which a pedestrian dummy crosses the road and gets detected by another car’s camera. As it can be seen the dummy’s presence is correctly acknowledged by the OF/OT and represented in the visualization tool by a yellow dot.

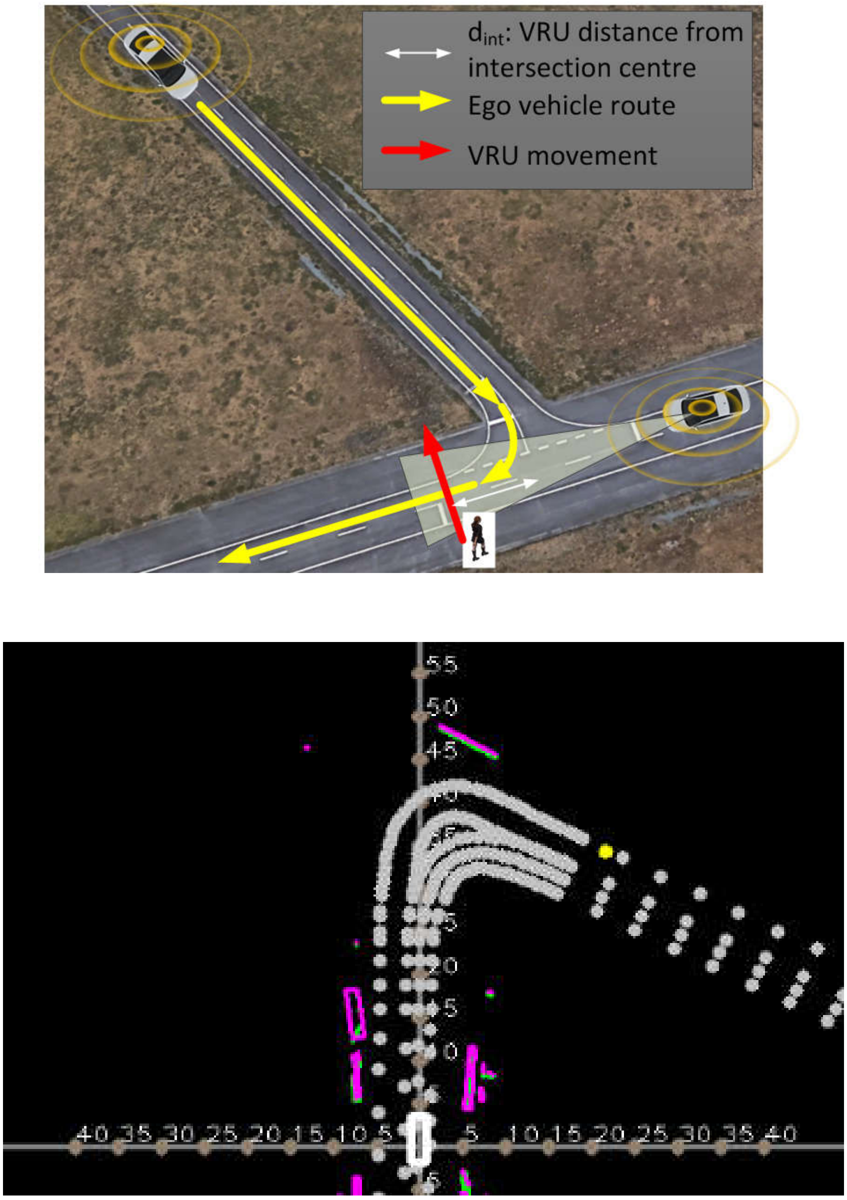

Figure 17: HMETC AD_SW OF/OT performance in presence of CPM receptions (white: ego vehicle, grey: digital map, pink: Lidar, the yellow dot represents the pedestrian detected via CPMs)

In the above mentioned cases, both the remote vehicle data received via CAMs and the pedestrian data received via CPMs are relative to the transmitting vehicle's coordinate system. Therefore, a geometric transformation has to be done so as to convert the received information with respect to ego-vehicle co-ordinates before it is sent to the OF/OT for fusion and tracking. The the X2Fusion library cooperative sensor does this transformation of IF_6 object data (Figure 5) with respect to ego vehicle co-ordinates and sends out the information to the OF module.

3.4.3. Enabling a cooperative perception environment

Besides running the functionalities for fusing and tracking detections of objects perceived via local sensors and V2X receptions, the HMETC AD_SW SF block supports generation of V2X messages that enable cooperative environmental perception at other cooperative automated vehicles. The details on the information exchanged in the V2X messages CAMs and CPMs and the mechanisms associated to their generation adhere to the specifications of the ETSI ITS standards and are explained in the deliverable D5.1 (Schindler J. , Blokpoel, Rondinone, & Vreeswijk, 2018).

CAM messages include dynamic information about the ego-vehicle. On the contrary, CPMs include a description of a set of objects detected via on-board sensors. In this section, a description of the functionalities associated to the selection of the objects included in CPM messages is given following the implementation of the HMETC vehicle.

Depending on the relative position between a given object and the ego vehicle, the object can be detected by one or more sensor types. For example, close objects to the vehicle front can be detected concurrently by Lidar medium range radar and camera. On the contrary, objects on the sides would be detected by corner radars only and possibly additionally by Lidar, assuming that either the front or rear Lidar has previously detected that object when placed in the front or in the back of the car, and that the object is being tracked by the Lidar tracker ECU. In unfortunate cases, a far object is detected by only one sensor type, either the Lidars or the radars. Whether an object is detected by one or more sensors gives an indication of the confidence with which the object is described and represented. Objects that are detected concurrently by multiple sensors types are therefore more worthy to be considered by the ego-vehicle OF/OT as well as cooperatively disseminated via V2X to other cooperative users.

The HMETC AD_SW builds up on this rationale to provide a functionality for deciding which detected objects shall be included in CPMs. This functionality is implemented in the Fusion2X library cooperative sensing module of the AD_SW depicted in Figure 5. Data from the OT module relative to detected objects is ranked before being included in the structures of IF_4 and sent to the C2X communication interface node (Figure 5). The ranking is done based on how many sensors have detected the object. The highest rank is given to objects that are concurrently detected by camera, Lidar and radars, the lower rank is given to objects that are detected by one sensor type only. Then, the object list output of the OT module is re-ordered based on the ranking. Objects with higher ranking get the first positions in the list. As the IF_6 structure can host only a limited amount of detected objects to be included in the CPM, the objects with the lower ranking are filtered out.

4 Conclusion

In this document, the project achievements in cooperative perception algorithm development and testing were presented.

In the course of a review of the state-of-the-art it was found that the cooperative perception algorithms known from literature have a common structure. They need data association, data fusion, tracking and temporal as well as special alignment building blocks.

However, the spectrum of known algorithms differs significantly in their level of sophistication, computational cost and performance. Combining the building blocks outlined in this report gives a wide spectrum of possible algorithmic implementations. Investigation of the whole spectrum is beyond the scope of this report. Instead, systems were implemented and tested that can be located at the different end of this spectrum. At HMETC, a fully featured real-time capable multisensory multitarget tracking pipeline was implemented and demonstrated in the controlled environment of a test track.

Fusion experiments were successfully performed for position measurements of the DLR's ViewCar2 received via CAM messages and its sensor detections by HMETC's test vehicle. It was however demonstrated that the state estimate of ViewCar2 could be improved.

Further experiments were made with a pedestrian dummy crossing the road at a location that was not visible for the test vehicle's sensors. It was demonstrated that extending the field of view of a vehicle works well. The pedestrian dummy was detected by the camera of a second vehicle, a CPM message was generated and transmitted to the HMETC test car and added to the list of observed objects.
The specific challenge of confidence ordering was identified and mastered during the implementation of the multi-sensor fusion pipeline at HMETC. The trust in a target detection is as high as the number of sensors that concurrently register the target. Consequently, confidence ranking is reasonable in the process of CPM message compilation. CPM messages have limited capacity. If the capacity of the message is exceeded, the target detections with the highest trust should be included first.

Another specific challenge is the preprocessing of data, especially when the number of target detections gets high. Velocity and map filtering were successfully tested in this context.

The activities at DLR focused on investigating the performance limits and boundaries of the algorithms in urban traffic scenes that were recorded and processed offline. Additionally, simulated test runs were carried out to explore the potential benefit of different types of algorithms.
The fully featured real time implementation uses Euclidean state vector distance metric and accounts for the current time without looking for time history. It proved to work well on the test track. Using Mahalanobis distance measurements over a time history is the most advanced approach known in literature but gives no benefit on the test track. However, in more complex urban environment scenarios, where multipath propagation effects adversely affected ego-localization accuracy and where the target density was high, time history based association could play out its advantage. This finding is relevant for the definition of testing procedures for cooperative perception algorithms. Further software code optimization is yet needed to be able to run the algorithm in real time.

Fast Covariance Intersection, the most powerful approach known yet from literature, was evaluated in Monte Carlo simulation runs. It proved to give higher measurement accuracy than each of the sensor data sources. However, simply averaging the state estimates will do well in most scenarios, especially when tracks are unstable and covariance estimates are missing.

The shown approaches are used in the real world implementations in WP6 and described in more detail in the upcoming deliverables D6.4 and D7.2, which are showing field test and evaluation results.

5 References

Anderson, J., Klara, N., D.Stanely, K., Paul, S., Constantine, S., & Oluwataba, A. (kein Datum). *Autonomous Vehicle Technology. A Guide for Policymakers, RAND corporation.* RAND Corporation, Santa Monica, Calif.

Berg, J. v., Ferguson, D., & Kuffner, J. (May 2006). *Anytime Path Planning and Replanning in Dynamic Environments.* Orlando, Florida: Proceedings of the 2006 IEEE International Conference on Robotics and Automation.

Bergenhem, C., Huang, Q., Benmimoun, A., & Robinson, T. (kein Datum). *Challenges of Platooning on Public Motorways.* 17th ITS World Congress.

Dariani, R. (2016). *Hierarchical Concept of Optimization Based Path Planning for Autonomous Driving.* Magdeburg: Otto-von-Guericke-Universität Magdeburg.

Dariani, R., Schindler, J., Leich, A., Rondinone, M., Matheis, D., & Walter, T. (2018). *Detailed concepts for cooperative manoeuvre and trajectory planning and for in-vehicle cooperative environment perception.*

Darms, M., Rybski, P. E., Baker, C., & Urmson, C. (09 2009). Obstacle Detection and Tracking for the Urban Challenge. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*.

Duraisamy, B., Schwarz, T., & Wöhler, C. (2015). On Track-to-Track Data Association for Automotive Sensor Fusion. *18th International Conference on Information Fusion.*

E.W.Dijkstra. (1959). *A Note on Two Problems in Connexion with Graphs.* Numerische Mathematik 1.

Forschungsinitiative Ko-FAS. Ko-PER - Kooperative Perzeption. (n.d.). Retrieved 02 25, 2019, from http://ko-fas.de/41-0-Ko-PER---Kooperative-Perzeption.html

Franken, D., & Huepper, A. (2005). Improved fast covariance intersection for distributed data fusion. *7th International Conference on Information Fusion.* IEEE.

Horizon 2020 Work Programme 2016-2017 - Smart, green and integrated transport. (2019, February). Retrieved March 2019, from http://ec.europa.eu/research/participants/data/ref/h2020/wp/2016_2017/main/h2020-wp1617-transport_en.pdf

Houenou, A., Bonnifait, P., Cherfaoui, V., & Boissou, J.-F. (2012). A Track-To-Track Association Method for Automotive Perception Systems. *Intelligent Vehicles Symposium.* Alcalá de Henares: IEEE.

Julier, S., & Uhlmann, J. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. *Proceedings of the 1997 American Control Conference.* IEEE.

LaValle, & Stevem. (October 1998). *Rapidly-exploring Random Trees: A new tool for path planning.* Computer science department, Lowa state University: Technical report.

Loeper, C., Brunken, C., Thomaidis, G., Lapoehn, S., Fouopi, P., Mesobach, H., & Koester, F. (October 6-9, 2013). *Automated Valet Parking as Part of an Integrated Travel Assistance.* Netherlands: Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013).

Müller, D., Meuter, M., Ghosh, L., & Müller-Schneiders, S. (2011). A generic video and radar data fusion system for improved target selection. *IEEE Intelligent Vehicles Symposium (IV).*

Niehsen, W. (2002). Information fusion based on fast covariance intersection filtering. *IEEE Conference on Information Fusion.*

Rajamani, R., Tan, H.-S., Law, B., & Zhang, W.-B. (July 2000). *Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons.* IEEE Transactions on Control Systems Technology, Vol. 8, No. 4.

ROS.org - tf. (2019, February). Retrieved 02 05, 2019, from http://wiki.ros.org/tf

S.Schmidt. (2013). *Ein Optimales Steurungs- und Reglungskonzept für autonome Elektrofahrzeuge.* Magdeburg: Otto-von-Guericke-Universität Magdeburg.

Sawade, O., Schäufele, B., Buttgereit, J., & Radusch, I. (2014). A cooperative active blind spot assistant as example for next-gen cooperative driver assistance systems (CoDAS). *IEEE Intelligent Vehicles Symposium Proceedings.*
Schindler, J., Blokpoel, R., Rondinone, M., & Vreeswijk, J. (2018). *MAVEN Deliverable 5.1, "V2X communications for infrastructure-assisted automated driving".*
Schindler, J., Blokpoel, R., Rondinone, M., Walter, T., & Přibyl, O. (2018). *MAVEN Deliverable 6.3: Intermediary report of the integration.*
Schindler, J., Dariani, R., Rondinone, M., & Walter, T. (2018). Dynamic and Flexible Platooning in Urban Areas. *AAET Automatisiertes und vernetztes Fahren.* Braunschweig.
Schindler, J., Dariani, R., Rondinone, M., & Walter, T. (2019). Implementation and Testing of Dynamic and Flexible Platoons in Urban Areas. *AAET 2019 – Automatisiertes und vernetztes Fahren.* Braunschweig.
Seeliger, F. (2017). Fahrzeugübergreifende Informationsfusion für ein Kreuzungsassistenzsystem.
Seeliger, F., & Dietmayer, K. (2014). Inter-vehicle information-fusion with shared perception information. *International IEEE Conference on Intelligent Transportation Systems.* IEEE.
Seelinger, F., & Dietmayer, K. (2014). Inter-vehicle information-fusion with shared perception information. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC).*
Smith, R., & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*.
Steuernagel, S. (2019). *Track-to-Track Association and Fusion for Cooperative Automotive Environment Perception.* Georg-August-Universität Göttingen.