D 6.2 – SMALL SCALE EVALUATION TRIALS

Authors: Petros Kapsalas, Nikos Piperigkos, Andreas Kloukliniotis, Jordi Casademont, Marta Kasprzak, Mina George Farahat, Wojciech Jaworski, Antonio Alvarez Romero, Javier Fernandez Hidalgo, Alessandro Zanella, Rosemma Gario, Alessandro Cacciatore, Aris Lalos, Apostolos Fournaris

WP6 – Industry Driven Trial and Evaluation

Abstract

This report contains the output of Task 6.2 which defines and designs the demonstration trials and the validation process. It consists of designing and planning the way the pilot studies will be organized, supported, and managed throughout the duration of the project. The deliverable is organized around two use case groups. First one Human-Robot Interaction in manufacturing environment includes following use cases: a design operation continuum evaluation and resilience and safety. Second group of connected and autonomous L3-L4 vehicles use cases consists of: human in the loop control, cybersecurity issues and cooperative awareness.

Funded by the Horizon 2020 Framework Programme
of the European Union

Deliverable Information	
Work Package	WP6 Industry Driven Trial and Evaluation
Task	T6.2 Deployment of Small-Scale Trials
Deliverable title	Small-scale evaluation report
Dissemination Level	Public
Status	Final
Version Number	2.0
Due date	28/02/2022
Project Information	
Project start and duration	1.01.2020-31.12.2022
Project Coordinator	Industrial Systems Institute, ATHENA Research and Innovation Center 26504, Rio-Patras, Greece
Partners	1.ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS (ISI) - Coordinator 2. FUNDACIO PRIVADA I2CAT, INTERNET I INNOVACIO DIGITAL A CATALUNYA (I2CAT), 3. IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD (IBM ISRAEL 4. ATOS SPAIN SA (ATOS), 5. PANASONIC AUTOMOTIVE SYSTEMS EUROPE GMBH (PASEU) 6. EIGHT BELLS LTD (8BELLS) 7. UNIVERSITA DELLA SVIZZERA ITALIANA (USI),

	8. TAMPEREEN KORKEAKOULUSAATIO SR (TAU) 9. UNIVERSITY OF PELOPONNESE (UoP) 10. CATALINK LIMITED (CATALINK) 11. ROBOTEC.AI SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA (RTC) 12. CENTRO RICERCHE FIAT SCPA (CRF) 13. PANEPISTIMIO PATRON (UPAT)
Website	www.cpsosaware.eu

Control Sheet

VERSION	DATE	SUMMARY OF CHANGES	AUTHOR
0.1	15/12/2021	Toc and First contribution of Partners	*All*
1.0	15/01/2022	Updated input on Connected and Automated Vehicles	*ISI, UPAT, PASSEU*
1.1	10/02/2022	Updated Input in the manufacturing pillar and circulation for REVIEW by IBM, I2CAT	*CRF, UPAT, PASSEU*
2.0	28/02/2022	Final version to be submitted	*PASSEU*

	NAME
Prepared by	PASEU
Reviewed by	I2Cat, IBM
Authorised by	ISI
DATE	RECIPIENT
-	Project Consortium
26/02/2022	European Commission

Table of contents

1. Introduction 7
2. Evaluation Trials 7
2.1 Human-Robot Interaction in Manufacturing Environment 7
2.1.1 Demonstrator's set-up 7
2.1.1.1 A design operation evaluation 9
2.1.1.2 Resilience and safety 13
2.1.1.3 Safety System and tasks sequences 15
2.1.2 Workcell setup, and 1st functional and reachability testing 18
2.1.3 Anthropometrics definition 19
2.1.4 Different approaches for the estimation of the operator's height via visual analysis 21
2.1.4.1 Evaluation using different scenarios in the virtual environment of the simulator 22
2.1.4.2 Evaluation in three different operator's height scenarios for the use case 1 24
2.1.4.3 Evaluation in three different operator's height scenarios for the use case 2 26
2.1.4.4 Observational study 27
2.2 Connected and Automated Vehicles 32
2.2.1 Human in the loop control use case in single vehicle scenario 32
2.2.1.1 Evaluation concept of the targeted components and use case 32
2.2.1.2 Testing Procedure and Data Acquisition 33
2.2.1.2.1 Simulator laboratory test procedures 41
2.2.1.3 Key Performance Indicators 42
2.2.1.3.1 Quantitative indicators 42
2.2.1.3.2 Qualitative Indicators 44
2.2.2 Cyber Security Use-Case 45
2.2.2.1 Cyber-Attacks Introduction 45
2.2.2.2 Removal of Cyber-Attacks at the Camera Level 46
2.2.2.2.1 Total Variation 48
2.2.2.2.2 Anisotropic Diffusion 51
2.2.2.3 Evaluation concept of the targeted components and use case 54
2.2.2.4 Evaluation concept of the targeted components and use case 54
2.2.2.4.1 Testing Environment 55
2.2.2.4.2 Abnormal/Malicious Behavior test setup 56
2.2.2.5 Testing procedure, data acquisition protocols and preliminary results 57
2.2.2.5.1 Preliminary results 58
2.2.2.6 Planning of Large-scale trail test procedure 61
2.2.2.7 Dataset Acquisition Process (Real Vehicle) 63
2.2.2.7.1 Cyber Attack Mitigation- Evaluation Protocol 67
2.2.2.7.2 Cyber Attack Mitigation- Evaluation Protocol 68
2.3.1 Co-Operative Awareness Scenarios 69
Evaluation concept of the targeted components and use case 69
2.3.1.1 Multimodal object detection solutions 69
2.3.1.1.1 Cooperative Localization, Tracking and Awareness tasks of WP3 71
2.3.1.1.2 Robotec Simulation Framework 74
2.3.1.2 Testing environments 76
2.3.1.3 Simulation Based Evaluation 77
2.3.1.3.1 Dataset Acquisition Process (CARLA) 77
2.3.1.3.2 Dataset Acquisition Process (Real Vehicle) 80
2.3.1.3.2.1 Structure of the Testing Dataset 81
2.3.1.3.2.2 Statistical Evaluation Process 82
2.3.1.3.2.3 Inter communication simulation (V2X simulator) 83
3. Conclusions 96

4. References 97

List of figures

Figure 1: Final CRF Layout in laboratory demonstrator 9
Figure 2: Mixed Reality scope 10
Figure 3: Main industrial manufacturing mixed reality analysis 10
Figure 4: CRF MR tools activities diagram flow 12
Figure 5: 3D object mapping (left) and Spatial mapping (right)........ 13
Figure 6: CAD-model of Rack with windshield........ 14
Figure 7: Force-Torque Sensor FTN – Omega 191 assembled on the gripper before final assembly 14
Figure 8: Virtual Safety area in the work-cell 15
Figure 9: Main robot tasks sequence – Explanation in Table 2 below........ 17
Figure 10: CRF Robot workcell installation. 18
Figure 11: Overview of the control system's steps. 19
Figure 12: The 25 landmarks of the BODY_25 model. 22
Figure 13: Example of the extracted json files. 23
Figure 14: Dimensions of the box located in the scene. 24
Figure 15: Recognition of box corners and estimation of their coordinates. 24
Figure 16: Two frames capturing the collaboration task between operator and robot under three different heights of the operator (a) 170 cm, (b) 180 cm, (c) 190 cm. The camera is located in front and side of the operator. 25
Figure 17: Two frames capturing the collaboration task between operator and robot under three different height states of the operator (a) 170 cm, (b) 180 cm, (c) 190 cm. The camera is located exactly behind the operator. 27
Figure 18: Simulation and preliminary evaluation of the operator and collaborative robot contact phases. The robot stops and the red light comes on........ 28
Figure 19: Xsens systems representation 29
Figure 20: VTA idea 29
Figure 21: Pointer for Gaze monitoring 30
Figure 22: Gaye and Gesture Usage 31
Figure 23: Virtual CAD in real environment........ 31
Figure 24: Training on the job MR tools 31
Figure 25: (a,b) Step-1: Dropdown menu and select "Screen Record". (c) Step-2: Allow the recording of audio. (d) Step-3: Press " Start now". (e) Step-4: The recording should start 33
Figure 26: (a) Welcome screen of the DSM application. (b) The front camera is engaged, and the face detection has been initiated. 34
Figure 27: (a) Result of the "Eye closure" scenario. (b) Result of the "Yawning" scenario. (c) Result of "Distraction" scenario...... 35
Figure 28: (a-b) Generated JSON file. (c) Session is over and the JSON is saved........ 36
Figure 29: Small scale trial evaluation environment 55
Figure 30: Expected/nominal behaviour of the small scale trial on cybersecurity validation........ 56
Figure 31: Corrupted firmware due to network transmission loss visual result 58
Figure 32: Malicious firware due to man in the middle attack visual result 59
Figure 33: Malicious firware due to man at the source attack visual result........ 60
Figure 34: Overall cybersecurity compliant demonstration and validation scenario for network communication in the automotive pilot........ 62
Figure 35: Picture of our ego-vehicle with sensors mounted 63
Figure 36:First Experimental Setup 63
Figure 37:ADAS System Architecture Diagram........ 64
Figure 38: ADAS System Cameras........ 64
Figure 39: ADAS System Sonars 65
Figure 40: Example of Occupancy Grid Map, which contains obstacle cells (white), free-space cells (black), and unknown cells (grey). Red cell are reflecting the current update of cells of obstacle points. 68
Figure 41:: Multimodal fusion approach and generation of range measurements 70

Figure 42: Fusion outcome qualitative evaluation ... 70
Figure 43: Proposed multi-modal fusion architecture ... 72
Figure 44: Artery architecture ... 73
Figure 45: Simulation framework setup ... 74
Figure 46: RoSi view of V2X messages together with GPS and Cooperative Localization errors. ... 74
Figure 47: High level architecture of V2X simulator. ... 75
Figure 48: Visualization of the copy of simulated world in V2X simulator ... 75
Figure 49: From Top to bottom and left to right RGB camera, Depth Camera, Lidar, Semantic Segmentation Camera ... 79
Figure 50: Rendered 3D Bounding Boxes ... 80
Figure 51: CARLA scenes examples ... 81
Figure 52: Evaluation of Cooperative Awareness solutions in the simulated framework ... 82
Figure 53: Structure of the LTE-PC5 resource grid with 10 subchannels per subframe ... 86
Figure 54: Sensing Based Semi-Persistent Scheduling ... 87
Figure 55: PRR over distance with a varying CAM interval for LTE-PC5 Mode-4 (blue solid lines) and 802.11p (red dashed lines) . 90
Figure 56: NAR for LTE-PC5 Mode-4 and 802.11p in Fast and Slow Highway scenarios. ... 91
Figure 57: Position Error for LTE-PC5 Mode-4 and 802.11p in Fast and Slow Highway scenarios. ... 92
Figure 58: NAR for LTE-PC5 Mode-4 and 802.11p with varying CAM interval. ... 93
Figure 59: Position Error for LTE-PC5 Mode-4 and 802.11p with varying CAM interval. ... 94

List of tables

Table 1: Challenge 2 task analysis ... 16
Table 2: Main robot/operator tasks sequence (sequence referred to Figure 9) ... 17
Table 3: Calculated distances (in pixel) and estimated heights (in cm) for the use case 1 under different heights of the operator (with estimated S=2.1). ... 26
Table 4: Calculated distances (in pixel) and estimated heights in cm for the use case 2 under different heights of the operator (with estimated S=2.4). ... 27
Table 5: Sensors’ configuration Overview ... 78
Table 6: CAMERA SPECIFICATIONS ... 80
Table 7: LIDAR SPECIFICATIONS ... 80
Table 8: IMU AND GNSS SPECIFICATIONS ... 80

1. Introduction

As the outcome of D6.2, which designed and implemented the demonstration and evaluation scheme based on the research literature, industrial standards, and state-of-the-art methods, the following report describes the selected use-cases and describes in detail the dataset usage, the structure, the generation process and the planning of the pilot studies. The evaluation trials use cases are concentrated around two areas:

- Human-robot interaction in manufacturing environment, with the following use cases: a design operation continuum evaluation, and resilience and safety.
- Connected and autonomous L3-L4 vehicles, with the use cases: human in the loop control, cybersecurity issues, and cooperative awareness.

D6.2 provides detailed descriptions of how the demonstration of planned and conducted while also it discusses the steps for quantifying the outcome for each use case, describing evaluated CPSoSaware components The objective of the evaluation/ demonstration scenarios for each use case is to study the potential and limitation of the CPSoSaware system components in dedicated environments to ensure the best system's performance and to provide informed experimental feedback for future enhancement of the technology.

2. Evaluation Trials

2.1 Human-Robot Interaction in Manufacturing Environment

2.1.1 Demonstrator's set-up

A full preliminary analysis of the safety constraints, of the task phases and of the layout (full simulation of the position and dynamic of robot) was made in order to comply with European safety requirements and CRF internal policies. In particular, the set-up required the installation according to EU Machinery directive (2006) with the definition of all the safety procedures and documentation in order to fully define the boundary conditions for the final certification. The Kuka safe zones (non intrusion zones defined in the the Safe programming module of the Kuka safe robots - PLd Cat3 according to ISO 10218, part2:2011) had to be defined and coordinated with the Safety Eye zone controller.

CRF also identified and procured the last commercial version of the windshields for testing and later assembly with a particular attention to existence of variants of configurations necessary to test the "training on the job" logics implemented in the CPSoSaware project"

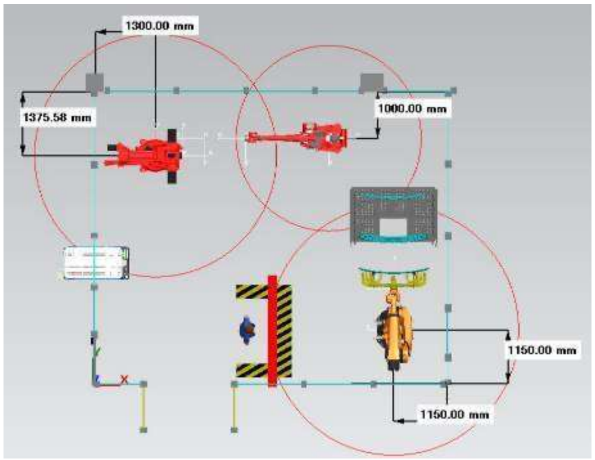
CRF also performed the study of the ergonomic issues related to the use-case in a dedicated and "ad hoc developed" VR environment in order to identify specific aspects of the application to be considered into the final optimal demonstrator activity. The workplace was optimized accordingly to the expected scenarios.

The deployment of the work cell in CRF premises took place till its complete realisation and integration. Because of limitations in accessing the laboratories and the additional slowdown of some procedures (in particular related to ICT new connections) in CRF due to the COVID19 restrictions and the limited presence in the laboratories. The CRF robot setup is used also in another project ending at the beginning of 2022; the delay described above required, as a conseguence, the need to move the setup in a location different from CRF in order to finalize the use-case deployment. This situation affected and will affect the CPSoSaware project till April due to the fact that the robot won't be available on site for local testing and some activity foreseen in the fourth semester will be delayed and will be completed in the third year in the context of T6.4.

During integration and setup stages, CRF defined and tested the base programming of the robot for the CPSoSaware project's scenarios. Considering the "Safe control" nature of the scenario, the base programming is mainly focused on the definition of the safety logics of the system, Safety systems, procedures and so on. In the CPSoSaware project CRF is setting up a collaborative workstation to enable the testing and development of the technologies developed by the project's partnership.

In the period of testing and development of the "small scale trials" in the project, CRF have been focused on the workcell construction and set-up. During the same period some development (mainly in WP3 and WP5) related to functional developments for the use of Virtual Reality (VR) and Mixed Reality (MR) have been made. The physical testing and small trials to validate partial developments are described in D6.2.

During the period of activities described in D6.2, the final setup was made in CRF premises according to the planned set-up. A base layout for the intervention was defined accordingly and depending on the functions to be demonstrated see Figure 1. This layout has been respected in general terms and only minor adjustments have been necessary due to constraints in positioning the equipment in reference to existing infrastructure.



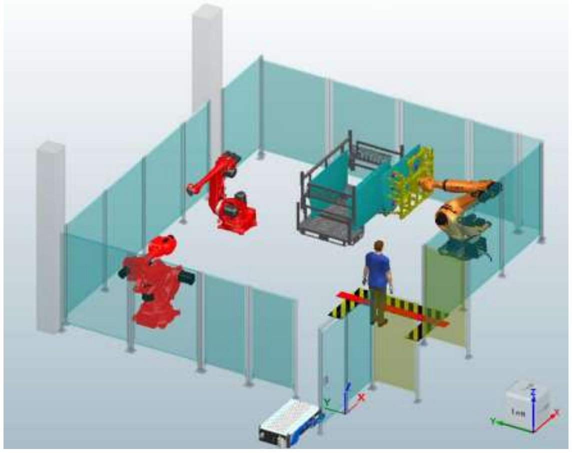


Figure 1: Final CRF Layout in laboratory demonstrator

The work-cell was set up in the period from May 2021 to October 2021. Testing of the small scale trials foreseen preliminarily such as Operator’s state monitoring were expected to happen on the robot in the work cell, but so far have not been studied. Delays in the material supplies and integration, delayed the workcell setup. Nevertheless, the setup has been completed integrating all the additional safety systems required for better and safer integration to the CPSoSaware system.

In the following subchapter, the tests performed in WP6.2 are described. The state of safety of the workcell is introduced in the “Safety and Resilience” use-case.

2.1.1.1 *A design operation evaluation*

Mixed reality design and simulation is developed in order to improve the “smartness” and self-adaptability of the workcells in the automotive manufacturing field.

The developments cover all the required technologies from Virtual to Mixed realty. The virtual reality simulation is performed in cooperation with CRF-UPAT; the training on the field assisted by MR is mainly developed by CRF in T5.3.

CRF focuses on the Mixed Reality development based on the use of the Hololens with training on board tools capable of assisting the operator performing his activities during the operations and suggesting the best approach to perform them.

The use of mixed-reality devices and tools permits to overcome the limitations due to augmented reality using tablets and smartphones that do not allow hands-free operations. In these cases, the MR tools can integrate information and virtual CAD drawings in the real environment for an improved level of interaction and interpretation of the scenario.

Mixed Reality is defined as the merging of real and virtual worlds to produce new environments and visualizations where physical and digital objects co-exist and interact in real time. In particular MR aims to:

✓ combine the best aspects of both virtual reality and augmented reality.

✓ Create an environment in which users seamlessly navigate through both the real and virtual environments at the same time. Instead of residing in an entirely virtual world (i.e., virtual reality), virtual objects are anchored into a user’s real world space and augment their real world environment, making virtual interactions appear to be “real.” These interactions mimic our natural perception such as objects getting bigger as you get closer and the changing of perspectives as you move around an object.

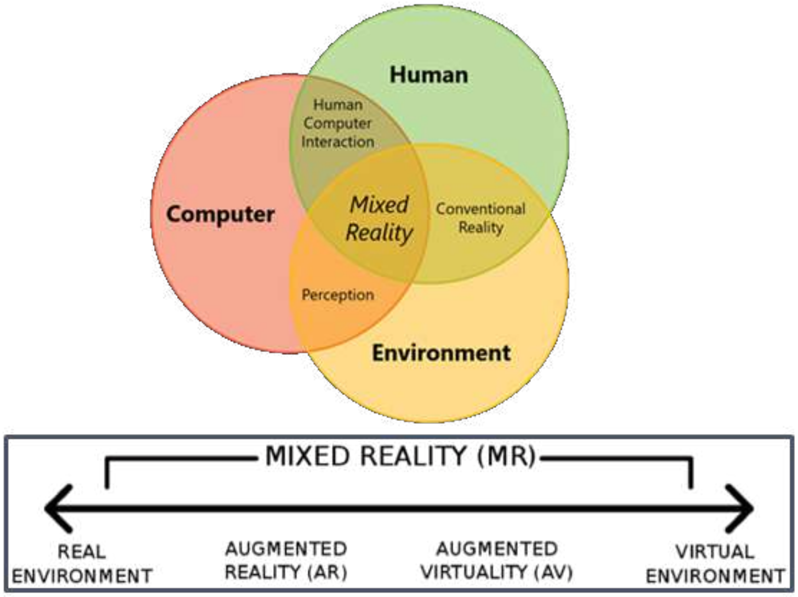


Figure 2: **Mixed Reality scope**

In the field of industrial manufacturing the mixed reality is already used for several analysis:

- ***Functional mock-up***: build mockups that combine physical and digital elements

- ***Maintenance/Training global workforce for remote teams***: No matter people location, language (real time translator) offers more flexibility. Remote activities on devices and workstations. Remote maintenance assistance

- ***RAVE***: Real Asset Virtualization Environment, associate virtual 3D assets to real time data associated with those assets.

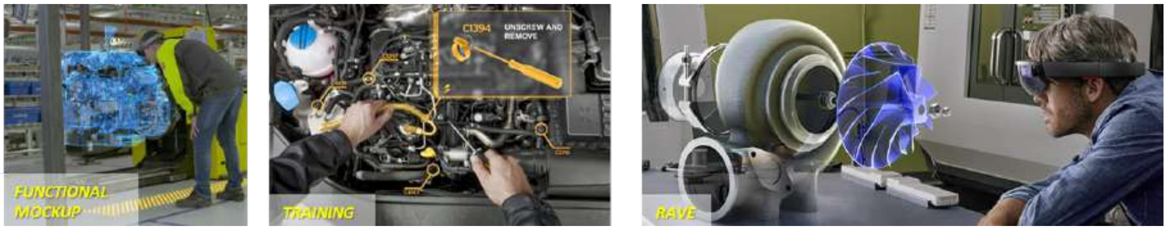


Figure 3: Main industrial manufacturing mixed reality analysis

From CRF benchmark analysis the most suitable tool is the Microsoft hololens.

The use of Hololens 2[1], a state-of-the-art tool in the field of mixed reality, will help the operator in performing the tasks. Various processes are linked together, according to the project's targeted integration of different intelligent and collaborative systems.

This technology employs holograms (3D images with correct prospective and POV - Point of View) projected on glasses in order to see and interact with virtual elements within a real environment.

The work activities consist in:

The development of a training tool made in a mixed reality (MR) environment, which is a combination of real physical and virtual objects. At this stage, the user is guided by a virtual UI, but the work tools and equipment are real, so the user can perform the task with little help. The system will recognize trends in touch events of the control UI, time spent interacting with the UI, and when the UI is still needed. The evaluation procedure will maintain some common features with the VR evaluation procedure to ensure continuity of evaluation.

Within the MR experience, continuous training is ensured by the implementation of the Edit feature, which makes it possible to modify some elements of the training applications. A new set of instructions will then be requested from the operator, and the change in required procedures will initiate the adaptation process.

[1] https://www.microsoft.com/en-us/hololens

The procedure is as follows:

- Update of the set of instructions.
- Transmission to the AR/XR wearable device on the updated instruction.
- Execution of the new procedures while monitoring the operator’s actions.
- Continuous update of the state with evaluation of the training performance (cycle time).
- Stop of the training on the job when the result is considered satisfactory (by the operator, system, input limitation, and others).

The logical cycle of the training that will be integrated and developed is briefly described in the Figure 4 below.

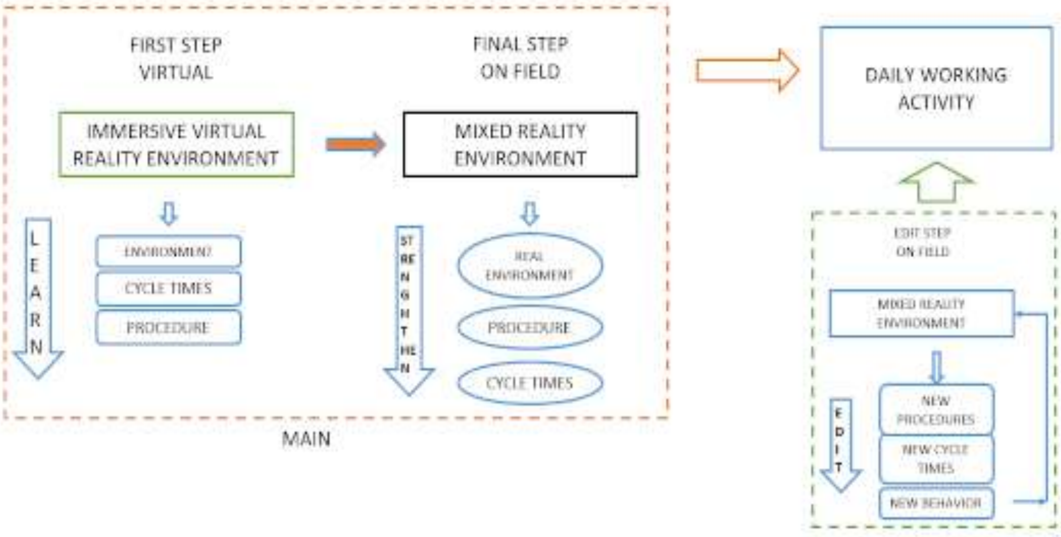


Figure 4: **CRF MR tools activities diagram flow**

The developed tool consists in:

- ✓ User interface to execute the various steps in a guided way through textual, graphic or 3D indication.
- ✓ Guided control of the various operations that the operator will execute in order to verify the correct execution of the operation with eventual security control through the use of CAD animations or texts correctly positioned in the real space.
- ✓ Remote communication for sending mail/Alert

The methodology must include high level of interaction between real and virtual environments assuring:

- Picking objects and their virtually assembly to the engine.

- Spatial mapping.
- Synchronized moving of real parts and holograms.

The features that could be implemented are:

- Picking objects gestures with clicker/joystick/gloves.
- Animation Synchronism holograms vs real environment/Collisions.
- Collision Feedback visive/acoustic/physic.
- Physic: gravity and rigid body dynamic.
- Additional info by markers/tags (typical of ARenv.).

Figure 5: **3D object mapping (left) and Spatial mapping (right)**

CRF have been developing and setting up the structure of the programs and MR interface for interaction with the operator in the “design operation continuum evaluation”, nevertheless the full integration will be performed in the final robot environment in the last semester.

Activities performed on similar demonstrators have been made in CRF to test the technologies and start developing the interfaces that will be later integrated.

2.1.1.2 *Resilience and safety*

The Safety aspects related to the Manufacturing Workcell developments have been further analysed while setting up the full workcell. The workcell consists mainly of:

- The robot
- its end-effector (pick & place and manipulation end gripper)
- the control system (robot controller, external PC for HG motion control…)
- The Safety system (Safety eye, additional cameras, access control, safe controller)
- Other environment components:
  - Windshield rack
  - Side line logistics

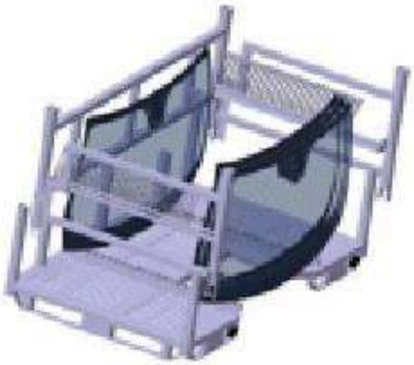


Figure 6: **CAD-model of Rack with windshield**

For the installation in the set-up a simplified version of the safety rack was necessary. The simplification was due to the fact that in the demonstrator, no conveyor to simulate the final assembly was possible. Because of this, there is no possibility to drop the windshield in a location which is different from the pick up zone. The system will thus pick the windshield from the rack (Figure 6); perform a manipulation and drop again in the same position in order to be able to restart the cycle again. The workcell is based on a KUKA KR 150- 2700 robot, with in-built safety feature (selected according to its payload, reach and the integrated safety). The robot integrated safety level required is PLd Category 3 ensuring thus a safe control and stop capability and equipped with safety management SW.

The gripper for the windshield manipulation is a customized gripper capable of performing Hand guiding actions; nevertheless, the function isn't expected to be used in the CPSoSaware project. A Force-Torque sensor is introduced to implement the HG into the collaborative system as an added safety unit. Based on the total maximum estimated payload of 120-130 kg, and robot accelerating mostly at 1G (since human-robot collaborative application) and at-most 2G acceleration and humans exerting torques of around 350 Nm, model FTN-Omega 191.

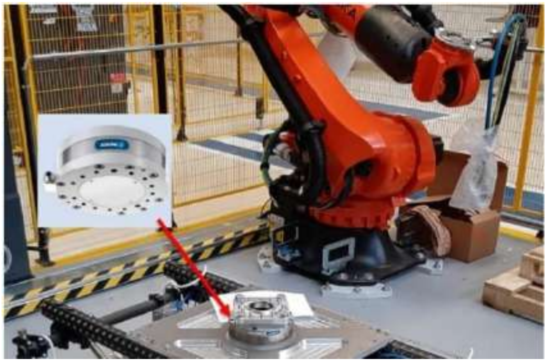


Figure 7: **Force-Torque Sensor FTN – Omega 191 assembled on the gripper before final assembly**

A Force-Torque sensor (Figure 7) is introduced into the collaborative system as added safety unit. Based on the total maximum estimated payload of 120-130 kg, and robot accelerating mostly at 1G (since human-robot collaborative application) and at-most 2G acceleration and humans exerting torques of around 350 Nm, model FTN-Omega 191.

2.1.1.3 *Safety System and tasks sequences*

In the work-cell the Safety EYE system from PILZ is included for a safe monitoring of the workcell. The System will monitor the accesses to a series of safety zones defined in the cell. This system will run in a parallel, high priority thread together with the Robot's Safe operations zones program in order to monitor and ensure the safe coordination of the robots and the logic of the interlocked doors.
Any additional logic from the CPSoSaware control won't interfere with the safety low level thread.
Entering in details of the Safety zones programming a preliminary definition at the ground of four levels of safety zones (yellow, orange red and violet – See Figure 8) is made. A task sequence is defined with reference to the figure. It is important to consider that:

1) In the demo the windshield container will be a slim stand with only one windshield.

2) The floor signs are customized to identify immediately the operating zones and could be reflected in MR

3) The coloured areas in the background are virtual and indicative; the horizontal signs for the operator will be made with adhesive tapes on the ground; eventually they could be projected to the ground or represented in MR

4) The operating speed in the absence of risks is called V; V> V1> V2. The values of speeds are based on ISO/TS 15066 calculation in SSM mode; indicatively V = 500mm/s; V1 = 350mm/s; V2 = 250mm/s

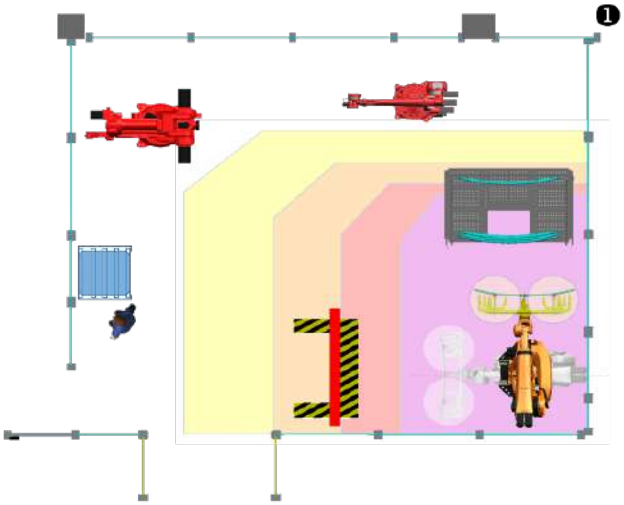


Figure 8: **Virtual Safety area in the work-cell**

From the interfacing logics between the robot with X11/X13 safety input options and the output of the Safety Eye, the safety zones have the following meaning and connection:

- **Yellow** area. If violated it slows the robot to speed V1. If the operator enters this area, the Safety Eye sets an output, input to the robot (IN_1), which activates an override reduction below the reduced speed (*) (set value with Pilz based on Risk Analysis). This override must be set in the job program
- **Orange** area. If violated it slows the robot to speed V2. If the operator enters this area '' AND '' the robot exceeds the maximum reduced speed, the robot stops in Stop 0. The Safety Eye sets pin 1 of

X13 to 0: the VRED. In operator program I have to program the stop of the robot. In practice, the robot slows down and stops.

- **Red** area. Forbidden to the operator. Standstill active (pin 7 -SBH1). The robot must be already stopped (as requested by the yellow zone) otherwise, activating the stand still it will stop in stop 0.
- **Purple** area. Always prohibited to the operator (it cannot reach it without violating the red zone) but logically green from the robot point of view otherwise it would generate false alarms with the movements of the robot; the robot at this stage cannot get out of it. Space within which the robot moves. The safe tool can only move in this area. If the operator is in this area, the Safety Eye sets pin 8 -SBH2 or the according to Stand Still.

With the operator in the windshield loading / unloading area (the rack), the robot is stopped in STAND STILL from **Red** area.

From the initial tasks sequence previously defined (here detailed with the addition of the corresponding safety mode of the robot) a graphical sequence of the main robot-human interaction phases are designed.

Table 1: Challenge 2 task analysis

	Robot's Task	HRC Mode	Safety and information system	Operator's task
1	Picks up one windshield, groups, rotates toward the operator remaining in violet zone.	SMS – High speed standard	Operator is informed of the next operation and prepares components for the assembly (MR assisted).	Other operations on the workcell: Logistics, HMI
2	The robot extends from grouped position and to extended one; it goes to an interactive assembly position (defined by anthropometric adaptation). The CpSoSaware application can here adjust the eight of the gripper position to the golden zone according to operator anthropometrics.	SSM		
3	Stays in a stationary position offering counterforce to assembly operations in golden zone.	Safe stop	The operator has all the components ready for integration and performs cyclic repetition of assembly operations assisted by the MR system for different components with particular attention for new/updated components and procedures till a "continue" command".	visual check or other preliminary operations on the windshield on the gripper.
				Goes to logistics containers
				Picks up the first towel and sensor
				Performs the assembly
				Goes to logistics containers
7	Releases "safe Stop" position and continues operations.	HG/Safe Stop	The operator is advised that the robot starts moving and to go in safe zone.	Releases the robot and exits the interactive zone.
8	Assembles the windshield to the chassis.	SMS – High speed standard	The MR HMI confirms end of operation and next cycle.	Performs other operations on the workcell.

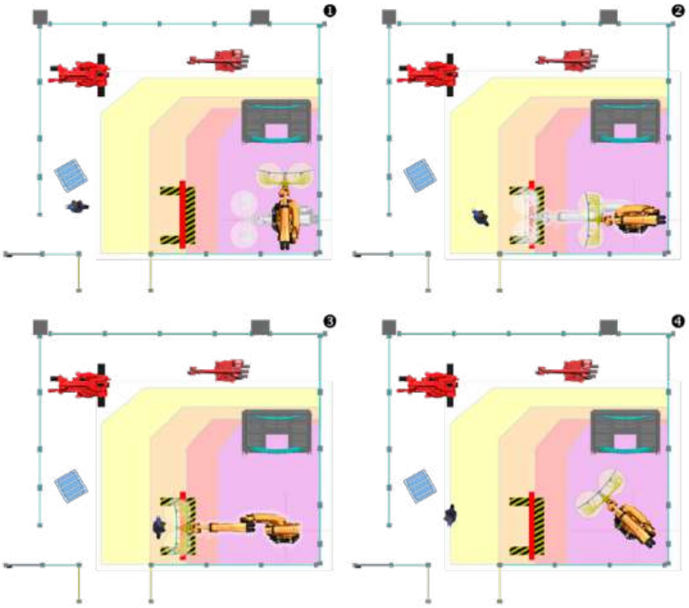


Figure 9: Main robot tasks sequence – Explanation in Table 2 below.

Table 2 describes and comments the phases depicted in the figure above.

Table 2: Main robot/operator tasks sequence (sequence referred to Figure 9)

ROBOT tasks	OPERATOR
a) Pick up windshield b) It moves into a grouped position c) Rotates towards the operator's work area	Performs other operations in the cell while remaining outside the "risk" areas.
d) From the grouped position the robot stretches up to the operational area of the operator	Approaches the direction gripper front without violating the red zone.
e) The main axes of the robot must stop or limit to the maximum. The operator has three types of interaction: Hand Guiding with gripper orientation a few degrees; Assembly of sensors and rear-view mirrors a stationary robot that acts as a "table". Other operations away from the robot. The robot must only move in Hand Guiding	The operator performs a series of assembly operations on the windscreen. Between one and the other it goes to take components from logistics container; between one and the other can perform a repositioning in HG of gripper.
f) The robot performs tasks a) and b) in reverse going to put the windshield on the stand (in the real application it would go to glue on the chassis and then take one other)	The operator does the opposite cycles 2 and 1 returning to logistics

The above Safety instructions have been implemented at low level on the robot, tested for reliability and integrated with the "Safety Eye" Safety zone programming control.

2.1.2 Workcell setup, and 1st functional and reachability testing

The cell has been constructed according to the description in the previous paragraphs. Figure 10 below show the running operations and integration phases.

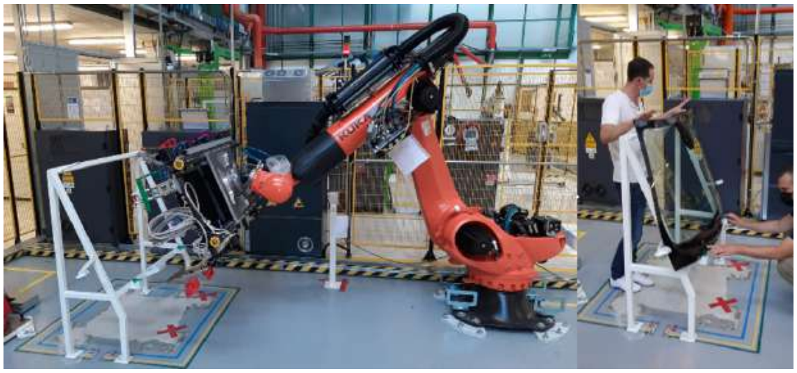


Figure 10: **CRF Robot workcell installation.**

The scenarios are integrated in the work cell described above, both to ensure safety for the operator, and to integrate a higher level of ergonomic analysis and adjustment of the equipment in real time, guaranteeing an adaptive level of adjustments by improving ergonomics for the operator.

To partially validate these concepts, a series of small-scale trials have been performed before integration.

First of all, we considering all requirements for the use case like Safety, feasibility, ergonomics and so on.

Regarding **safety**, the use of ergonomic analysis is such to optimize the workplace (previous analysis and description) and to define the initial position of the tools with respect to the operator, respecting the complete Golden Zone[2] with:

- ✓ Initial recognition of the height starting from the anthropometric classification of the operator.
- ✓ Adaptation of the robot and gripper position according to a predefined table of setup positions and related trajectories.
- ✓ Ergonomic analysis to optimize the workstation in a virtual environment, to obtain a trajectory of the robot, in a real environment, as much as possible adapted to the human.

Active real-time adaptation of robot trajectories during use case phases will not be performed; robot trajectories will be pre-planned and validated in VR, but not reactively adapted during Human-Robot interaction. Once the system recognizes the operator's anthropometry, it identifies the correct final gripper position in the Golden Zone and resumes the pre-planned trajectory to reach the final position. The purpose is to avoid programming the robot with collision avoidance techniques that might be needed in the case of adaptive trajectories. Some tests in VR were used to identify the assisted and cognitive training method aimed in safety.

[2] The Golden Zone is the ideal work area for the operator, which guarantees the reduction of activities without added value and of difficult or unnatural operations, keeping the materials and tools in specific places, in front of or on the side, in the correct arrangement so that they are taken in the normal order of use, reducing fatigue, increasing productivity.

2.1.3 Anthropometrics definition

Mechanical hazards are not only those that can harm an operator during the human-robot collaboration, but also include the lack of consideration of ergonomics that can lead to physical and/or cognitive stains. Physical ergonomics is concerned with human anatomy, anthropometric, physiological, and biomechanical characteristics as they relate to physical activity. When an operator is physically interacting with a robot, trying to accomplish a task, his/her posture is inevitably influenced by the robot's movement. Although humans are not controllable, robots are. In this way, robots should be programmed so as to adapt their collaborative trajectory based on human posture **Error! Reference source not found.**.
A system able to be adapted according to human anthropometrics is dependent on several components. Firstly, the system has to receive some input from the operator in order to be able to adapt. This input could be actively provided by the operator, however, since the aim of a CPSoS is to relieve the operator and make the robot responsible for autonomously adapting, monitoring of the human is the most preferable method. This requires sensors of one or more types. Also, a control system, probably consisting of several submodules, is needed for processing the input received from the sensors, decision making, and generation of control output for a collaborative robot.
The main goal of the system is the optimization regarding the ergonomic conditions. The operator's posture is an essential aspect of ergonomics and its accurate detection (via a monitoring system) is inevitably important. The monitoring system includes both the sensors and the processing module. A posture processing module should be able to calculate information about the operator’s posture based on the sensor's input. Subsequently, an ergonomic estimation module should be able to estimate some measure of the ergonomic state of the operator. Based on the ergonomic information a decision-making module should find the right set of actions in order to minimize strain and ergonomic risk factors based on a robot response model. Finally, a robot control module should send control information to the robot such that the wanted set of actions is carried out **Error! Reference source not found.**.

The pipeline of the control system is presented in Figure 11.

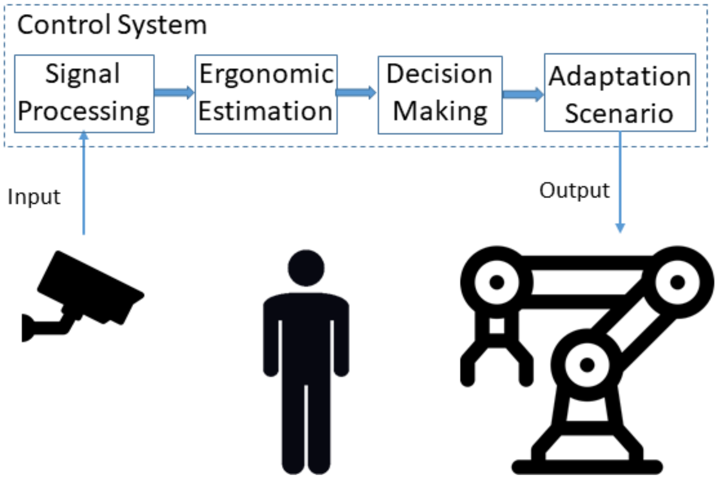


Figure 11: **Overview of the control system’s steps.**

A requirement for using a non-invasive vision-based pose estimation method is to be able to extract outputs like key points (landmarks) or a "skeleton" model of the human, in real-time. The open-source method for the landmark extraction of the human's posture, which is used in this task, is the OpenPose algorithm. OpenPose is able to detect multiple humans, it can run both on the CPU and GPUs and it can be used to detect landmarks in both 2D and 3D. OpenPose also enables the use of different pre-trained pose models. The most commonly used formats are the BODY_25 (25 landmarks) and the COCO (18 landmarks).

In manufacturing, the work of the robots is performed as a sequence of actions that are already known and pre-defined. A parameter that can affect the robot's movement is the body of the operator (height). This parameter does not change the trajectory of the robot's action but can be used in order to select the best adjustment of the robot's position and configuration in order to optimize the ergonomics assessment of the operator. The adjustment happens only one time at the beginning of the robot's operation. For security reasons, it is not convenient to have a dynamic adjustment.

Taking into account the previous description, this task focuses on exploring computer vision for ergonomic assessment with the purpose of monitoring the operator and adapting the position of an assistive collaborative robot in order to improve the working environment for the human, by improving the operator's posture in the work environment. The reference scenario is related to the safety aspects. It is used as a pre-processing step of the ergonomic analysis to optimize the workplace and define the initial position of the robot with respect to the operator's anthropometrics. More specifically, the steps that are followed are described below:

- Initial estimation of the operator's height and operator's anthropometrics classification.
- Adaptation of the robot's position and relative trajectories according to predefined scenarios based on the operator's classification.

An RGB camera is used to monitor the human's actions and then a pose estimation algorithm is running to extract the posture landmarks and also to calculate the current anthropometric state, in real-time. A total of 3 different classifications are used based on the operators' height, leading to 3 adaptable robot responses (adaptation scenarios) to allow the human to work in an optimal ergonomics state. The algorithmic framework identifies the operators' class, based on their height, and the corresponding selected scenario configures the cooperating robot's movement parameters to adapt to the environment in a way that is ergonomically most comfortable for the interacting user. To mention here that it is required not to perform an active real-time adaptation of the robot's trajectories during the use-case phases; the robot's trajectories should be pre-planned and validated in VR, but not reactively adapted during the Human-Robot interaction. Once the system recognizes the operator's anthropometrics, it identifies the proper final position of the gripper in the safety zones and picks up the pre-planned trajectory to reach the final position. The aim is to avoid programming the robot with collision avoidance techniques that might be necessary in case of adaptive trajectories.

The camera delivers input which is processed and used for generating control output for the robot. The system is meant for working in close relation to human operators and adapting to them. The system is intended to be a basic first version of a fully functional system, meaning a system adapting to the ergonomic conditions of an operator, based on visual input. The task addressed, with respect to the use case, adjusting the robot's motion based on human height. The adaptation of robotic movements will occur based on well-known ergonomic facts. Simulations in a virtual environment would be also used for further ergonomic analysis in order to optimize the workstation, so as to obtain a trajectory of the robot, in a real environment, as much as possible adapted to the human.

2.1.4 Different approaches for the estimation of the operator's height via visual analysis

Four approaches are considered and the final approach will be later identified.

Approach 1. The camera and the operator have to be directly in front of a flat 2D surface. Additionally, the camera should have always the same fixed distance and angle to the 2D surface (background). It has to be assumed that the person is flat in order for the pinhole camera concept to be used. For this approach, we need to proceed with the following processing steps.

- Calibration of the camera via a printed 2D pattern (a chessboard for example). It imports the pattern to be always as flat as possible on your background. Generate multiple images on different points of the background and try to cover the complete visible space.
- Calculate the Euclidian distance in mm / or cm (defined in calibration xml file). This step assumes that we project points on a 100 % percent flat 2D surface because our z-dimension is set to zero here.

Approach 2. An easy-to-detect "object" of a known geometry inside the picture can be used, which determines the size as some kind of comparison. Some camera parameters, like focal length, it is also needed to be known.

Approach 3. The use of 2 or more Cameras and 3D reconstruction. This lead might to higher accuracy. Furthermore, the operator could stand now everywhere in the camera's field.

- Calibration is needed.
- Use 3D reconstruction for distance calculation.

Approach 4. Use an 3D-Scanner or Kinect-System.

2.1.4.1 *Evaluation using different scenarios in the virtual environment of the simulator*

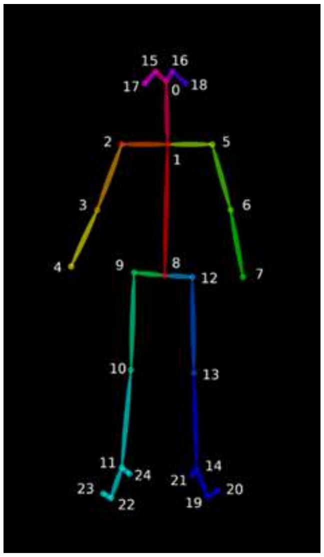


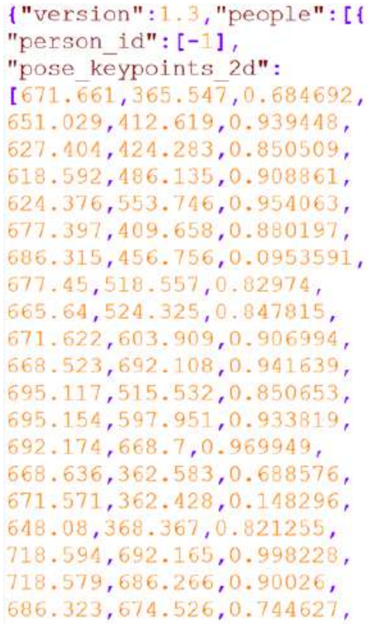
Figure 12**: The 25 landmarks of the BODY_25 model.**

For the extraction of operator’s body landmarks, the video recordings captured by the simulator were analyzed in OpenPose using the BODY_25 landmark model that tracks the following 25 landmarks (as shown in Figure 12): nose, neck, mid hip and two points for the eyes, ears, shoulders, elbows, wrists, hips, knees, ankles, heels, big toes and small toes. The outputs of the OpenPose analyses yielded **Error! Reference source not found.**:

1) JSON files for each frame of the video (see Figure 13) sequence containing pixel coordinates (origin at the upper left corner of the image) of every landmark detected in the frame.

2) A new video file in which a stick figure that represents the detected landmarks is overlaid onto the original video recording.

3) The JSON files are used for further analysis in MATLAB.



```
{"version":1.3,"people":[{
"person_id":[-1],
"pose_keypoints_2d":
[671.661,365.547,0.684692,
651.029,412.619,0.939448,
627.404,424.283,0.850509,
618.592,486.135,0.908861,
624.376,553.746,0.954063,
677.397,409.658,0.880197,
686.315,456.756,0.0953591,
677.45,518.557,0.82974,
665.64,524.325,0.847815,
671.622,603.909,0.906994,
668.523,692.108,0.941639,
695.117,515.532,0.850653,
695.154,597.951,0.933819,
692.174,668.7,0.969949,
668.636,362.583,0.688576,
671.571,362.428,0.148296,
648.08,368.367,0.821255,
718.594,692.165,0.998228,
718.579,686.266,0.90026,
686.323,674.526,0.744627,
```

Figure 13: **Example of the extracted json files.**

We assume the existence of 3 different classes based on the operator's height. More specifically, class 1 consists of operators with height < 175 cm, in class 2 the operators' height is between 175 and 185 cm, and finally in class 3 the operators are taller than 185 cm. Additionally, we apply experiments using two different positions of cameras, as presented in **Figure 1**Figure 16 and Figure 17.

First, the mean value of points 17 and 18 is estimated as well as the mean value of points 21 and 24, as presented in Figure 2.

Then, the distance (in pixels) between these two mean values is estimated, according to the Euclidean distance equation (Table 3 & Table 4). The next step is to find an approximation of the scaling factor. We calculate a scaling factor to obtain dimensionalized coordinate values from the pixel coordinates. The scaling factor (s) is calculated as:

S = distance (in pixel) / distance (in cm)

To estimate this factor, a known object with known dimensions, in a known distance of the operator is required.

To improve the object's size measurement accuracy, a proper camera calibration is necessary. Before we can measure the size of an object in an image, we first need to calibrate our system. In this task, we used a "pixels per metric" technique. Nevertheless, better accuracy can be obtained by performing a proper camera calibration by computing the extrinsic and intrinsic parameters:

· Extrinsic parameters are rotation and translation matrices used to convert something from the world frame to the camera frame

· Intrinsic parameters are the internal camera parameters, such as the focal length, to convert that information into a pixel

2.1.4.2 *Evaluation in three different operator's height scenarios for the use case 1*

Set up of this use case:

- Height of camera h_c = 3.346 m
- Distance between camera and operator d = 6.342489 m
- Position of the camera: In front of the operator

We assume that the dimensions of an object (Figure 14), which are located in a stationary location of the work environment, are known. In this example, we assume the existence of a box where the red vector represents the x-axis, the green vector represents the y-axis and the blue vector represents the z-axis. The dimensions of the box are 0.7985516 m on the x-axis, 0.94705 m on the y-axis, and 0.8202643 m on the z axis.

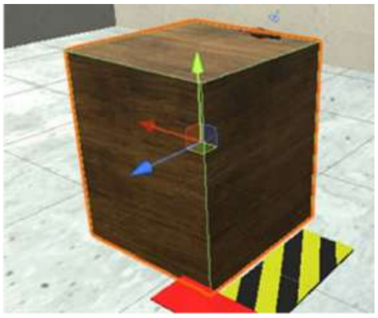


Figure 14: **Dimensions of the box located in the scene.**

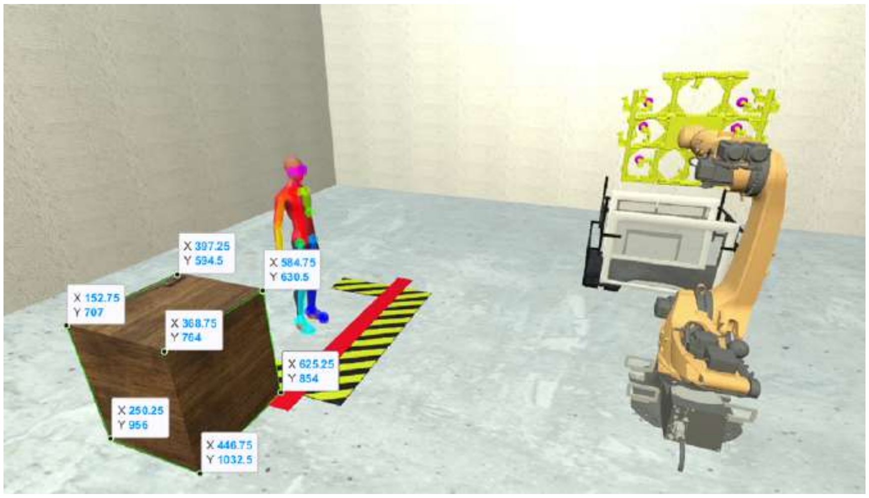


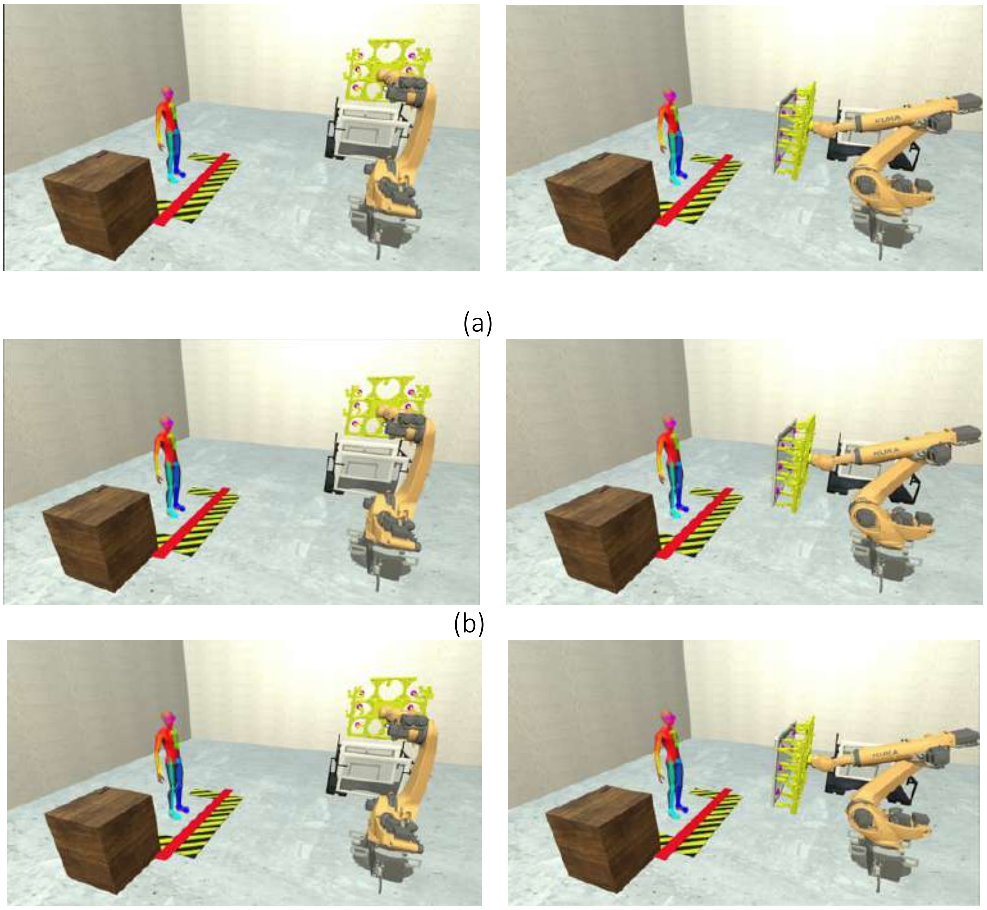
Figure 15: **Recognition of box corners and estimation of their coordinates.**

The seven visible corners of the box are recognized (Figure 15), and the (x,y) coordinates of each point are estimated. More specifically, the presented points are: p1 (368.75, 764), p2 (584.75, 630.5), p3 (625.25, 854), p4 (446.75, 1032.5), p5 (250.25, 956), p6 (152.75, 707), p7 (397.25, 594.5). Then, the distance in pixels between these points is estimated and it is divided with the real distance in cm in order to estimate the scale factor.

Side distances of the box in the x dimension
d(p1-p2) = 253.9257 → 3.1798 p/cm
d(p3-p4) = 252.4371 → 3.1612 p/cm
d(p6-p7) = 269.1403 → 3.3704 p/cm

Side distances of the box in the y dimension
d(p2-p3) = 227.1398 → 2.3984 p/cm
d(p1-p4) = 279.6002 → 2.9523 p/cm
d(p5-p6) = 267.4084 → 2.8236 p/cm

Side distances of the box in the z dimension
d(p1-p6) = 223.3943 → 2.7234 p/cm
d(p2-p7) = 190.9247 → 2.3276 p/cm
d(p4-p5) = 210.8661 → 2.5707 p/cm



(a)

(b)

(c)

Figure 16: Two frames capturing the collaboration task between operator and robot under three different heights of the operator (a) 170 cm, (b) 180 cm, (c) 190 cm. The camera is located in front and side of the operator.

To mention here that the used landmarks (i.e., points 17 and 18 of Figure 12) represent the corresponding points of the two ears. This means that the estimated height is not the total height of the human but the height from the ears of the operator to the ground. So approximately 12-15 cm, between ears and the top of the hand, are not estimated in this case but they have to be added.

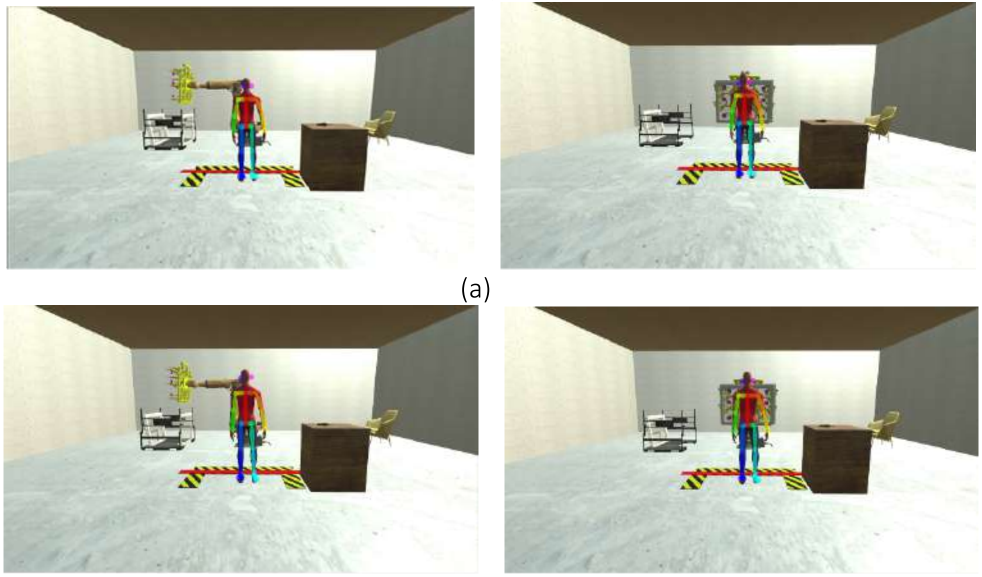
Table 3: Calculated distances (in pixel) and estimated heights (in cm) for the use case 1 under different heights of the operator (with estimated S=2.1).

	170		180		190	
	pixel	cm	pixel	cm	pixel	cm
Frame 1	3.288.501	1.565.953	3.523.984	1.678.088	3.774.702	1.797.477
Frame 2	3.288.166	1.565.793	3.524.153	1.678.168	3.774.381	1.797.324
Frame 3	3.288.030	1.565.729	3.524.135	167.816	3.774.372	179.732
Frame 4	3.287.964	1.565.697	3.524.225	1.678.202	3.774.661	1.797.458
Frame 5	3.287.828	1.565.632	3.524.176	1.678.179	3.774.644	179.745
Frame 6	3.287.689	1.565.566	3.524.264	1.678.221	3.788.777	180.418
Frame 7	3.287.768	1.565.604	3.524.214	1.678.197	3.789.002	1.804.287
Frame 8	3.287.825	1.565.631	3.524.411	1.678.291	3.788.718	1.804.151
Frame 9	3.288.111	1.565.767	3.524.351	1.678.262	3.788.911	1.804.243
Frame 10	3.288.205	1.565.812	3.524.354	1.678.264	3.789.237	1.804.399
Average	3.288.009	1.565.718	3.524.227	1.678.203	3.781.741	1.800.829

2.1.4.3 *Evaluation in three different operator's height scenarios for the use case 2*

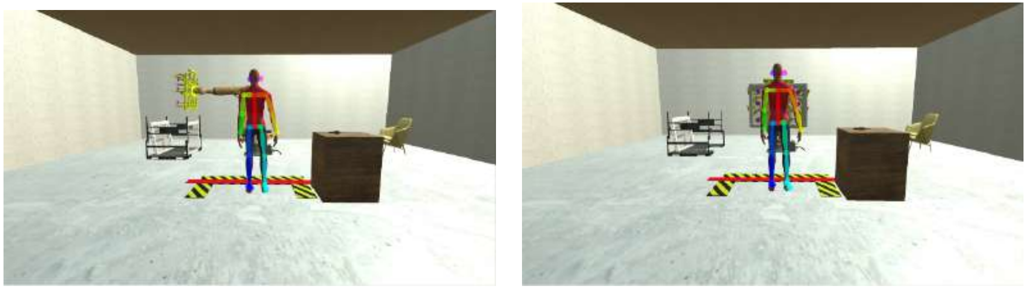
Set up of this use case:

- Height of camera h_c = 1.312904
- Distance between camera and blue point d = 3.95367
- Position of the camera: behind the operator



(a)

(b)



(c)

Figure 17: **Two frames capturing the collaboration task between operator and robot under three different height states of the operator (a) 170 cm, (b) 180 cm, (c) 190 cm. The camera is located exactly behind the operator.**

Following the same approach, as this discussed in the previous paragraph, we take the results presented in Table 4. As we can observe, the estimated heights in both use cases are very similar.

Table 4: Calculated distances (in pixel) and estimated heights in cm for the use case 2 under different heights of the operator (with estimated S=2.4).

	170		180		190	
	pixel	cm	pixel	cm	pixel	cm
Frame 1	3.723.870	1.551.613	3.987.025	166.126	4.208.545	175.356
Frame 2	3.723.852	1.551.605	3.987.350	1.661.396	4.208.514	1.753.548
Frame 3	3.724.168	1.551.737	3.987.215	166.134	4.208.575	1.753.573
Frame 4	3.723.577	155.149	3.987.090	1.661.288	4.208.631	1.753.596
Frame 5	3.723.695	155.154	3.987.061	1.661.275	4.208.655	1.753.606
Frame 6	3.723.682	1.551.534	3.973.145	1.655.477	4.208.615	175.359
Frame 7	3.710.125	1.545.885	3.987.181	1.661.325	4.208.725	1.753.635
Frame 8	3.710.254	1.545.939	3.987.246	1.661.353	4.208.615	175.359
Frame 9	3.723.850	1.551.604	3.987.262	1.661.359	4.208.560	1.753.567
Frame 10	3.723.950	1.551.646	3.987.417	1.661.424	4.208.541	1.753.559
Average	372,1102	155,0459	398,5799	166,075	420,8598	175,3582

2.1.4.4 *Observational study*

In an operation in which the user interfaces with robots or tools additional safety evaluations are necessary.

In the reproduction in the virtual environment particular attention was considered from the point of view of safety.

It has been evaluated if the distances that are required by the standard (ISO 13855 and ISO 15066 for collaborative robots), are sufficient also from the behavioural point of view, i.e. if the user feels comfortable outside that area and how he performs the approach to the collaborative robot.

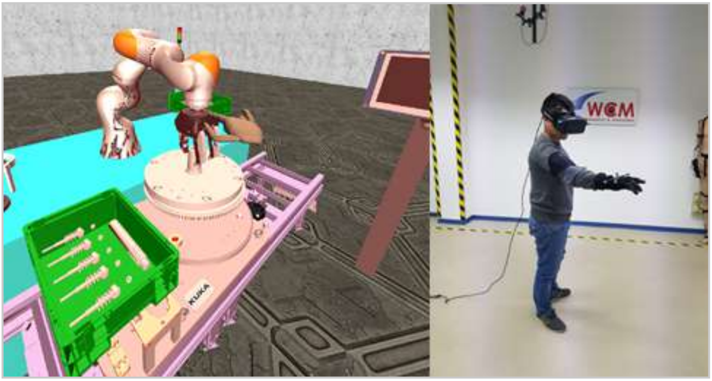


Figure 18: **Simulation and preliminary evaluation of the operator and collaborative robot contact phases. The robot stops and the red light comes on**

Ergonomics is related to safety as it refers to long term health problems. Ergonomics is the approach to study and optimize the body physical parameters of the operators during his repetitive tasks. It analyses all the operator's tasks in search of wrong postures, positions, and loads in order to avoid them and give support to the operator.

Factors that affect the operator's ergonomics are:

- Weight of lifted objects (usually supported by mechanical lifters called partners or manipulators).
- Repetition rate of movements.
- Wrong postures introducing weird awkward body angles (e.g. too much bending, operations with elbow above shoulders for a long time).

All the above situations require design workplace optimization or the use of additional supporting tools or procedures. The use of Collaborative Robotics itself is often approached thanks to its capability to support operators in heavy load lifting and in performing repetitive tasks.

As previously mentioned, regarding the ergonomics analysis tools used to optimize the task, CRF use the Xsens motion capture systems for body movements and the Cyberglove for fine hand movements. These systems could be use both in our VR system and in mixed reality contest.

Xsens MVN Biomech system is a motion capture system based on 17 interoperating inertial sensors placed on a «strap» suit. The Xsens system returns as output the values of rotation around X, Y and Z axis for each body joint.

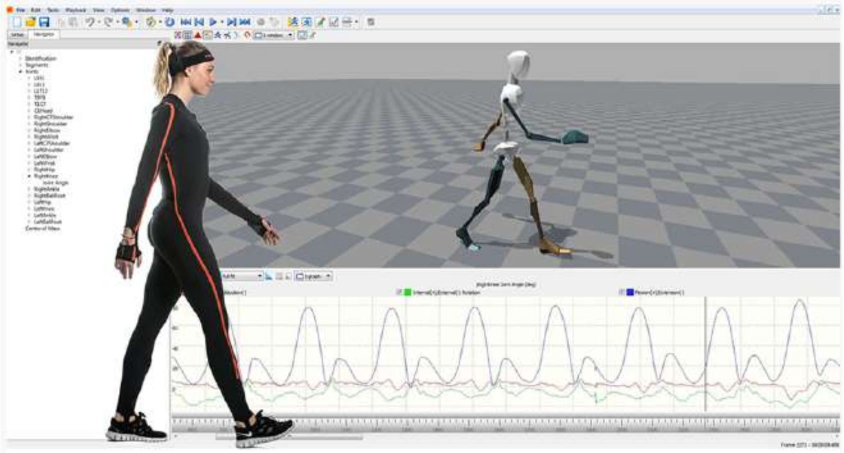


Figure 19: Xsens systems representation

CRF, has developed the software VTA to assist the ergonomic analysis; it is focused to a detailed and objective evaluation of the working activities carried out on a workstation, conducted as required by international and corporate standards.

The VTA software can import the output data of Xsens and the standard file of the Biovision Hierarchy (.BVH), in order to automatically evaluate the postures assumed by the user during the whole working task.

In the CPSoSaware scenario CRF will use (by eventually adapting) the VTA software with the aim to evaluate ergonomic indexes like CL Ocra, OWAS, Rula, etc.

These features will be studied by CRF and UPAT; the system is planned to perform the body recognition by cameras and output the BVH format information; the system will furnish, step by step evaluation of ergonomics indexes of interest using the adapted algorithms from the VTA software.

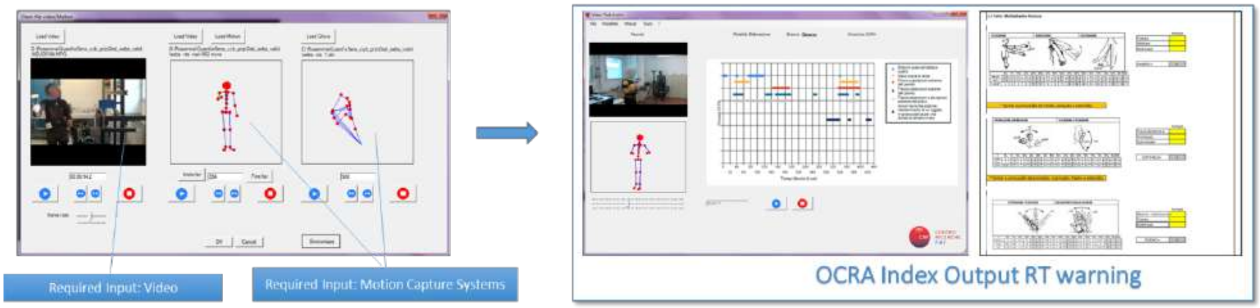

Required Input: Video
Required Input: Motion Capture Systems
OCRA Index Output RT warning

Figure 20: **VTA idea**

The tool will be made joining information on Safety and ergonomics analysis; it's working cycle will produce? a continuous flow of video images (partially pre-elaborated) to generate RT warnings related to wrong postures, ergonomics risks and so on. This information will be communicated to the operator in MR.

The functionalities available in the basic and medium application are implemented and preliminary tested.

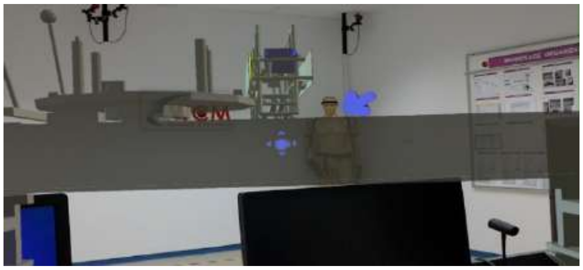


Figure 21: **Pointer for Gaze monitoring**

Gaze: monitored through a pointer (Figure 21) in the hololens; it will change morphology if user is looking at objects or not. A Pointer is smart; if the hand is detected it changes to hand representation.

- Object aware
- Hand detection
- Hand command detection

Gesture: The hand represented within the scene is recognized in certain defined postures as Tap, Pointer and open hand. The tap gesture also controls various actions such as the movement of objects.

- Object manipulation without controllers by hand
- Rotation
- Translation

Spatial mapping: is a feature which provides a representation of real-world surfaces around the device. This can be used by application developers to make the applications environment aware.

- Space scanning time settable
- Planes detection
- Object placement with dimension control

Tests in laboratory were made to evaluate these features (see Figure 22 and Figure 23)

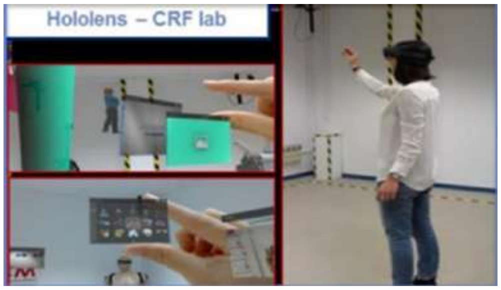


Figure 22: **Gaye and Gesture Usage**

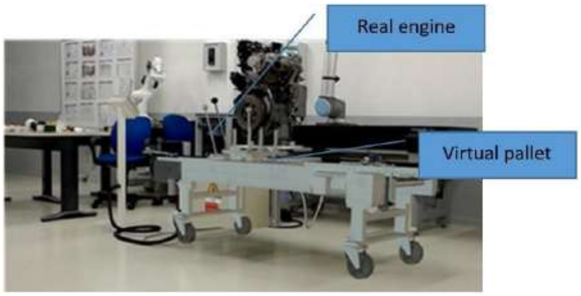


Figure 23: **Virtual CAD in real environment**

The training on the job MR tools that is implemented and under continuous development and study in the CPSoSaware scenarios is structured as in following figure:

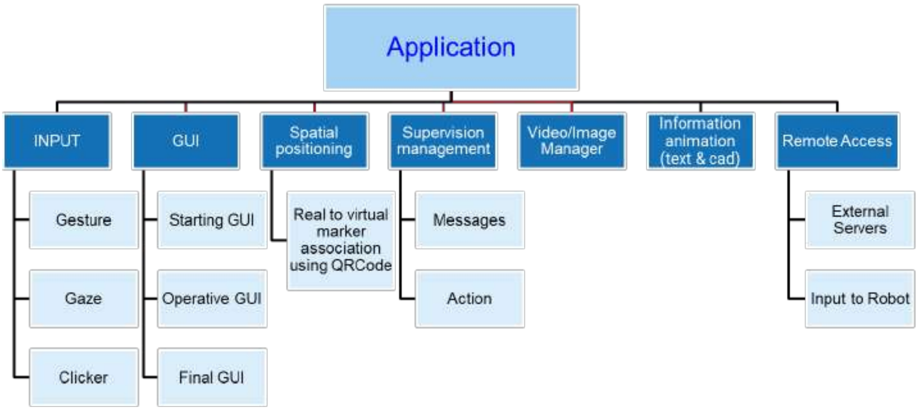


Figure 24: **Training on the job MR tools**

2.2 Connected and Automated Vehicles

CPSoSAware components are developed to be applied and tested in both Industrial Pillar and Automotive Pillar. The automotive pillar, is further decomposed into three use-cases: the Human in the Loop, Cybersecurity and Cooperative Awareness, which are described in the subsections 2.2.1, 2.2.2, and 2.2.3 correspondingly.

2.2.1 Human in the loop control use case in single vehicle scenario

Catalink has developed a Driver State Monitoring (DSM) Android application which aims at providing a real-time monitoring and assessment of the drowsiness level of a driver. Considering that, the most available and most exploitable object nowadays is undoubtedly the smartphone, the main idea of the proposed methodology is to utilise this availability and propose a real-time DSM solution without the need of introducing several external and potentially obtrusive devices. The application uses the front camera of the smartphone in order to monitor throughout the whole duration of the driving session the general status of the driver and to estimate his/her fatigue level.

The estimations are conducted utilising Google's ML Kit[3], a standalone library which offers the possibility of on-device machine learning processing. More specifically, ML Kit provides the means of integrating machine learning capabilities into an application, through the exposure of the so-called vision APIs. Those APIs refer to both video and image analysis and can be exploited in a great deal of use cases such as face detection, barcode scanning, pose detection, text recognition etc. Within the DSM application ML KIT was utilised to achieve a fast, efficient, and real-time face recognition and facial landmarks extraction.

The analysis is taking place in a per frame basis where for each incoming video frame we are trying to determine i) if the driver has his/her eyes closed (either one of them or both of them), ii) if the driver is yawning and iii) if the driver is looking left or right with respect to the camera. The core functionalities of the DSM application are given in detail in D3.1.

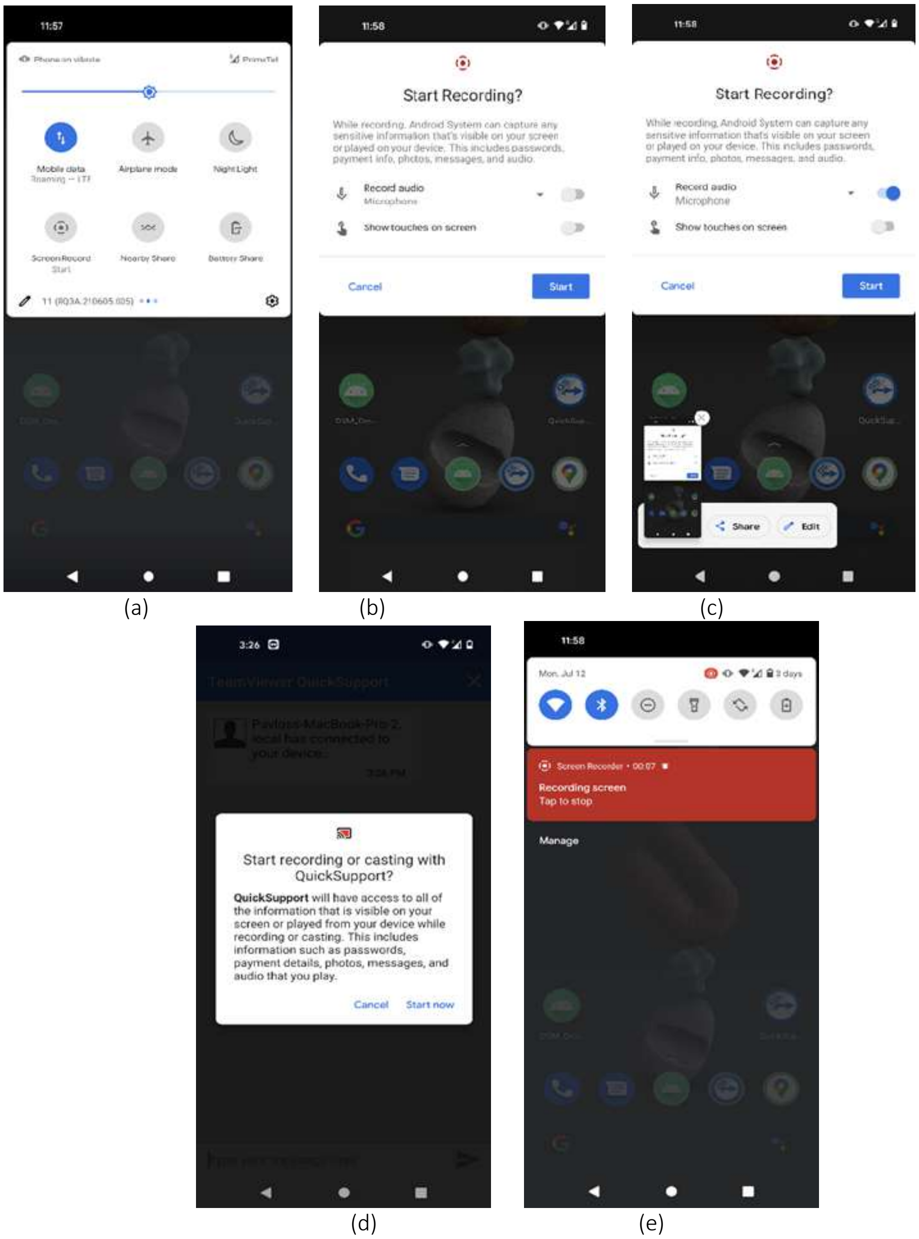
2.2.1.1 Evaluation concept of the targeted components and use case

The validation of the DSM Android application was conducted through a set of on-road experiments which was undertaken by the teams of Catalink and Robotec. The requirement for those experiments was the existence of a smartphone powered by Android with the minimum Android version being the 8.0 (Oreo). The participants were asked to place the device on the dashboard of their car, in such a place where their face would be clearly visible from the front camera of the phone. Afterwards, they were asked to initiate a small driving session and test the following scenarios: i) keep their eyes closed for a few seconds, ii) yawn several times for a few minutes and iii) look right or left for a few seconds.

[3] https://developers.google.com/ml-kit/

2.2.1.2 Testing Procedure and Data Acquisition

The interaction of the users with the Android DSM application was straightforward, with just one step needed by their side begoring starting the application. In order to be able to collect the videos of the driving sessions, for enabling the possibility of a later analysis of those, the users were asked to start a recording screen session from their phone following the procedure depicted in Figure **25**.



(a)

11:57

(b)

11:58

Start Recording?

While recording, Android System can capture any sensitive information that's visible on your screen or played on your device. This includes passwords, payment info, photos, messages, and audio.

Record audio — Microphone: off

Show touches on screen: off

(c)

11:58

Start Recording?

While recording, Android System can capture any sensitive information that's visible on your screen or played on your device. This includes passwords, payment info, photos, messages, and audio.

Record audio — Microphone: on

Show touches on screen: off

(d)

3:26

Pavlos's-MacBook-Pro-2.local has connected to your device. 3:26 PM

Start recording or casting with QuickSupport?

QuickSupport will have access to all of the information that is visible on your screen or played from your device while recording or casting. This includes information such as passwords, payment details, photos, messages, and audio that you play.

(e)

11:58

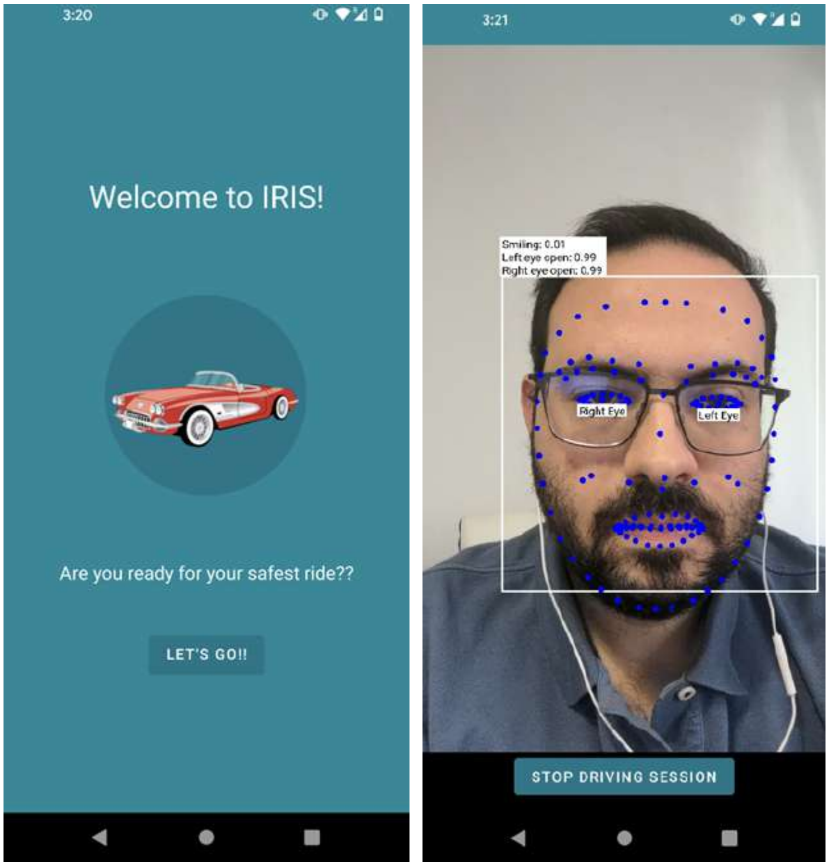
Mon, Jul 12

Screen Recorder • 00:07

Recording screen
Tap to stop

Figure 25: (a,b) Step-1: Dropdown menu and select “Screen Record”. (c) Step-2: Allow the recording of audio. (d) Step-3: Press " Start now”. (e) Step-4: The recording should start

Once the user opens the application, he/she will see a welcome screen which prompts him/her to initiate the driving session as depicted in Figure 26-a. The user should press the button “Let’s go!!” in order to start the session and by doing that the front camera of the smartphone will be engaged and the detection of the face and the facial landmarks of the user will be visible on the screen as depicted in Figure 26-b.



Welcome to IRIS!

Are you ready for your safest ride??

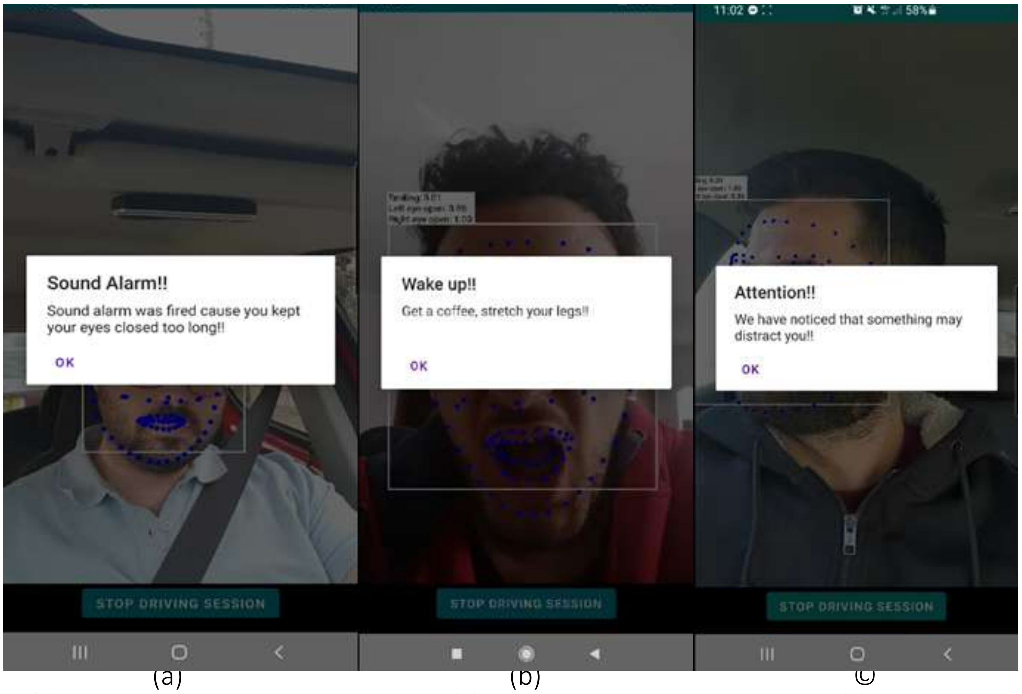
Smiling: 0.01
Left eye open: 0.99
Right eye open: 0.99

(a) (b)

Figure 26: (a) Welcome screen of the DSM application. (b) The front camera is engaged, and the face detection has been initiated.

The scenarios that the user should check are three and they are listed below:

1. ***Eye closure***: In this scenario the user should keep his/her eyes closed for a few seconds (approximately 2-3sec). The application should raise a sound alert accompanied with a message on the screen as the one shown in Figure 27-a
2. ***Yawning***: In this scenario the user should yawn as he/she would normally do. The application should raise a prompt message advising him/her to take a break as depicted in Figure 27-b
3. ***Distraction***: In this scenario the user should look left/right for a couple of seconds. The application should raise a prompt message notifying the user that he/she is distracted as depicted in Figure 27-c



Sound Alarm!!

Sound alarm was fired cause you kept your eyes closed too long!!

OK

Wake up!!

Get a coffee, stretch your legs!!

OK

Attention!!

We have noticed that something may distract you!!

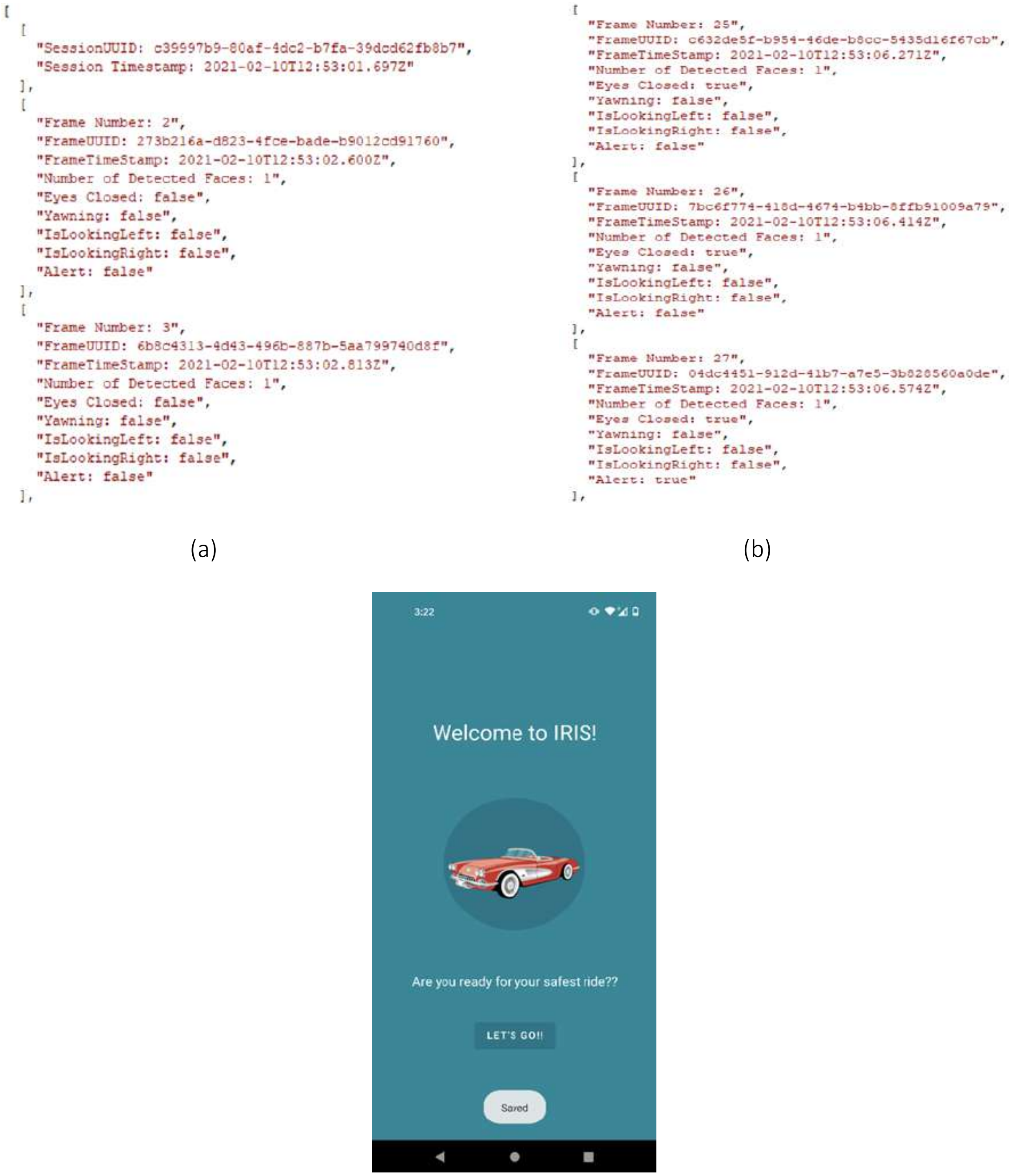
OK

(a) (b) ©

Figure 27: (a) Result of the “Eye closure” scenario. (b) Result of the “Yawning” scenario. (c) Result of “Distraction” scenario.

The user should not interact with the messages on the screen since they will disappear after a few seconds automatically.

Once the driving session is over the user should press the “Strop driving session” button in order to save the corresponding JSON file (Figure 28-a). The user should see the main page of the application and a Toast message saying “Saved” as it is depicted in Figure 28-b.



```
[
  [
    "SessionUUID: c39997b9-80af-4dc2-b7fa-39dcd62fb8b7",
    "Session Timestamp: 2021-02-10T12:53:01.697Z"
  ],
  [
    "Frame Number: 2",
    "FrameUUID: 273b216a-d823-4fce-bade-b9012cd91760",
    "FrameTimeStamp: 2021-02-10T12:53:02.600Z",
    "Number of Detected Faces: 1",
    "Eyes Closed: false",
    "Yawning: false",
    "IsLookingLeft: false",
    "IsLookingRight: false",
    "Alert: false"
  ],
  [
    "Frame Number: 3",
    "FrameUUID: 6b8c4313-4d43-496b-887b-5aa799740d8f",
    "FrameTimeStamp: 2021-02-10T12:53:02.813Z",
    "Number of Detected Faces: 1",
    "Eyes Closed: false",
    "Yawning: false",
    "IsLookingLeft: false",
    "IsLookingRight: false",
    "Alert: false"
  ],
```

(a)

```
[
  "Frame Number: 25",
  "FrameUUID: c632de5f-b954-46de-b8cc-5435d16f67cb",
  "FrameTimeStamp: 2021-02-10T12:53:06.271Z",
  "Number of Detected Faces: 1",
  "Eyes Closed: true",
  "Yawning: false",
  "IsLookingLeft: false",
  "IsLookingRight: false",
  "Alert: false"
],
[
  "Frame Number: 26",
  "FrameUUID: 7bc6f774-418d-4674-b4bb-8ffb91009a79",
  "FrameTimeStamp: 2021-02-10T12:53:06.414Z",
  "Number of Detected Faces: 1",
  "Eyes Closed: true",
  "Yawning: false",
  "IsLookingLeft: false",
  "IsLookingRight: false",
  "Alert: false"
],
[
  "Frame Number: 27",
  "FrameUUID: 04dc4451-912d-41b7-a7e5-3b828560a0de",
  "FrameTimeStamp: 2021-02-10T12:53:06.574Z",
  "Number of Detected Faces: 1",
  "Eyes Closed: true",
  "Yawning: false",
  "IsLookingLeft: false",
  "IsLookingRight: false",
  "Alert: true"
],
```

(b)

(c)

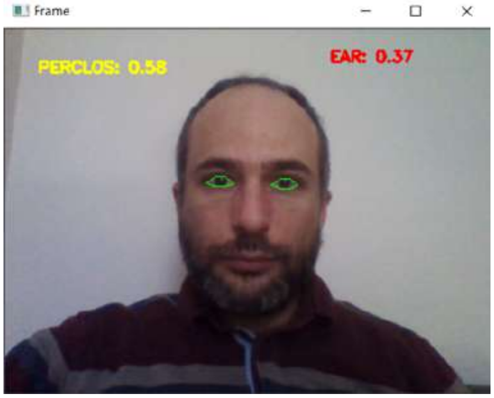
Figure 28: (a-b) Generated JSON file. (c) Session is over and the JSON is saved

Hence, as arises from the above, the data that we are acquiring from a conducted driving session is a recorded video and the JSON. Both of the files are stored internally on the phone and they can be retrieved and used for further analysis.

UPAT has developed a drowsiness detection implementation that provides real-time monitoring of users/drivers and it estimates their drowsiness based on metrics (e.g., EAR, PERCLOS) proposed by the literature (More details about the implementation have been discussed and presented in D3.1). Additionally, when the driver has his/her eyes closed (both or either one of them) for more than 48 frames, then an appropriate alert (i.e., sound and visual text) informs him/her about this dangerous situation. In this section, we will present and evaluate the performance of the application under different real-life conditions. More specifically, we established two sets of experiments that are performed in in-door (set of Figures A-D) and out-door (set of Figures E-H) environments. Other parameters that we take into consideration are:

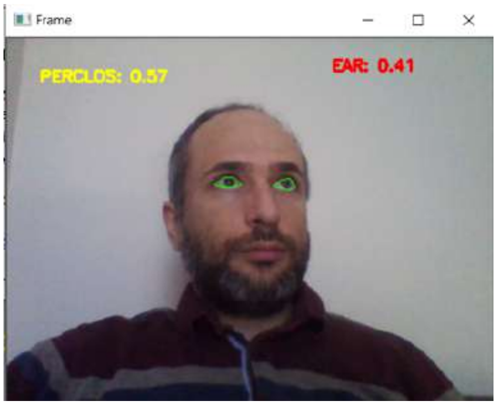
- Close and from distant (in Figures D) monitoring of the user’s face
- User wearing (In Figures C-H) or not glasses (In Figures A-B)
- Different directions of the user’s head (In all the set of Figures)
- Open eyes, one open eye (only the left or only the right) (in Figures B), and closed eyes for more than 48 consecutive frames in the row (In all the set of Figures)
- Sunlight behind (in Figures F & H) or in front of (in Figures E, G & H) the user’s face
- Complicated patterns of background (in Figures F)
- Mixed sunlight conditions (shadows and light) (in Figures G)
- Capturing during motion of the user through dynamically changed light conditions (in Figures H)

A) In-door, close monitoring without glasses in different directions of the user’s head



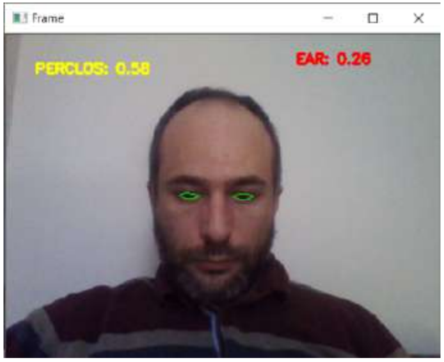
PERCLOS: 0.58
EAR: 0.37

(A-1)



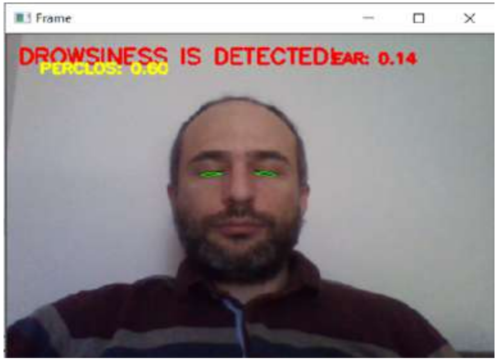
PERCLOS: 0.57
EAR: 0.41

(A-2)



PERCLOS: 0.58
EAR: 0.26

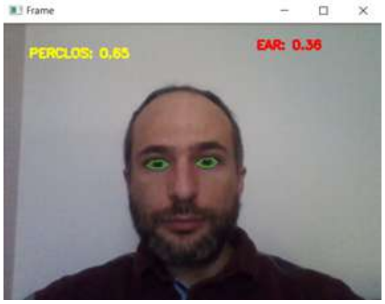
(A-3)



DROWSINESS IS DETECTED
PERCLOS: 0.60
EAR: 0.14

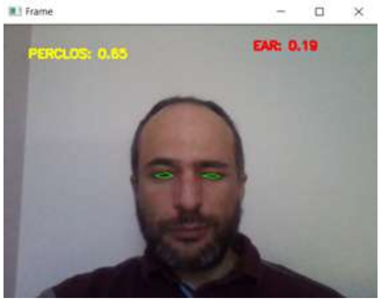
(A-4)

B) In-door monitoring with one open eye



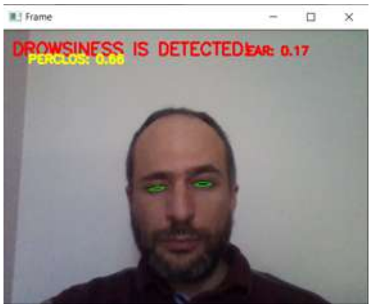
PERCLOS: 0.65
EAR: 0.36

(B-1)



PERCLOS: 0.65
EAR: 0.19

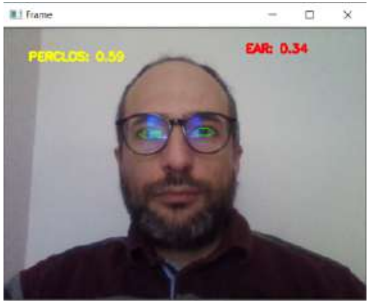
(B-2)



DROWSINESS IS DETECTED
PERCLOS: 0.66
EAR: 0.17

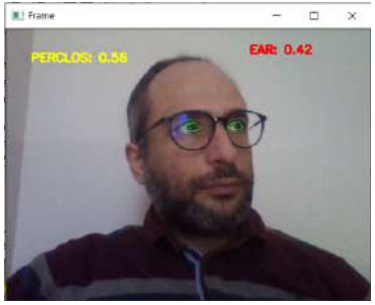
(B-3)

C) In-door, close monitoring where the user wears glasses



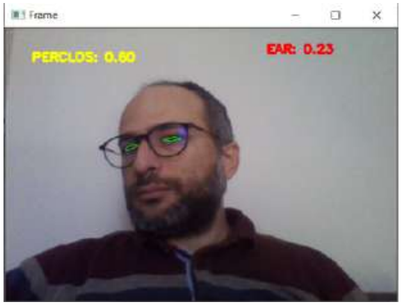
PERCLOS: 0.59
EAR: 0.34

(C-1)



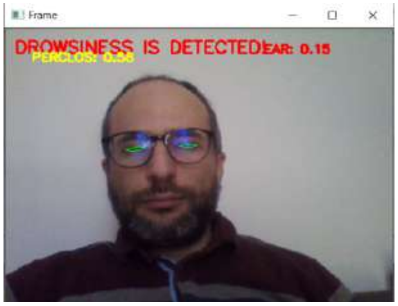
PERCLOS: 0.56
EAR: 0.42

(C-2)



PERCLOS: 0.60
EAR: 0.23

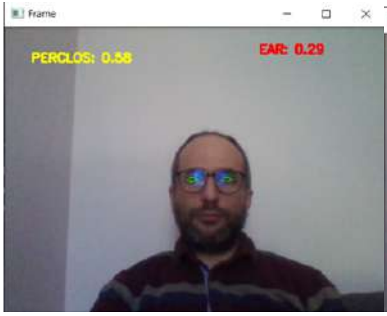
(C-3)



DROWSINESS IS DETECTED
PERCLOS: 0.56
EAR: 0.15

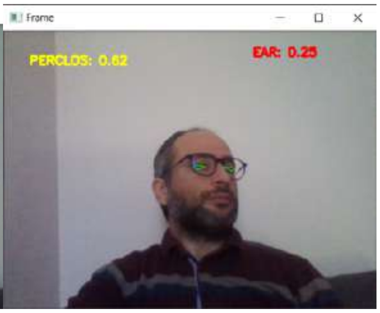
(C-4)

D) In-door monitoring from distant



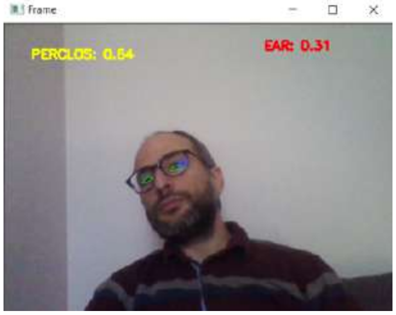
PERCLOS: 0.58
EAR: 0.29

(D-1)



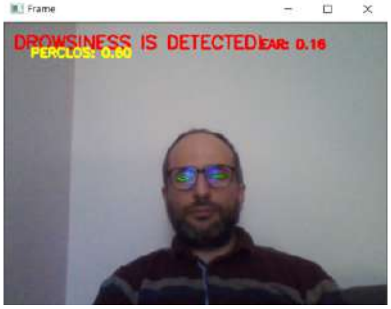
PERCLOS: 0.62
EAR: 0.25

(D-2)



PERCLOS: 0.64

EAR: 0.31



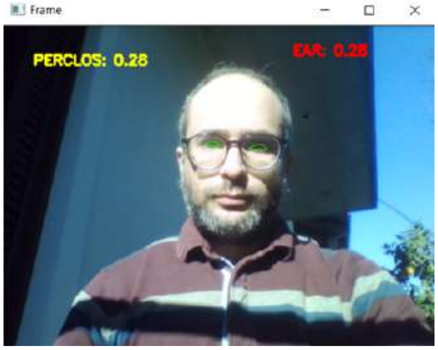
DROWSINESS IS DETECTED

PERCLOS: 0.60

EAR: 0.16

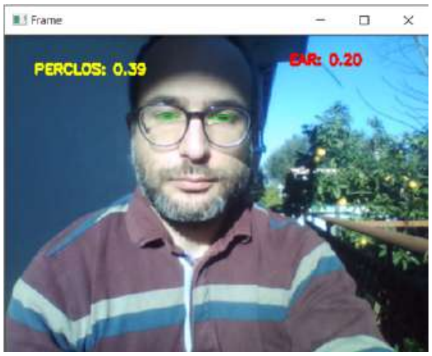
(D-3) (D-4)

E) Outdoor monitoring with sunlight in front of the user's face



PERCLOS: 0.28

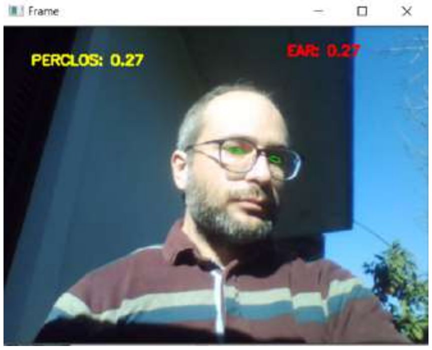
EAR: 0.28



PERCLOS: 0.39

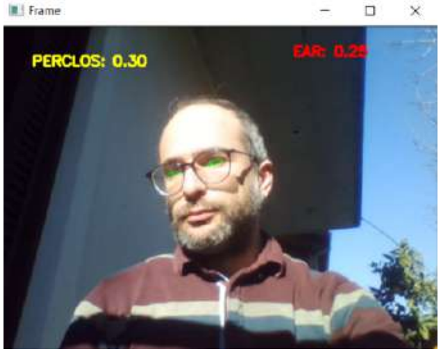
EAR: 0.20

(E-1) (E-2)



PERCLOS: 0.27

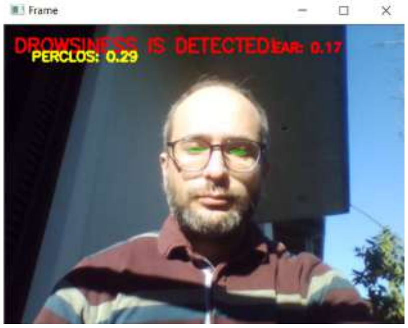
EAR: 0.27



PERCLOS: 0.30

EAR: 0.25

(E-3) (E-4)



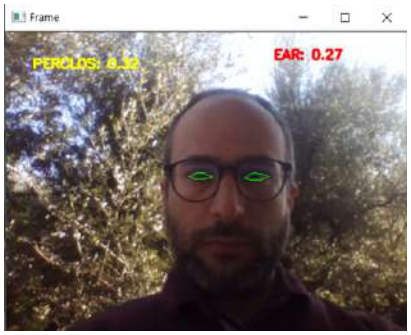
DROWSINESS IS DETECTED

PERCLOS: 0.29

EAR: 0.17

(E-5)

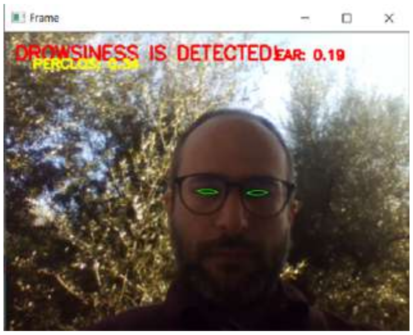
F) Outdoor monitoring with sunlight behind the user's face in a complicated pattern of background



PERCLOS: 0.32

EAR: 0.27

(F-1)



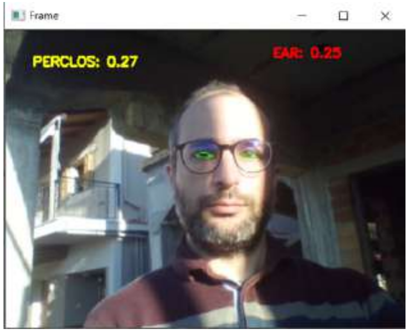
DROWSINESS IS DETECTED

PERCLOS: [illegible]

EAR: 0.19

(F-2)

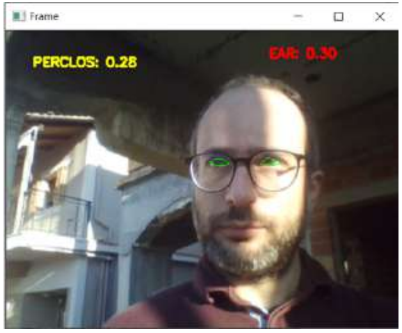
G) Outdoor monitoring in mixed light conditions (shadow and sunlight)



PERCLOS: 0.27

EAR: 0.25

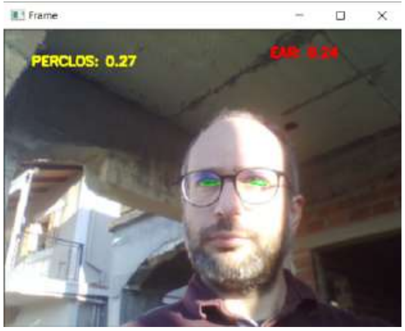
(G-1)



PERCLOS: 0.28

EAR: 0.30

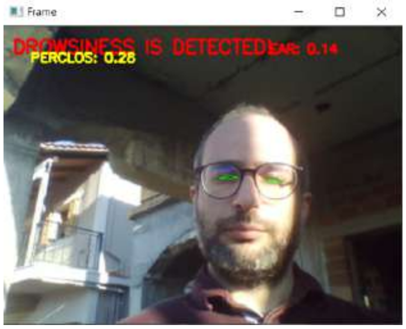
(G-2)



PERCLOS: 0.27

EAR: 0.24

(G-3)



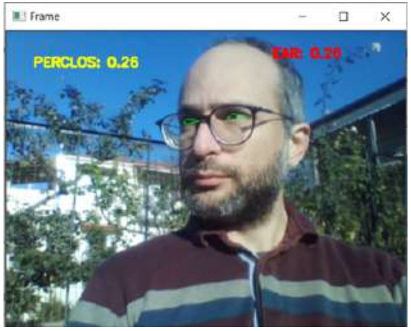
DROWSINESS IS DETECTED

PERCLOS: 0.28

EAR: 0.14

(G-4)

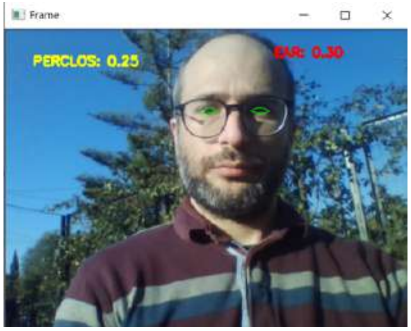
H) Outdoor monitoring while the user moves under dynamically changed light conditions



PERCLOS: 0.26

EAR: 0.26

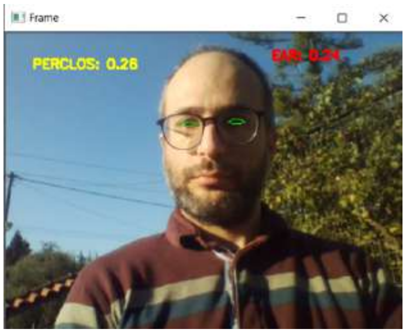
(H-1)



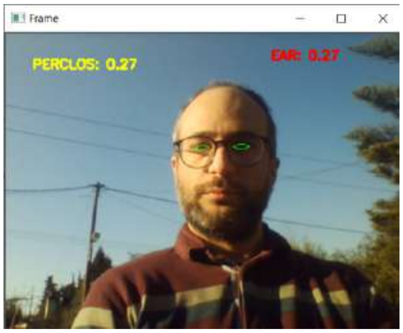
PERCLOS: 0.25

EAR: 0.30

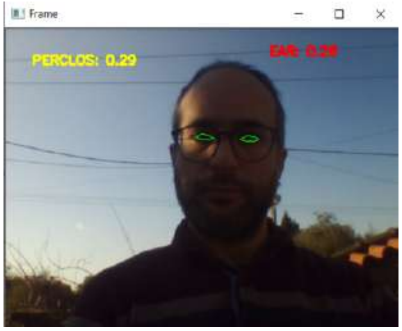
(H-2)



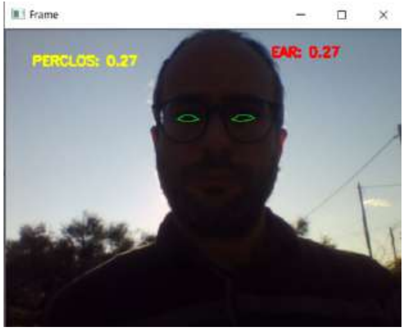
(H-3)



(H-4)



(H-5)



(H-6)

2.2.1.2.1 Simulator laboratory test procedures

One of the main methods of validating DSM solutions is laboratory testing in the simulator environment. The use of a simulator allows manipulating the driver state while providing safe test conditions and provides a possibility to collect multiple measures, depending on the chosen method of testing.

Drowsiness testing procedures refer to long, monotonous tasks with the use of a driving simulator and are usually conducted during the night (i.e., between 11 PM and 5 AM) and to register drowsiness and microsleep events. The use of car simulators in drowsiness-related studies serves to provide the safety of the participants. This is especially important when considering the validation of the driver monitoring systems against high levels of the Karolinska Sleepiness Scale (KSS), when the risk of road accident occurrence grows. Experimental methods for driver distraction testing include controlled manipulation of driver attention with the use of the so-called secondary task (not related to driving). One of the most popular types of such tasks is a visual-manual task that engages two important senses: visual and haptic channels (e.g., infotainment system operating, texting). This task not only requires cognitive resources but also degrades road observations and steering wheel control.

The DSM solutions described in the chapter above will be validated in simulator conditions, in scenarios designed to test both drowsiness and distraction detection features of the DSMs.

To provide the most reliable data, the testing with the use of a simulator may be conducted in two sessions: a day and a night one. The day session serves as a baseline and the night session allows to observe the natural process of decreasing alertness and growing drowsiness of the subjects.

Research instruments applied in the simulator scenario will include:

- tested DSM applications,

- KSS keypad,
- a visual-manual task,
- eye tracking with Smart Eye Pro system,
- context camera,
- demographic survey,
- Fatigue Assessment Scale, and
- Simulation Sickness Questionnaire.

The sample of 10 participants with equal gender distribution will also take into account ethnic and age diversity. The participants will be active but non-professional drivers, with shift workers excluded.

The simulator used for testing is a high-fidelity car simulator positioned on a moving platform. Several driving parameters will be recorded during the drive, including vehicle speed, position in lane, steering wheel angle, and use of acceleration/brake pedals. Two driving scenarios will be used – a daytime and a nighttime one. Both scenarios have moderate traffic and cover different surroundings, like highway, urban and rural areas.

Karolinska Sleepiness Scale is applied in the form of a keypad with numbers 1 to 9 placed horizontally. The answer to KSS is requested every 5 minutes, which is compliant with the European Commission General Safety Regulations (2020). Every answer and its precise timestamp are registered in a log.

For the purpose of eye tracking, the Smart Eye Pro system of 3 NIR cameras is used to monitor head pose, gaze directions, blinks, and eyelid and eye movements with high accuracy. Gaze directions are represented by two separate gaze vectors, one for each of the eyes. The system allows detecting blinks and their full characteristics, including eyelid closing, opening amplitude and speed, blink duration, saccades, and fixations. Additionally, an RGB camera is used to record the course of the drives, serving as an additional source of context information.

Basic demographic data are acquired from every participant. Fatigue Assessment Scale is used to control the fatigue/arousal level of each participant before they start the drive. The occurrence of simulation sickness is monitored with the use of the Simulation Sickness Questionnaire (Kennedy et al., 1993) to ensure the wellbeing of the participants and to limit the influence of simulator sickness on the results.

2.2.1.3 *Key Performance Indicators*

Performance measurement based on Key Performance Indicators is a common way to evaluate program functioning. Two types of indicators can be identified – quantitative and qualitative.

2.2.1.3.1 Quantitative indicators

Quantitative indicators are presented with a specific objective numeric value, and they are not subjected to distortion, personal feelings, prejudices, or interpretations. That is the reason they are so commonly used in the evaluation process.

For the proposed metrics, the confusion matrix with derivatives will be calculated. Confusion matrix is a performance measurement method for classification problems, where a specific table layout is used to visualize the results of an algorithm. As shown below, the confusion matrix is a 2x2 contingency table with four different combinations of predicted (DSM application) and actual (ground truth) values.

		DSM application	
		Positive	Negative
Ground truth	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Derivatives calculated from the error matrix are helpful to describe the performance of a measure, and they should be used altogether to gain valuable insight into the classifier's efficacy. Based on the confusion matrix following metrics can be easily determined:

- **Recall** (sometimes referred to as sensitivity or true positive rate, TPR) informs how many correct predictions there were among all the positive classes. It can be viewed as the probability that a relevant positive state will be recognized by a DSM algorithm. High values are desired.

$$\boxed{Recall = \frac{TP}{TP + FN}} \tag*{1)}$$

- **Precision** (positive predicted value) is the fraction of true positives (items correctly labelled as belonging to the positive class) among all elements labelled as belonging to the positive class. High values are desired.

$$\boxed{Precision = \frac{TP}{TP + FP}} \tag*{2)}$$

- **Specificity** (true negative rate, TNR) is the fraction of correctly assigned negative classes among all the negative cases. The higher value, the better.

$$\boxed{Specificity = \frac{TN}{TN + FP}} \tag*{3)}$$

- **Accuracy** informs how many correct predictions there was (either true positive or true negative) among all the classes. High values are desirable.

$$\boxed{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}} \tag*{4)}$$

The output values from the DSM appliance can be measured against the ground truth data from the Smart Eye Pro system acquired during the simulator study. Selection of performance indicators can be done based on alike variables from both systems. Targets for proposed indicators are yet to be discussed.

Based on outcomes that are possible to compare, verification of the following metrics is proposed:

1. ***Eyes Closed****:* For *Eyes Closed* generated by DSM application and *RightEyelidOpening* and *LeftEyelidOpening* data from ground truth system, the comparison of outputs will be performed. Recalculation of ground truth signals need to be performed in order to obtain the result for the closure of both eyes.
2. ***Gaze Direction:*** For gaze direction data, compliance verification will be performed with the *IsLookingLeft* and *IsLookingRight* output generated by DSM application and *GazeHeading* from ground truth system output. *GazeHeading* is a left/right angle of the gaze direction given in radians. The threshold for rotation angle used for validation purposes needs to be established.
3. ***PERCLOS:*** For PERCLOS assessment, recalculation of *RightEyelidOpening* and *LeftEyelidOpening* signal from ground truth system needs to be performed. The output of recalculation will be used as a reference while comparing the values from the drowsiness detection algorithm.

Please note that DSM and ground truth data need to be synchronized for the effective evaluation of both program performances.

2.2.1.3.2 Qualitative Indicators

Qualitative indicators represent non-numeric insights, such as personal feelings or experiences. It is also possible (and advisable) to conduct a qualitative evaluation of the program. This can be done by examining the opinions of study participants about how the program works. It should be noted that DSM application returning an undesirable number of false positive alerts will negatively impact user experience and trust in the system (which in real-life conditions may lead to turning it off), as well as potentially causing distraction instead of preventing it.
Qualitative data will be acquired by traffic psychologist in the form of a semi-structured interview with the participant at the end of the study.

Proper analysis of the data collected during the simulator study will allow to effectively assess and improve the performance of algorithms detecting drowsiness and/or distraction.

2.2.2 Cyber Security Use-Case

CPSoSAware has developed the components to detect potential cyber-attacks in both the sensors' and the communication layers Section 2.2.2 discusses the algorithmic solution employed to detect and mitigate the attack, as well as the scenarios, the dataset and the evaluation metrics employed to assess the potential and the limitations of the cyber-attack mitigation engine. CPSoSaware cyber-attack detection and mitigation engine considers cyberattacks on the sensor layer, which act by restoring the sensor's signal quality, thus minimizing the disturbance od the scene understanding modalities.

2.2.2.1 Cyber-Attacks Introduction

During recent years we have witnessed a notable leap forward in terms of car manufacturing. Nowadays cars have become really easy to drive, even for the clumsiest drivers, thanks to a multitude of new features that continuously assist the driver during the whole route. One example could be the flexible steering that allows to change on demand the resistance of the steering wheel. Generally speaking, it is recommended that the wheel shows more resistance in highways to help keep the car trajectory at high speed and avoid undesired invasions of the neighbor lane, while when driving in cities the wheel should be softer to ease maneuvering in narrow streets. Many of these features also contribute to the safety of the passengers. Two examples would be the warning of involuntary change of lane and the danger it involves, or the assistant to start moving the car in steep slopes, avoiding dangerous recoil with a likely collision on any vehicle awaiting just behind. Built-in infotainment systems show the drivers useful information like fuel consumption, average speed, distance covered, or temperature inside and outside among others. On top of it, connectivity for cars has been developed, making it possible to have local Wi-Fi available to, for instance, enable infotainment systems internal applications or to follow the car by means of GPS location.

All this progress, which has been aligned with *"the transition from electromechanical to electronic and software-driven systems"*, [2] has been possible thanks to the increasing integration of diverse electronic components in the car. Sensors, actuators, processors… an internal network (a CAN bus is a recurrent example) governs all these features and for the sake of the safety and comfort of all passengers everything must work perfectly. The presence of these components and their networking capabilities make cars prone to be the target of cyber attackers, which eventually may disrupt their normal operation. This would make the car unsafe and increase the likelihood of an accident with unpredictable consequences for the passengers. That is why cyber attacks against cars must be a highly critical subject and huge efforts must be put into knowing well how they are performed and how to keep them from succeeding.

Cybersecurity in cars is a topic that gained attention in the past decade. Several examples can be provided to illustrate the potential severity of cyber attacks against cars. A group of researchers demonstrated in 2015 that from a remote location, it is possible to *"disable the brakes for a few seconds* *and operate the steering wheel before shutting the engine off completely"*[4]. Cyber attackers can hack the infotainment system and potentially display private information. Cyber attacks could limit the computational capacity of the car internal system. *"Malicious voice commands encoded in the sound of online videos can sneakily control the vehicle as people watch these videos in the car"*[5]. Man-in-the-Middle attacks can change the content of a message exchanged between two vehicles, generating confusion and degrading joint performance. A black-hole attack is about *"blocking a message without the car being aware of the blocked message"* [2] and in consequence making communication between the sender and the recipient of messages impossible. Also, Denial of Service (DoS) attacks *"happen when adversaries consistently send high priority messages that circumvent genuine low priority messages"*. Finally, and keeping in mind that these are just few examples of the myriad of cyber attacks that can be carried out against cars, malicious traffic can be injected in the internal network of the car.

The impact of these attacks may be lighter or more severe depending on a number of aspects. The attack may simply degrade a bit the performance of the car, without the driver even noticing. Performance may also be affected in a noticeable and unpleasant way for the user. An attack may end up with the need to decommission the service and pull over until there will be reassurance to take the car back to the road. In the worst-case scenario a cyber attack may entail a car-accident with a number of victims.

2.2.2.2 Removal of Cyber-Attacks at the Camera Level

The nature of most image processing problems is often ill-posed, meaning that there may not exist a solution for these problems. In order to alleviate the ill-posedness, the regularization process must be adopted. The underlying theory behind regularization is that the ill-posed problem may be reformulated in a way such that an appropriate solution to the reformulated problem can be found. Such solution is usually sufficient to approximate the original problem in an admissible level. Furthermore, the reformulation process can be improved by imposing some constraints such as smoothness (regularity) on the given problem. In other words, the conditioning of the given problem can be improved by imposing smoothness. Afterwards, a desirable numerical solution can be obtained by minimising the regularised energy functional using the calculus of variation introduced in the last section. In the context of image analysis, a function normally has a data fidelity term and a regularisation/smoothness term. Therefore, one may represent general image processing problems with the following energy function.

$$E(u) = \int_{\Omega} D_{f(x)}\big(u(x)\big)dx + a \int_{\Omega} R\big(u(x)\big)dx,$$

where $f(x) \colon (\Omega \subset R^n) \to R^d (n = 2 \; and \; d = 1)$ is the input image function. $\int_{\Omega} D_{f(x)} u(x) dx$ denotes the data fidelity term associated with f. It has different forms depending on applications. For example, in image denoising it is formulated according to the noise type. $D_{f(x)}\,(u(x)) \; = \; (u(x) - f(x))^2$ if $f(x)$ is

[4] https://www.uscybersecurity.net/automotive-industry/

[5] https://www.sciencedirect.com/science/article/pii/S0001457520316572

corrupted by Gaussian white noise, or $D_{f(x)}\,(u(x)) \;=\; |u(x) \;-\; f(x)|$ if $f(x)$ is corrupted by impulsive noise. $\int_{\Omega}\;R\,(u(x))dx$ is the regularisation term, and α is a positive regularisation parameter. Technically, the robust behaviour of variational methods is in fact attributed to the smoothness term $\int_{\Omega}\;R\,(u(x))dx$ that can consider information of neighboring data points. It is worth mentioning that this feature is particularly useful especially when the image contains missing information and noise. In this context, the regularisation parameter plays a role in controlling the level of smoothness of the solution.

A PDE is an equation that involves functions as variables and their partial derivatives. PDEs arise in many areas of physical science, since they are very effective in modelling a wide variety of phenomena such as sound, heat, electrostatics, electrodynamics, fluid dynamics, elasticity, or quantum mechanics. For the first type of PDEs, we are interested in the following second order linear equation

$$u(x) \;-\; \theta\Delta u(x) \;=\; g(x) \tag{5}$$

Here, $g(x)\colon (\Omega \subset R^{n}) \rightarrow R^{\,d}\,, x \in \Omega$ ($n = 2$ and $d = 1$), which is associated with the input image $f(x)$, u is the function to be calculated. Δ is the Laplace operator or Laplacian which is also denoted as $\Delta = \nabla \cdot (\nabla u) = div(\nabla u)$, where ∇ is the gradient operator mapping scalar functions to vector functions and $\nabla\,\cdot$ is the divergence operator (also symbolised "div") mapping vector functions to scalar functions. The divergence of a vector function, i.e., $p \;=\; (p_1, \ldots, p_n) \in\; R_n$ is given by

$$div(p) \;=\; \partial_{x_1}p_1 \;+\cdots+\partial_{x_n}\,p_n \tag{6}$$

With the divergence and gradient at hand, it is easy to check that

$$\Delta u \;=\; \partial_{x_1}\,\partial_{x_1}\,u \;+\cdots+\partial_{x_n}\,\partial_{x_n}\,u \tag{7}$$

In (5), u and g are real valued functions on an image grid, and g is given and u is sought. θ is a positive coefficient resulted from applying a variable splitting algorithm for an image processing problem. Equations (5) is said to be second order linear because the Laplacian in (7) is a second order linear differential operator and after (5) is fully discretized on an image grid the resulting equations are a linear system. Boundary condition also needs to be considered for (5). To benefit from the discrete fast Fourier transform (FFT) solver, which leads to a closed-form solution of u, we apply the periodical boundary with the FFT. A fourth order linear PDE is also studied, which has the form of

$$u\,(x) \;+\; \theta div^2\,(\,\nabla^2 u\,(x)) = \;g\,(x) \tag{8}$$

where θ and g have the same meaning as those in (5). ∇^2 is the Hessian operator and div^2 represents the second order divergence operator. If it is applied to a matrix valued function

$$q = (q_{11} \ \cdots \ q_{1n} \ \vdots \ddots \vdots \ q_{n1} \ \cdots \ q_{nn} \ ) \in R^{nxn}$$

we have

$$div^2 \left(q\right) \ = \ \partial_{x_1} \partial_{x_1} q_{11} \ + \cdots \ + \partial_{x_1} \partial_{x_n} q_{1n} \ + \cdots \ + \partial_{x_n} \partial_{x_1} q_{n_1} \ + \cdots \ + \partial_{x_n} \partial_{x_n} q_{nn} \qquad (9)$$

Hence, one can easily check that

$$div^2 (\nabla^2 u) = \partial_{x_1} \partial_{x_1} \partial_{x_1} \partial_{x_1} u + \cdots + \partial_{x_1} \partial_{x_n} \partial_{x_1} \partial_{x_n} u \ + \cdots \ + \partial_{x_n} \partial_{x_1} \partial_{x_n} \partial_{x_1} \ u \ + \cdots \\ + \partial_{x_n} \partial_{x_n} \ \partial_{x_n} \partial_{x_n} \ u \qquad (10)$$

(8) is a fourth order linear equation because the differential operators in (10) are the fourth order linear operator. After (8) is discretized on an image grid, the resulting equations are a linear system. Since the image is defined on a regular grid, the FFT solver can be employed very efficiently to solve (8). Another fourth order linear PDE we are concerned about has the form of

$$u \ (x) \ + \ \theta \Delta \ (\Delta u \ (x)) \ = \ g \ (x) \qquad (11)$$

where θ and g have the same meaning as those in (5). If we use the definition (7), we have the following equation for the fourth order differential operator

$$\Delta (\Delta u) = \partial_{x_1} \partial_{x_1} \partial_{x_1} \partial_{x_1} u + \cdots + \partial_{x_1} \partial_{x_1} \partial_{x_n} \partial_{x_n} u + \cdots + \partial_{x_n} \partial_{x_n} \partial_{x_1} \partial_{x_1} u + \cdots \ + \partial_{x_n} \partial_{x_n} \partial_{x_n} \partial_{x_n} u \qquad (12)$$

If we compare the mixed fourth order partial derivatives in (12) against those in (10), we can notice that the order of the first order partial derivatives is slightly different. However, in accordance with the Clairaut’s theorem, the mixed derivatives are equal if in each derivative we differentiate with respect to each variable the same number of times. In this sense, (8) is equivalent to (11) in a continuous setting.

2.2.2.2.1 Total Variation

Let us first recall the definition of this model. Consider a function $f \ : \ R^2 \ \rightarrow \ R$. We are interested in finding some u in BV, the space of functions with bounded variations, which solves the following minimization problem:

$$\frac{inf}{u \in BV(R^2)} \left\{ \int_{R^2} |Du| + \lambda \int_{R^2} |f - u| dx \right\} \qquad (13)$$

This type of energy was introduced by S. Alliney in [4] [5] in a discrete setting for one-dimensional signals. M. Nikolova showed in [6] that such an approach is efficient to restore images corrupted by impulse noise. Chan and Esedoglu [7]then gave a first theoretical analysis of the model in [7], exhibiting some very specific properties. For instance, in the case when f is the characteristic function of a disc, the solution of (13) is either f itself or 0, depending on the value of λ. Thanks to the notion of Cheeger sets, we will generalize this type of results in the present paper. Notice that this behavior is specific to the TVL1 model. For instance, in the celebrated TVL2 model (or Rudin-Osher-Fatemi, ROF, model) [8], one solves the problem:

$$\underset{u\in BV(R^2)}{inf}\left\{\int_{R^2}|Du| + \lambda\int_{R^2}|f-u|^2dx\right\} \qquad (14)$$

It is well-known that if f is the characteristic function of a disc, then the solution u of (14) is also the characteristic function of the same disc, but with a smaller contrast, whatever the value of λ in $(0,+\infty)$. This particular behavior of the solutions of the TVL1 model (13) was the motivation of the work in [7], that has been the inspiration of several subsequent works. Allard [9] [10] noticed while writing this paper: as in the present paper, he also uses geometrical measure theory results to study the TVL1 model. In particular, regularity results similar are derived in a more general setting. The specificity of our analysis, apart from aiming explicitly at image processing applications, is that it is based on recent results by V. Caselles and his collaborators [11][12][13]. In particular, by focusing on the case of dimension 2, we give a characterization of the solution of TVL1 in the convex case through classical operators from mathematical morphology. Moreover, it motivates an efficient geometrical scheme for minimizing the TV-L1 model.

In this section, we first present a few basic results about the TV-L1 functional, then we state the equivalence between TV-L1 and a geometric problem. For $f \in L^1$, we want to solve:

$$\underset{u\in L^1}{inf}\mathcal{E}(u) := |u|_{TV} + \lambda\int_{R^2}|f(x)-u(x)|dx \qquad (15)$$

The following result is well-known: it is a standard application of the direct method of calculus of variations ([7][14]). There exists at least one solution to the Problem (14). Let us define:

$$T : L^1 \to P(L^1) \qquad (16)$$

the operator which maps f to the set of solutions $Tf = \{u \in L^1, u\ is\ solution\ of\ (15)\}$.

The set T_f is a convex closed set in L^1. Let us notice that, since the functional is not strictly convex, many solutions may exist. This is the reason why we need to define T as a set-valued map. It was noticed in [7][8] that, using the coarea formula, the energy $E(u)$ could be reformulated as a sum of energies over the level sets of u:

$$E(u) = \int_{-\infty}^{+\infty} Per\{x,u(x)>\mu\} + \lambda|\{x,u(x)>\mu\}\Delta|\{x,f(x)>\mu\}|d\mu \qquad (17)$$

In view of this formula, one may consider the family of geometric problems, given a function $f \in L^1$:

$$\underset{U\subset R^2}{inf} PerU + \lambda\left|U\Delta F_\mu\right| \ (P_\mu) \qquad (18)$$

where $\mu \in R\ and\ F_\mu = \{x, f(x) > \mu\}$. Existence of solutions for each of these problems is standard since it can be proved by the direct method of the calculus of variations. Here again, we can define a multivalued map T that maps F_μ to the set of solution sets T F_μ for Problem (P_μ). Actually, [7], is shows that if the data is the characteristic function of a set ($f = x_F$), then every level set of a solution u of (15) is a solution of the geometric problem associated to F. In [14], it was pointed out that stacking solutions of the geometric problem led to a solution of (15). The following theorem states the equivalence between the family of geometric problems on level sets and the functional TVL1 problem. This result is implicitly assumed to establish several properties in[14][15], such as the contrast invariance of the TVL1

Let us notice that the mapping $t \mapsto F_t$ is non-increasing for the inclusion, therefore it is continuous everywhere but on a countable set of points (we shall denote this set I). Let us consider a countable dense set $D \subset R$ (for instance $D = Q$). First step: We are going to construct by induction a family $(U_q)_{q \in D.}$ To this end, let $(q_n)_{n \in N}$ be an enumeration of D. We choose an optimal set $U_{q0}$for problem (P_{q0}). Let us assume that we have chosen $U_{qj} (j = 0 \dots n)$ satisfying the nesting property and let us consider q_{n+1}. Three cases are possible. If there exists $m, p \leq n$ such that $q_m < q_{n+1} < q_p$ (we can assume that q_m (resp. q_p) is the greatest (resp. smallest) such element), then let us consider $U \subset R^2$ solution of $(P\ _{q_{n+1}})$, and set:

$$U_{q_{n+1}} = (U \cap U_{q_m}) \cup U_{q_p} \tag{19}$$

By induction hypothesis $U_{q_p} \subset U_{q_n} + 1 \subset U_{q_m}$, and therefore the family $(U_{qj})j = 0 \dots n + 1)$ satisfies the nesting property. $U_{q_{n+1}}$ is solution of $(P_{q_{n+1}})$. The other two cases $(\forall j \leq n,\ q_j < q_{n+1}\ or\ \forall j \leq\ n, q_j > q_{n+1})$ can be dealt with similarly (take $U\ \cap\ U_{q_m}$ or $U \cup U_{q_p}$). Therefore, we have built a countable family of optimal sets satisfying the nesting property.

Second step:

Now let us define the family $(U_t^*)_{t \in R}$ by:

$$U_t^*\ = \bigcap\nolimits_{q<t, q \in D} U_q. \tag{20}$$

This family has the nesting property; let us show that it is solution of problem (P_t) for almost every $t \in R$. Let $t \in R \backslash I$, and (t_n) an increasing sequence in Dconverging to t.

For all $V \subset R^2, Per\ V\ +\ \lambda |V\ \Delta F_{t_n}| \geq Per\ U_{t_n} + \lambda |U_{t_n} \Delta F_{t_n}|$. We take the lower limit as $n \to +\infty$:

$$Per\ V + \lambda|V\ \Delta\ F_t| \ \geq\ Per\ U_t^* \ +\ \lambda|U_t^*\Delta F_t| \qquad (21)$$

where we used the continuity of $t \ \longmapsto |F_t|$, and the lower semi-continuity of the perimeter. Eventually, the elements of the family $(U_t^*)_{t\in R}$ are solutions of (P_t), except maybe for $t \ \in I$. Thus, we can then build a function u^* by the formula:

$$u(x) \ =\ sup\{t \in R, x \in U_t^*\}. \qquad (22)$$

Let then v be a solution of problem (19) and let us call Vt its level sets. By the

correa formula, it is clear that u is a solution too, thus:

$$\int_{-\infty}^{+\infty} PerV_t + \lambda|V_t\Delta F_t|dt \ =\ \int_{-\infty}^{+\infty} PerU_t^* + \lambda|U_t^*\Delta F_t|dt \qquad (23)$$

We deduce that there is actually equality for almost every t.

2.2.2.2.2 Anisotropic Diffusion

Anisotropy in diffusion means that the smoothing induced by the PDE can be favoured in some directions and prevented in others. This is specified by local eigenvectors and eigenvalues of the diffusion tensor field. Diffusion coefficients are thus location and direction dependent, generalizing the approach of Perona and Malik [16] which is only location dependent. Importantly, efficient schemes for anisotropic diffusion have been recently made possible by the breakthrough in [16]. Non-Linearity in diffusion means that diffusion tensors are automatically generated from the processed image. A possible application of NLAD is to enhance a fingerprint image by smoothing tangentially to the lines. Evidence is also plentiful for NLAD relevance in many other image processing applications, but its use has been limited by technical aspects so far. We intend to alleviate such limitations with the present contribution.

Notations: Let $d \ \in \{\,2,3\}$ denote the image dimension, let $\Omega \subset R^d$ be the image domain, and let V be the pixel space (e.g. $[0,1]$ for grayscale, $[0,1]^3$ for color, Rd for vectors). Throughout the paper, we informally consider an idealized cartoon image model, involving a set $\Gamma \subset \Omega$ of image contours of dimension $d-1$. The processed image $u\colon \Omega \longrightarrow V$ is smooth on $\Omega\ /\ \Gamma$, but has discontinuities across Γ, and is overall corrupted by e.g. by additive white noise. A key feature of NLAD is its ability to detect the set Γ and smoothen tangentially to it. Finally, let S_d^+ denote the collection of symmetric positive definite $d \times d$ matrices, and let I_d be the identity matrix. To each $D \ \in \ S_d^+$ we associate the norm $\|\varepsilon\|_d := \sqrt{\langle e, D_e\rangle}, e\ \epsilon\ R^d$ where $\langle .,.\rangle$ denotes the standard scalar product on R^d. Linear Anisotropic Diffusion requires two main inputs: an image u0 serving as an initial condition, and a field of diffusion tensors. In order to reduce user input, the diffusion tensors D can be defined in terms of the filtered image u. The resulting PDE is called non-linear anisotropic diffusion

$$\partial_t u = div(D_u, \nabla u) \qquad (24)$$

complemented, again, with Neumann boundary conditions. Perona and Malik [16] suggested to use the following non-linear isotropic (i.e., proportional to the identity matrix) tensors

$$D_u(x,t) := c_u(x,t) I_d, \text{ with } c_u(x,t) = \frac{1}{\sqrt{1+\frac{\|\nabla u(x,t)\|^2}{\lambda^2}}} \qquad (25)$$

where $\lambda > 0$ is a user specified constant. Diffusion is prevented where the conductivity $c_u(x,t)$ is small, in other words where $|\nabla u(x,t)|$ is large, such as along the image contours Γ. It has been the subject of considerable academic and industrial interest revealing that, in spite of its numerous qualities, it is mathematically ill posed, unstable, often leads to unsightly "staircasing" visual artifacts, and is not adequate for oscillating patterns. For this purpose, we have implemented and describe in the following Coherence Enhancing Diffusion *(CED)* and Edge Enhancing Diffusion *(EED)*, which are based on more complex tensor constructions introduced by Weickert [19]. Our first ingredient is the Gaussian convolution kernel: given a standard deviation $\sigma > 0$.

$$K_\sigma(x) = \frac{1}{\sigma^d} K_1\left(\frac{x}{\sigma}\right), K_1(x) = \frac{1}{(2\pi)^d} exp\left(\frac{-\|x\|^2}{2}\right) \qquad (26)$$

The structure tensor $S_u : \Omega \longrightarrow S_d^+$, defined below, is a robust estimator of the gradient direction in an image u, even if this image has oscillating textures. It depends on two small positive parameters: the noise scale σ, and the feature scale p. We denote by $*$ the convolution operator, and by $v \otimes v = vv^T$ the self-outer product, which yields a semi-definite symmetric matrix.

$$S_u = K_p * (\nabla u_\sigma \otimes \nabla u_\sigma), \text{ where } u_\sigma := K_\sigma * u \qquad (27)$$

If u is an image with vector pixels, then S_u is the sum of the structure tensors associated to the components of u. Assume that u is a scalar image, fix a time t and a point $x \in \Omega$, and denote $S := S_u(x,t),\ v := \nabla u(x,t)$. Let also $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_d$ denote the eigenvalues of S, sorted by increasing magnitude, and $e_1, \dots, e_d$ the corresponding unit eigenvectors. If u is sufficiently smooth, then the largest eigenvalue approximates the gradient squared norm: $\lambda_d \approx \|v\|^2$, while the corresponding eigenvector approximates the gradient direction: $e_d \approx \pm \frac{v}{\|v\|}$. Weickert's diffusion tensors [19] $D := D_u(x,t)$, are defined in terms of this eigen-analysis of the structure tensor $S := S_u(x,t)$:

$$D = \sum_{1\leq i\leq d} \mu_i e_i \otimes e_i, \text{ where } S = \sum_{1\leq i\leq d} \lambda_i e_i \otimes e_i \tag{28}$$

Smoothing is promoted in the direction e_i if μ_i is large, and prevented if μ_i is small, for any $1 \leq i \leq d$. One should not shy away of designing more complex and application dependent variants; for instance, one may want to enhance filaments and tubular structures in 3D data. All depend on three parameters λ, m, α. The main one, $\lambda > 0$, is an edge detection threshold. The exponent m is typically 2 or 4. The small parameter α, typically $\frac{1}{100}$, determines the condition number of the diffusion tensors.

- Edge Enhancing Diffusion (EED) aims to avoid significant diffusion across the set Γ of image contours, but to allow it anywhere else. Note that for 3D images discontinuity planes will be enhanced, rather than edges. The first diffusion tensor eigenvalue is $\mu_1 = 1$, because the eigenvector e_1 is orthogonal to the image (approximate) gradient direction ed, hence never transverse to Γ. Other eigenvalues satisfy $\mu_i \approx \alpha \ll 1$ if $\lambda_i - \lambda_1 \geq \lambda$, $\mu_i \approx 1$ otherwise. The condition $\lambda_i - \lambda_1 \geq \lambda$ indeed suggests that the eigenvector e_i points through an image contour. Note that $\mu_1 = 1$.

$$\mu_i \coloneqq 1 - (1-\alpha)exp\left(-\left(\frac{\lambda}{\lambda_i - \lambda_1}\right)^m\right) \tag{29}$$

The choice of Weickert, to set $\mu_1 = 1$, may lead to undesired effects: one always performs diffusion in at least one direction. An undesirable side effect is that the image is blurred close to the angles of its contour set Γ. We believe that such salient features should be preserved, hence we introduce a Conservative variant of EED (cEED) for which μ_1 can be small, when appropriate, so as to prevent diffusion around the angles of Γ. Precisely:

$$\mu_i \coloneqq 1 - (1-\alpha)exp\left(-\left(\frac{\lambda}{\lambda_i}\right)^m\right) \tag{30}$$

If all eigenvalues are set equal $\mu_1 = \cdots = \mu_d$, then the diffusion tensors are isotropic, in other words scalar multiples of the identity. The following isotropic variant of EED is close in spirit to the Perona-Malik model [16]: diffusion is prevented in the neighbourhood of the image contours Γ, regardless of direction.

$$\mu_i := 1 - (1 - \alpha) exp\left(-\left(\frac{\lambda}{\lambda_d}\right)^m\right) \quad (31)$$

- Coherence Enhancing Diffusion (CED) prevents diffusion except along local image structures which have a coherent direction. The diffusion tensor eigenvalues satisfy $\mu_i \approx \alpha \ll 1$, unless if $\lambda_d - \lambda_i \geq \lambda$, in which case $\mu_i \approx 1$. The condition $\lambda_d - \lambda_i \geq \lambda$ indeed suggests that e_d points through an image feature, and that e_i points tangentially to it. Precisely: (note that $\mu_d = \alpha$)

$$\mu_i := \alpha + (1 - a) exp\left(-\left(\frac{\lambda}{\lambda_d - \lambda_i}\right)^m\right) \quad (32)$$

The above formula often leads to false positives: at a position with large gradients, but without a clear preferred direction, one may very well have $\lambda_d - \lambda_i \geq \lambda$ (for instance if $\lambda_d = 100\lambda$ and $\lambda_i = 95\lambda$). A more reliable coherence detector is $\lambda_m - \lambda_i \gg \lambda + \lambda_i$, which leads to a Conservative variant of CED (cCED): (note that $\mu_d = \alpha$).

$$\mu_i := \alpha + (1 - a) exp\left(-\left(\frac{\lambda + \lambda_i}{\lambda_d - \lambda_i}\right)^m\right) \quad (33)$$

2.2.2.3 Evaluation concept of the targeted components and use case

Given the test definition and analysis that has already been specified in Deliverable D.6.4, to evaluate the security pillar of the project we aim at setting up a defence in depth strategy that tries to capture cyber-attacks in layers. If the attacker manages to penetrate one layer (the attack is not detected) then then CPSoSaware solution should provide additional detection mechanisms on a different layer of the CPSoSaware framework. Our goal is to test the system layer as well as the CPS layer of the CPSoSaware framework in the automotive pilot by adapting our solution to the pilot and providing prevention, detection, response and mitigation (when possible) in a cyber-attack. This attack can be targeting a single CPS (one specific car) or it can be systemic (targeting connected car clusters). Thus, the cyber-attack testing procedure consist of a multi-stage attack scenario that can target various layers of CAVs. (i.e Several attack entries points). The evaluation process is aiming at testing in a simulated/emulated CAV environment the

2.2.2.4 Evaluation concept of the targeted components and use case

Given the test definition and analysis that has already been specified in Deliverable D.6.4, to evaluate the security pillar of the project we aim at setting up a defence in depth strategy that tries to capture cyber-attacks in layers. If the attacker manages to penetrate one layer (the attack is not detected) then then CPSoSaware solution should provide additional detection mechanisms on a different layer of the CPSoSaware framework. Our goal is to test the system layer as well as the CPS layer of the CPSoSaware framework in the automotive pilot by adapting our solution to the pilot and providing prevention, detection, response and mitigation (when possible) in a cyber attack. This attack can be targeting a single

CPS (one specific car) or it can be systemic (targeting connected car clusters). Thus, the cyber-attack testing procedure consist of a multi-stage attack scenario that can target various layers of CAVs. (i.e Several attack entry points). The evaluation process is aiming at testing in a simulated/emulated CAV environment as well in a real car environment the following CPSoSaware components:

1 Security Accelerators for CPS Security Agents/Sensors
2 Xilinx XRT KPI Monitoring
3 CPS Layer Security Sensors/Agents
4 Security Runtime Monitoring
5 Deep Multimodal Scene Understanding

2.2.2.4.1 Testing Environment

At small scale in a controlled and simulation oriented environment the multi-stage cyber-attack lifecycle evaluation described in the above subsection consists of a remote server (preliminary a tcp server) connected through the internet to a client software residing within an ECU or OBU of a vehicle under test. Following various different OBU/ECU testing environment we can emulate such an environment in a MultiCore System-on-Chip heterogenous embedded linux based platform that includes FPGA fabric and includes the CPSoSware Hardware Security Token developed mainly in T3.5 of the project, or a Personal Computer that supports the ML/DL/SPL functionality described at the beginning of section 3, or a n AI NN supporting embedded platform that includes CPUs and GPUs (e.g Nvidia Jetson). The following figure, that has been originally presented in D6.4 has acted as the backbone of the small scale trail that was performed.

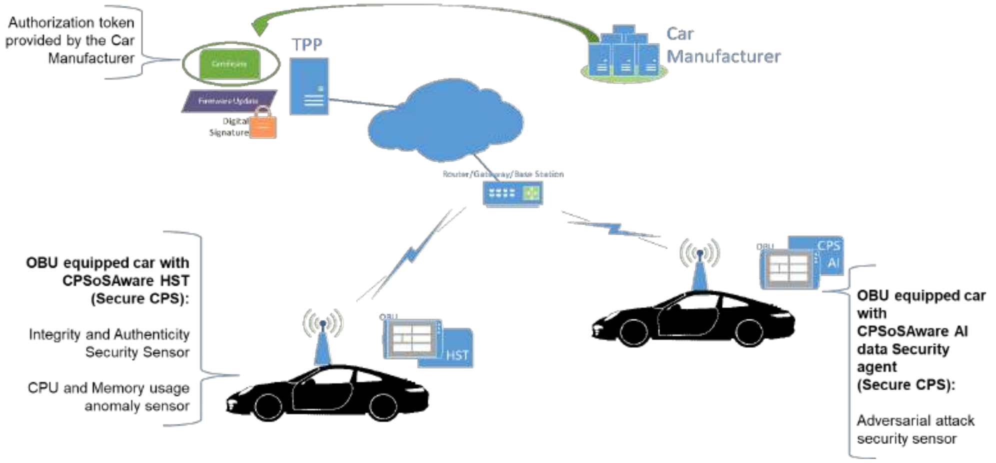
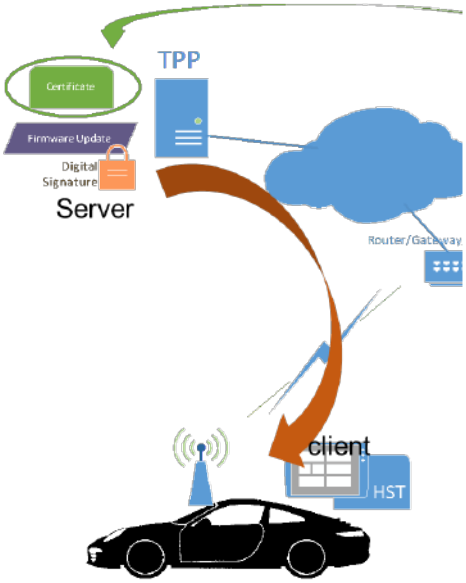

Authorization token provided by the Car Manufacturer
Certificate
Firmware Update
Digital Signature
TPP
Car Manufacturer
Router/Gateway/Base Station
OBU equipped car with CPSoSAware HST (Secure CPS):
Integrity and Authenticity Security Sensor
CPU and Memory usage anomaly sensor
OBU
HST
OBU
CPS AI
OBU equipped car with CPSoSAware AI data Security agent (Secure CPS):
Adversarial attack security sensor

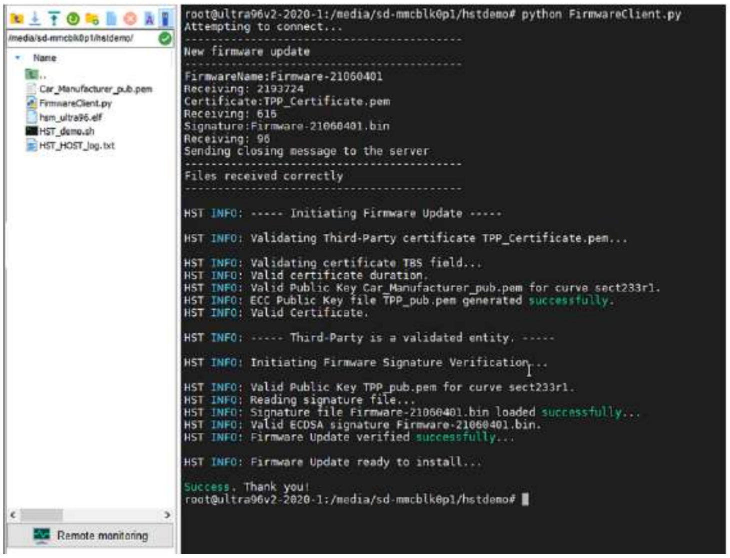
Figure 29: Small scale trial evaluation environment

The simulated cyberattack small scale trial’s use case scenario (Figure 29) originally prescribed in D6.4 has been followed. The trial scenario assumes that a car manufacturer has given permission (by providing credentials) to some third-party (OEM) developer to provide firmware/software updates on a car OEM subsystem. The credential is an appropriate certificate that includes an asymmetric cryptography public key assigned to the OEM developer (corresponding to its asymmetric cryptography private key). The certificate is digitally signed by the car manufacturer. The car manufacturer asymmetric cryptography public key is publicly known and is stored securely inside the CPS device (for example in the CPS Hardware security token) within the car. We also assume that the third-party OEM provider created an update-firmware that needs to be sent to the car and is using the public internet to transmit this firmware. The validate the authenticity of the firmware the OEM manufacturer provides along with it the certificate provided by the car manufacturer.

In the trial the scenario where the firmware/software is sent to the CPS and its security has been validated. An integrity and authenticity check has been made by the CPS security mechanisms (provided by the CPSoSaware Hardware security token (HST) or software security equivalent) and a success or failure is recorder as an event by the HST and forwarded to the SRMM system.

Upon successful authentication and message integrity check the firmware gets executed in the OBU of the car. In the following figure (Figure 30), the above-described nominal behavior of the test can be seen.





```
root@ultra96v2-2020-1:/media/sd-mmcblk0p1/hstdemo# python FirmwareClient.py
Attempting to connect...
---------------------------------------------
New firmware update
---------------------------------------------
FirmwareName:Firmware-21060401
Receiving: 2193724
Certificate:TPP_Certificate.pem
Receiving: 616
Signature:Firmware-21060401.bin
Receiving: 96
Sending closing message to the server
---------------------------------------------
Files received correctly
---------------------------------------------

HST INFO: ----- Initiating Firmware Update -----

HST INFO: Validating Third-Party certificate TPP_Certificate.pem...

HST INFO: Validating certificate TBS field...
HST INFO: Valid certificate duration.
HST INFO: Valid Public Key Car_Manufacturer_pub.pem for curve sect233r1.
HST INFO: ECC Public Key file TPP_pub.pem generated successfully.
HST INFO: Valid Certificate.

HST INFO: ----- Third-Party is a validated entity. -----

HST INFO: Initiating Firmware Signature Verification...

HST INFO: Valid Public Key TPP_pub.pem for curve sect233r1.
HST INFO: Reading signature file...
HST INFO: Signature file Firmware-21060401.bin loaded successfully...
HST INFO: Valid ECDSA signature Firmware-21060401.bin.
HST INFO: Firmware Update verified successfully...

HST INFO: Firmware Update ready to install...

Success. Thank you!
root@ultra96v2-2020-1:/media/sd-mmcblk0p1/hstdemo#
```

Figure 30: Expected/nominal behaviour of the small scale trial on cybersecurity validation

2.2.2.4.2 Abnormal/Malicious Behavior test setup

As described in D6.4, to validate the detection capabilities of the security pillar, we assume two types of attackers in the test small scale trial scenario. The first attacker is a man in the middle attacker that maliciously alters the firmware, the certificate or both during transmission over the communication channel. The second type of attacker is the Man at the end (or man at the source) attacker that alters maliciously the firmware, the certificate or both at the source i.e when these data are been generated in the OEM provider.

We assumed the when the malware gets activated in can be detected by some anomaly detection mechanism because it leaves traces of its behavior in the OS execution flow that can be monitored by appropriate detectors or characteristic patterns on the sensory system of the car. The malware in the tests attempt a Denial of Service to the car OBU/ECU, attempt information leakage and/or eventually perform adversarial attacks on the car’s sensors (as it is further described in D6.4).

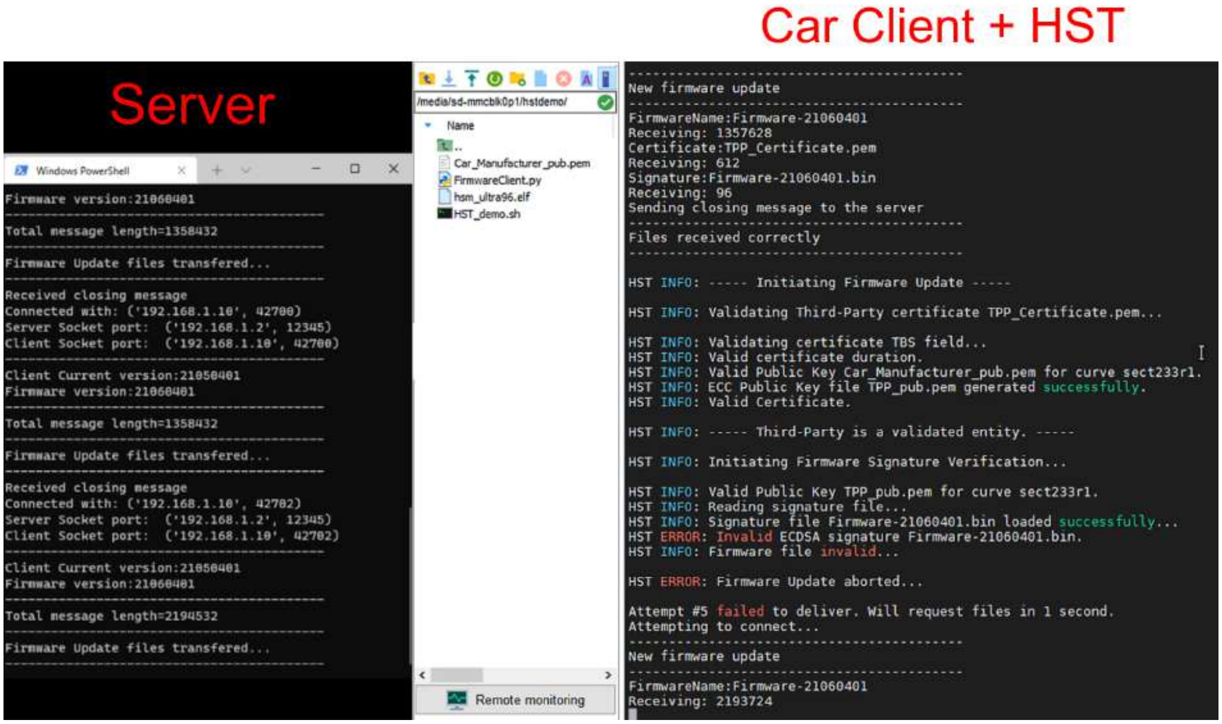
2.2.2.5 Testing procedure, data acquisition protocols and preliminary results

The small scale trial evaluation of the above scenario validates the cybersecurity detection capabilities at a small scale (on simulated environment) of the CPSoSaware architecture components on data integrity/authenticity attacks, on network security attacks and on availability (denial of service) attacks targeting a CPS device as those are manifested by the HST CPS component and its connection with the SRMM. Eventually the detection is performed in the SRMM system component where appropriate alerts are generated. We adopted the following test procedures:

- ***Corrupted firmware due to network transmission loss:*** In this test, the firmware is not maliciously altered however it is been sent to the car using an unreliable communication mean which potentially can fail to correctly transmit the firmware correctly. The goal of the test is to be able to avoid false positives in the SRMM tool due to repeatedly reported integrity failure events stemming from the HST at the CPS level (within the car OBU). We define a testing window of 50 transmissions where the 49 first transmissions are sending an erroneous firmware to the car OBU (equipped with the CPSoSware HST) and the last transmission is sending a correct firmware.
- **Malicious firware due to man in the middle attack:** In this test, the firmaware is been maliciously altered during transmission since we assume a man in the middle attacker residing on some intermediate network device eg. a router. The malicious modification may include apart from the firmware also the certificate associated with the firmware. Thus we define 2 subcases: a modified firmware with a legit certificate, a modified firmware with a modified certificate, a legit firmware with a modified certificate. We contact experiments for all this cases and obsed if the malicious behavior is detected by the SRMM through appropriate events sent from the HST at the CPS layer.
- **Malicious firware due to man at the source attack**: This test is similar to the previous one with the exception that we assume the attacker to be a malicious insider in the OEM provider. In this case, the malicious insider attacker changes firmware into malicious one at the source i.e the OEM provider and generates a valid digital signature using the OEM keys for this firmware while also providing the valid Car manufacturer certificate. The test aims to validate how the malicious firmware will be detected given that the integrity and authentication security sensor/agent in the HST will provide a false negative result (it won’t detect the malicious behaviour). The malicious firmware will, thus, get executed. However, the HST also has a secondary security sensor/agent that monitors abnormalities in the CPU/ memory usage. Triggering the malicious firmware will produce abnormal behaviour in the CPS that will be detected as an event and be forwarded to the SRMM.

2.2.2.5.1 Preliminary results

Following the above 3 test procedures, several experiments have been made using the setup described in the previous subsections. In the following figures we provide the test detection results as those are visually produced from the logging mechanism of the HST.

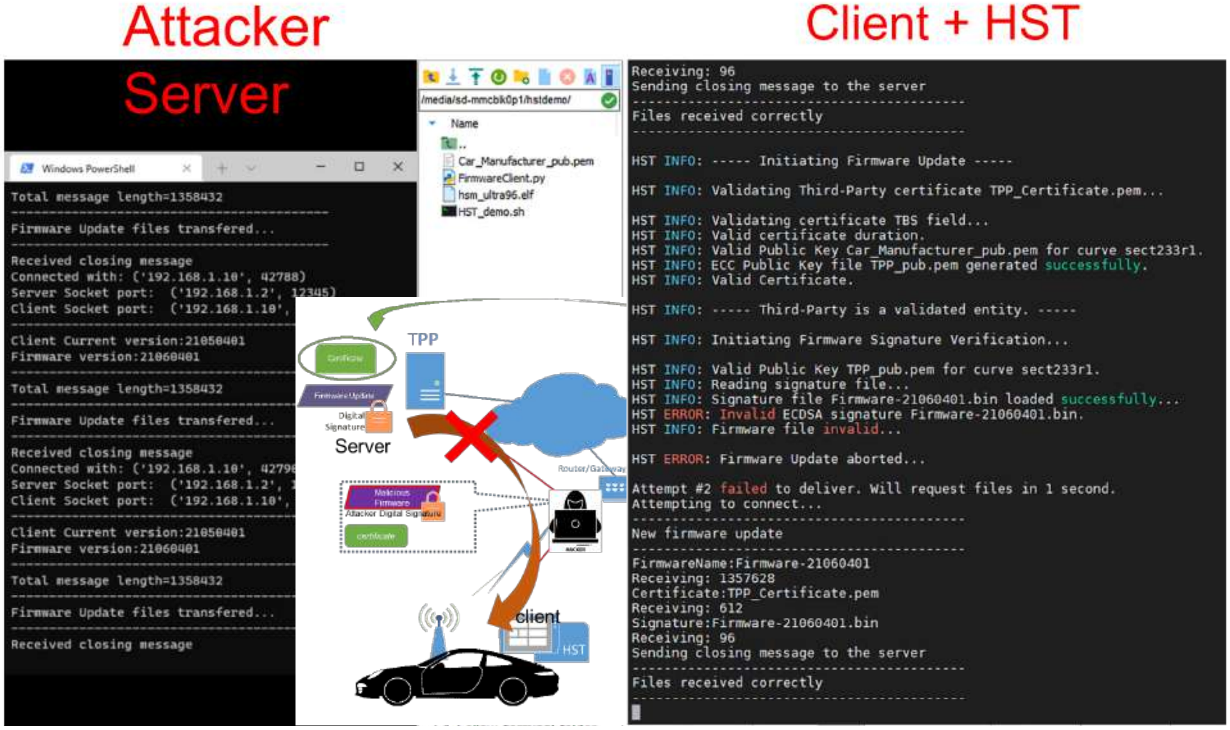


```
Firmware version:21060401
-------------------------------------------
Total message length=1358432
-------------------------------------------
Firmware Update files transfered...
-------------------------------------------
Received closing message
Connected with: ('192.168.1.10', 42700)
Server Socket port:  ('192.168.1.2', 12345)
Client Socket port:  ('192.168.1.10', 42700)
-------------------------------------------
Client Current version:21050401
Firmware version:21060401
-------------------------------------------
Total message length=1358432
-------------------------------------------
Firmware Update files transfered...
-------------------------------------------
Received closing message
Connected with: ('192.168.1.10', 42702)
Server Socket port:  ('192.168.1.2', 12345)
Client Socket port:  ('192.168.1.10', 42702)
-------------------------------------------
Client Current version:21050401
Firmware version:21060401
-------------------------------------------
Total message length=2194532
-------------------------------------------
Firmware Update files transfered...
-------------------------------------------
```

```
-------------------------------------------
New firmware update
-------------------------------------------
FirmwareName:Firmware-21060401
Receiving: 1357628
Certificate:TPP_Certificate.pem
Receiving: 612
Signature:Firmware-21060401.bin
Receiving: 96
Sending closing message to the server
-------------------------------------------
Files received correctly
-------------------------------------------

HST INFO: ----- Initiating Firmware Update -----

HST INFO: Validating Third-Party certificate TPP_Certificate.pem...

HST INFO: Validating certificate TBS field...
HST INFO: Valid certificate duration.
HST INFO: Valid Public Key Car_Manufacturer_pub.pem for curve sect233r1.
HST INFO: ECC Public Key file TPP_pub.pem generated successfully.
HST INFO: Valid Certificate.

HST INFO: ----- Third-Party is a validated entity. -----

HST INFO: Initiating Firmware Signature Verification...

HST INFO: Valid Public Key TPP_pub.pem for curve sect233r1.
HST INFO: Reading signature file...
HST INFO: Signature file Firmware-21060401.bin loaded successfully...
HST ERROR: Invalid ECDSA signature Firmware-21060401.bin.
HST INFO: Firmware file invalid...

HST ERROR: Firmware Update aborted...

Attempt #5 failed to deliver. Will request files in 1 second.
Attempting to connect...
-------------------------------------------
New firmware update
-------------------------------------------
FirmwareName:Firmware-21060401
Receiving: 2193724
```

Figure 31: Corrupted firmware due to network transmission loss visual result

In Figure 31, it can be seen that the transmission from the server (the OEM provider) to the client (the car OBU equiped with HST) is beem repeatedly failing the integrity and authenticity check. However, within the given window (50 transmissions) a valid firmware is sent. The HST is not able to detect that this is not a malicious firmware and every time generates a false positive however, the generated alerts are accumulated in the SRMM where the window is set and the false positives are evaluated. Since the procedure gives a legit firmware within the time window determined in the SRMM, the false positives do not trigger an alert.

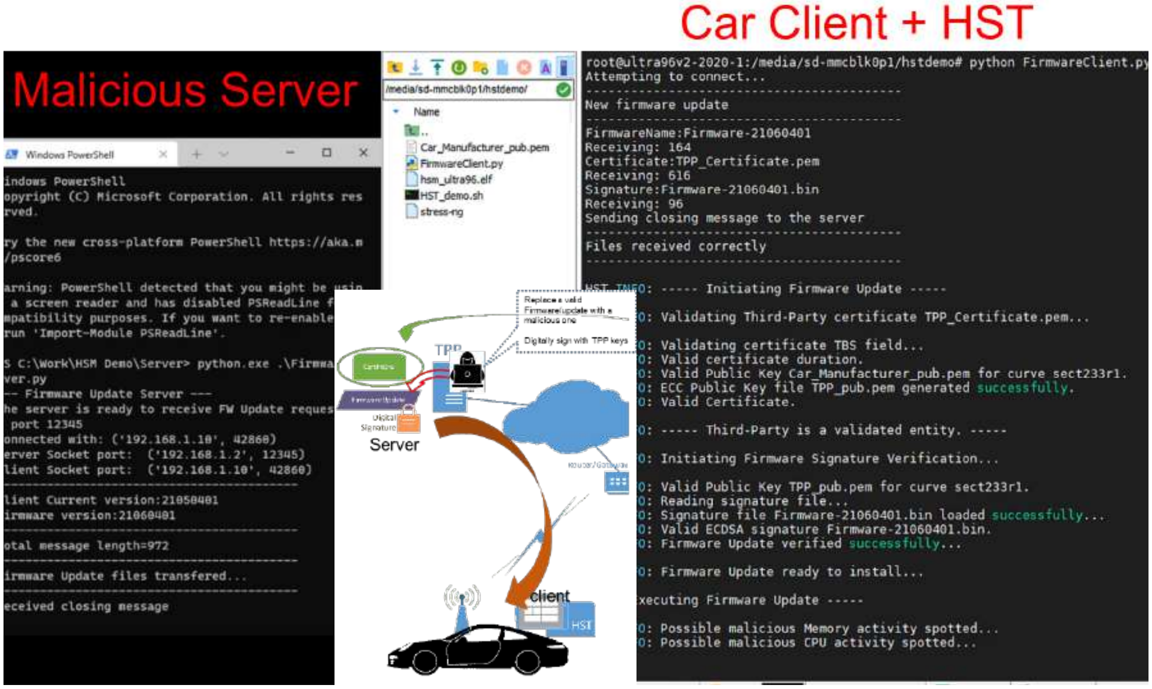


```
Total message length=1358432
-----------------------------------------
Firmware Update files transfered...
-----------------------------------------
Received closing message
Connected with: ('192.168.1.10', 42788)
Server Socket port:  ('192.168.1.2', 12345)
Client Socket port:  ('192.168.1.10',
-----------------------------------------
Client Current version:21050401
Firmware version:21060401
-----------------------------------------
Total message length=1358432
-----------------------------------------
Firmware Update files transfered...
-----------------------------------------
Received closing message
Connected with: ('192.168.1.10', 4279
Server Socket port:  ('192.168.1.2', 1
Client Socket port:  ('192.168.1.10',
-----------------------------------------
Client Current version:21050401
Firmware version:21060401
-----------------------------------------
Total message length=1358432
-----------------------------------------
Firmware Update files transfered...
-----------------------------------------
Received closing message
```

```
Receiving: 96
Sending closing message to the server
-----------------------------------------
Files received correctly
-----------------------------------------

HST INFO: ----- Initiating Firmware Update -----

HST INFO: Validating Third-Party certificate TPP_Certificate.pem...

HST INFO: Validating certificate TBS field...
HST INFO: Valid certificate duration.
HST INFO: Valid Public Key Car_Manufacturer_pub.pem for curve sect233r1.
HST INFO: ECC Public Key file TPP_pub.pem generated successfully.
HST INFO: Valid Certificate.

HST INFO: ----- Third-Party is a validated entity. -----

HST INFO: Initiating Firmware Signature Verification...

HST INFO: Valid Public Key TPP_pub.pem for curve sect233r1.
HST INFO: Reading signature file...
HST INFO: Signature file Firmware-21060401.bin loaded successfully...
HST ERROR: Invalid ECDSA signature Firmware-21060401.bin.
HST INFO: Firmware file invalid...

HST ERROR: Firmware Update aborted...

Attempt #2 failed to deliver. Will request files in 1 second.
Attempting to connect...
-----------------------------------------
New firmware update
-----------------------------------------
FirmwareName:Firmware-21060401
Receiving: 1357628
Certificate:TPP_Certificate.pem
Receiving: 612
Signature:Firmware-21060401.bin
Receiving: 96
Sending closing message to the server
-----------------------------------------
Files received correctly
-----------------------------------------
```

Figure 32: Malicious firware due to man in the middle attack visual result

In Figure 32, it can be observer that when some adversary alters the firmware or the signature or the certificate of the OEM provider in transit, then the HST on the CPS layer is able to generate an event that is directly fed to the SRMM thus generating alerts. The test has been made with all possible integrity failure combinations and in all cases the integrity and authenticity failure was discovered and was reported.



```
indows PowerShell
opyright (C) Microsoft Corporation. All rights res
rved.

ry the new cross-platform PowerShell https://aka.m
/pscore6

arning: PowerShell detected that you might be usin
 a screen reader and has disabled PSReadLine f
mpatibility purposes. If you want to re-enable
run 'Import-Module PSReadLine'.

S C:\Work\HSM Demo\Server> python.exe .\Firmwa
ver.py
-- Firmware Update Server ---
he server is ready to receive FW Update reques
 port 12345
onnected with: ('192.168.1.10', 42860)
erver Socket port:  ('192.168.1.2', 12345)
lient Socket port:  ('192.168.1.10', 42860)
---------------------------------------------
lient Current version:21050401
irmware version:21060401
---------------------------------------------
otal message length=972
---------------------------------------------
irmware Update files transfered...
---------------------------------------------
eceived closing message
```

```
root@ultra96v2-2020-1:/media/sd-mmcblk0p1/hstdemo# python FirmwareClient.py
Attempting to connect...
---------------------------------------------
New firmware update
---------------------------------------------
FirmwareName:Firmware-21060401
Receiving: 164
Certificate:TPP_Certificate.pem
Receiving: 616
Signature:Firmware-21060401.bin
Receiving: 96
Sending closing message to the server
---------------------------------------------
Files received correctly
---------------------------------------------

HST_INFO: ----- Initiating Firmware Update -----

O: Validating Third-Party certificate TPP_Certificate.pem...

O: Validating certificate TBS field...
O: Valid certificate duration.
O: Valid Public Key Car_Manufacturer_pub.pem for curve sect233r1.
O: ECC Public Key file TPP_pub.pem generated successfully.
O: Valid Certificate.

O: ----- Third-Party is a validated entity. -----

O: Initiating Firmware Signature Verification...

O: Valid Public Key TPP_pub.pem for curve sect233r1.
O: Reading signature file...
O: Signature file Firmware-21060401.bin loaded successfully...
O: Valid ECDSA signature Firmware-21060401.bin.
O: Firmware Update verified successfully...

O: Firmware Update ready to install...

xecuting Firmware Update -----

O: Possible malicious Memory activity spotted...
O: Possible malicious CPU activity spotted...
```

Figure 33: Malicious firware due to man at the source attack visual result

In Figure 33, the malicious firmware is generated at the source and the attacker manages to also forge the digital signature of the firmware and is also able to provide the correct Car manufacturer certificate. In this case no integrity and authenticity failure is identified as seen the figure. The malware is getting executed and performs some operation that deviates from the typical behavior of the CPS software. We assume that such abnormality is related to the CPU usage and Memory usage of the system. As can be seen from the visual preliminary results the CPU/memory usage anomaly detector can identify the malware, generate events and send the to the SRMM system which in turn produces alerts on this issue. The preliminary test have shown accurate detection only when the memory/CPU usage is considerably different compared to the CPS device nominal level. This is a highly viable scenario when considering one of the most populat attacks on the automotive which is denial of service attack.

The experiments that are performed as part of the preliminary tests on the cybersecurity in connected autonomous cars will follow two procedures. The first procedure is focused on the stages 1 to 3 of the multistage cyberattack lifecycle and includes testing of cybersecurity detection capabilities of the CPSoSaware architecture components on data integrity/authenticity attacks, on network security attacks and on availability (denial of service) attacks targeting a CPS device. The second testing procedure is focused on the detection of adversarial attacks (stage 4 of the multistage cyberattack lifecycle) on the camera sensors of a single vehicle using the CPSoSaware cyberattack detection and mitigation software engine provided by PASEU.

2.2.2.6 Planning of Large-scale trail test procedure

Beyond the above presented small scale trials, in this deliverable we can prescribe more complex cybersecurity large scale trials that introduce the variability and complexity of the network communication environment (not just the two end nodes presented in this deliverable), thus placing the cybersecurity as an aspect of a more sophisticated demonstration and evaluation process. The CPSoSAware project proposes a demonstrator (in the context of WP4), to show how SDN techniques and NFV technology can help with the dynamic configuration of the network, when interaction between different layers of the architecture is required. In order to validate the demonstrator, few scenarios have been designed and one of them relates to the cybersecurity field. In that scenario the OBU (on board unit) of the vehicle, thanks to the installed monitoring capabilities of sensors, cyber agent and the lightweight version of the SRMM (Security Runtime Monitoring Module, already presented in deliverable D4.3 and further discussed in upcoming D4.8) will detect a possible cyber attack incident and will report it all the way to the edge layer and possibly to the cloud layer. In this case the communication goes through all the layers defined in the architecture: the OBU on the vehicle (representing the CPS layer), the RSU (Road Side Unit), the Edge Layer and if required, the event will be analysed and correlated at the Cloud Layer, where specific alarms will be raised by the full version of the SRMM. Then a mitigation action will be chosen. In case that the event reaches the Cloud Layer, a possible course of action would be to issue a decommissioning message to that particular vehicle. Once the car receives the decommissioning message, it will display a message for the driver, to get out of the road and find a safe place to stop the car. When that is done, the OBU can send back the acknowledgement message and the Cloud System Layer can push, all the way down to the car, the binaries for a new firmware that will be automatically flashed into the required system. At the end, a request for commissioning message will be sent and once the Cloud System Layer decides that the incident is resolved, it will issue back a commissioning message that will indicate to the driver that it is now safe to resume the driving. More details about the communication flows in this scenario can be found on Deliverable 4.7. The following figure (Figure 34) depicts the scenario described in the text.

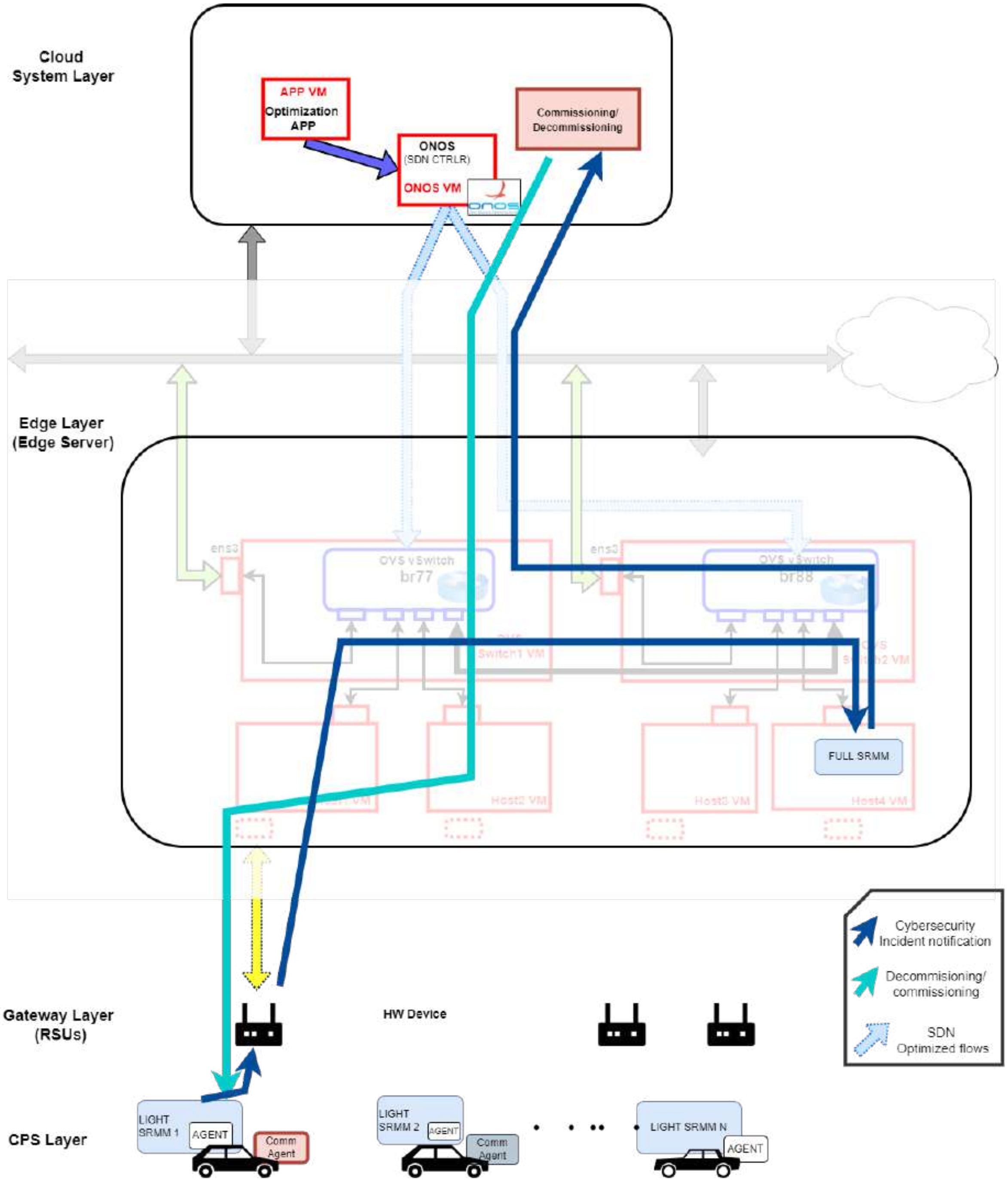


Figure 34: Overall cybersecurity compliant demonstration and validation scenario for network communication in the automotive pilot

2.2.2.7 *Dataset Acquisition Process (Real Vehicle)*

Our experiment takes into account images extracted from 100 recorded videos with 1 minute each in different parking areas, including supermarkets, office with angle, parallel and perpendicular parking scenarios. The vehicle platform used for data collection is illustrated in.

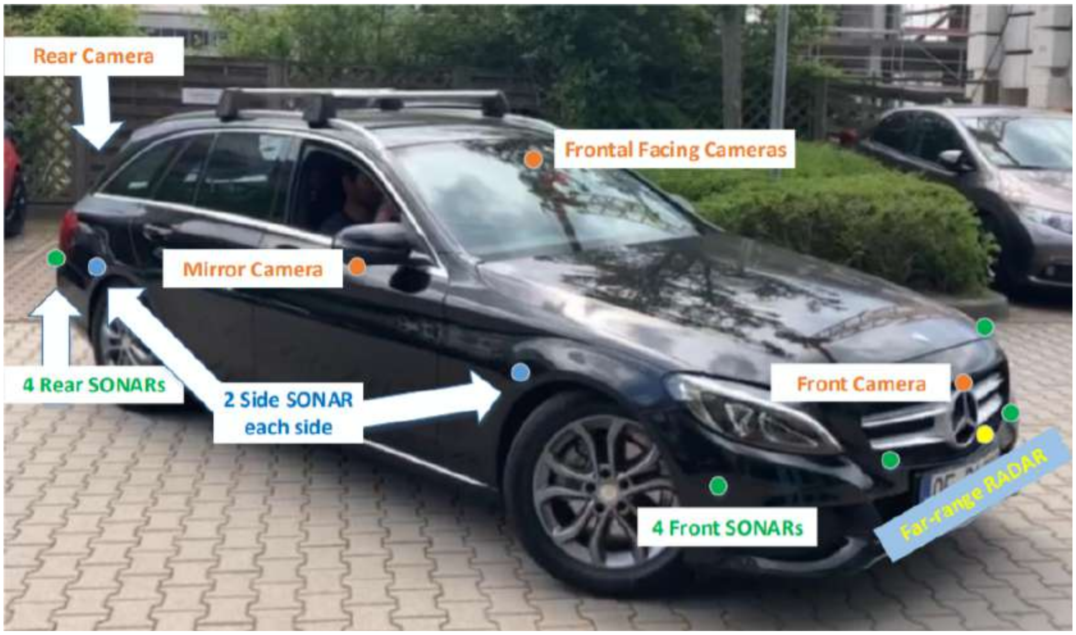


Figure 35: **Picture of our ego-vehicle with sensors mounted**

Each raw image (width 1280 and height 960) is attacked by noise, which are then used as inputs for our denoised models. The parameters for denoising models (see section **Error! Reference source not found.**) are tuned to generate a noise free image. And secondly, this denoised image is being fed to our computer vision algorithm to analyse the variation in output.

Figure 36 illustrates the architecture of the cyber-attack .and some instances of the mitigation output.

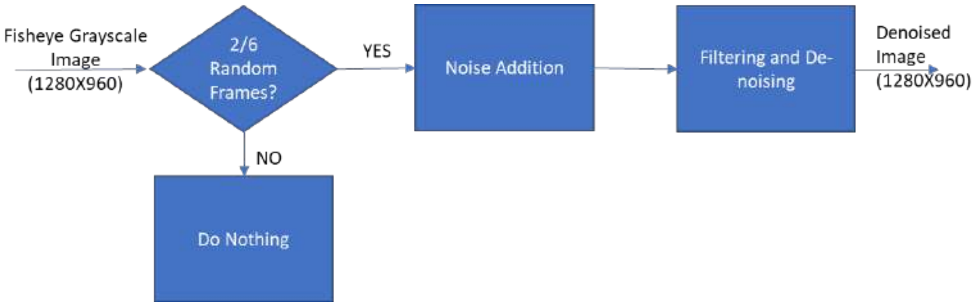


Figure 36:**First Experimental Setup**

- Overview of the recording system:

The main purpose of the system architecture is to be able to measure information and compute it to provide some ADAS functionality.

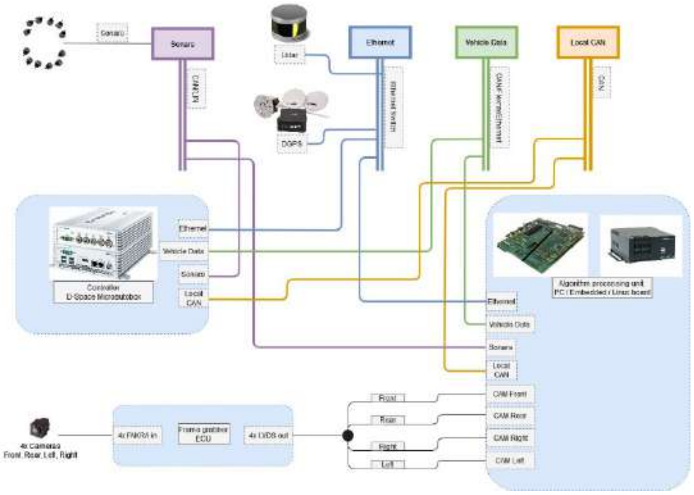


Figure 37:**ADAS System Architecture Diagram**

Mostly the system architecture is divided into sensors like sonars, cameras whose task is to collect data from the real world and from the vehicle itself and then transmits it through the communication networks which is connecting all the components together to then reach to the processing components like the CarPC and the Controller which use the transmitted data to either record the data or process it to make some decisions then if needed it can send back results and control signals through the communication networks again. In some cases it is needed to generate some ground truth data to be used to compare the system results in order it be able to compare and generate some metrics about how accurate the system is and if further fixes is needed or not, in this case some more sensors can be used like the LIDAR and the Differential GPS, which provides high accuracy information regarding the surroundings and the current position, speed, direction of the ego-vehicle which can be used later to generate ground truth data.

As described in Figure 37, system components are described as follows :

- <u>Sensors:</u>

The sensors are the components responsible for getting information from the real world and the ego vehicle itself to be sent and used by the rest of the ADAS system components.

  - <u>Cameras:</u>

Figure 38: **ADAS System Cameras**

Panasonic is using 4 Cameras(as in Figure 38) (Left, Right, Front, Rear), with resolution of 1280*960 and 30 fps. The cameras in use currently are UVC devices , more information regarding UVC protocol can be found in [20].

- **Sonars:**

Figure 39: **ADAS System Sonars**

Sonars (as in Figure 39) main task is to use sonar waves to detect objects and obstacles around the vehicle, currently Panasonic is using 12 sonars (4 Front, 4 Rear, 4 Sides) to cover the surroundings of the vehicle.

- **LIDAR (HDL-32E) [21]:**

LIDAR sensor is mounted on top of the vehicle to use laser to scan any objects or obstacles around the vehicle, currently Panasonic is using HDL-32E providing coverage of 360*32 degrees , mainly the LIDAR is used in the ground truth generation to be able to compare the detect objects verse the detected objects generated from the ADAs system, more information regarding the LIDAR can be found in [21].

- **Diffrential GPS (Applanix POSLV) [22]:**

DGPS is mounted on the vehicle to get accurate position and heading of the vehicle using the satellites connected to the 2 GPS antennas in the system with accuracy can reach to 2-5 cm, mainly it is used to generate ground truth which can be used to compare the accurate position of the vehicle verses the vehicle odometry generated by the ADAS system, more information regarding Applanix can be found in [22].

As explained in the overview the data collected by the sensors are sent across the system using the communication buses , which are discussed below.

- **Communication buses:**

Communication buses are the main connections between different components in the system, using the communication buses sonars can move data to different system points.

- **Vehicle Network Bus:**

Currently vehicle network buses supported are CAN/CAN-FD, FlexRay and Ethernet, the bus can be connected to various components and devices using various types of adaptors. The main use of the bus is to move the vehicle ego data across the whole system, the mostly used vehicle data is speed, steering, wheels pulses, wheels speeds, gear, blinkers, any optional accessories.

In order to be able to get the vehicle odometry data as much real time as possible it should be connected to the vehicle data bus without any processing unit in between which can introduce extra time delay.

- **Local Network Bus:**

Currently supported buses for Local bus are CAN/CAN-FD. The main purpose for this is to connect between different system components away from the vehicle main bus, according to the use-case it is possible to include local network bus in the recordings. The main reason to use the local network and not use vehicle network for inter system communication is to avoid the possibility that some data can affect the vehicle.

As explained in the overview the processing and the decisions making part is held in the processing components which are loaded with Panasonic software algorithms, more details can be found in the nest section.

- **Processing Components:**

Those components are the main components running different algorithms to consume the data from the system sensors.

  - **CarPC (Spectra-Power Box 12C0)[23]**:

    The main PC running the recording software as well as hosting the main algorithm software, it can be a windows machine, Linux machine, or an embedded environment. More information can be found in [23].
  - **Controller (D-Space Microautobox) [24]:**

    D-Space controller is the main component interacting directly with the vehicle actuators (acceleration control, steering control, braking control) , mainly it use the data coming from the CarPC & the sensors over the communication buses to control the vehicle according to the use case. More information can be found in [24].

Panasonic is mainly using its own recording engine which has been design to fulfil all the requirements related to the data synchronisation and data storage, a brief on the recording software and data it records:

- **Panasonic Recording Software:**

  Panasonic recording software is the main software to receive from all sensors and data nodes in the vehicle and prepare it to be recorded or even populated to any needed algorithm component, for the recording activity the recording software is storing all recorded data in an output folder containing the following items:
  - **AVI files:**

    The .avi files containing recorded video files, it is uncompressed raw recordings
  - **Events.txt file:**

    The main file containing different events in sequential order, for each event it will have its specific time stamp and whether it’s event data or a handle to the event data in another file (i.e., avi, binary….) All time stamps are relative to the first-time stamp in the file which is 0.00000 seconds.
  - **Binary files:**

    For special sensors like Velodyne , the recording software dumps the recorded data in a special binary file, and a parser to retrieve needed data from the binary file is also developed by Panasonic.
  - **Descriptor file:**

    Contains metadata regarding the recording itself , camera calibration data , weather , date , location , car type, which is used to properly store the recording.

One of the main objectives of our recording software is to guarantee high accuracy of synchronisation between different data capture by sensing components, we are supporting different mode of sync and time stamping as follows.

- <u>Time Stamps Modes:</u>

When Panasonic recording software receives a new event it attaches time stamp to it, still it should synchronise all the events time stamp together and still keep the least amount of deviation between the real time of the event and the attached time stamp. To achieve this we have 2 different modes of generating and attaching time stamps:

  - <u>Host Machine mode:</u>

    In this mode each event time stamp is generated on the host machine upon the reception of the event which makes handling time synchronization between different inputs much easier, but the accuracy of the time stamps and the standard deviation between the real time of the event and the generated time stamp depends on the software and the PC specs.

  - <u>Sensor Source mode:</u>

    In this mode each sensor and machine need to synchronise with a master clock sync provider (i.e., the CarPC or else), then each sensor shall send its data with its own time stamp, and he host PC is not generating any time stamps. This mode makes time stamps more accurate and does not depend on the PC specs, but it can be harder to maintain and handle the order of the events to keep the evets queue in order, while feed the events to the needed components (if needed).

Evaluation Protocol

According to the literature, the impact of noise is usually visually justified within the field of image analysis. In automotive, it is more critical to look at the impact resulted in influencing the performance of applications, and especially for autonomous navigation.

2.2.2.7.1 Cyber Attack Mitigation- Evaluation Protocol

Autonomous navigation is the main task of autonomous vehicle, which relied on a reliable occupancy grid map(OGM)[26][27] for obstacle avoidance and path planning. An example of occupancy is illustrated in Figure 40. It is of interest in this research to understand the impact of noise on the performance of autonomous vehicle in its autonomous navigation. Concretely, we would like to understand how much is degradation of the OGM when noise is added to the input images? And thus, will such degradation worsen the performance or decision making of the autonomous vehicle? In fact, there are several generic approaches [28][29] for evaluating OGM in robotics, but being just mainly for references. Since there are many different functions using OGM for different purpose during autonomous navigation, such as automated emergency braking, collision avoidance, autonomous parking, etc.; it is not obviously to justify the impact without concrete evaluation per specific purpose.

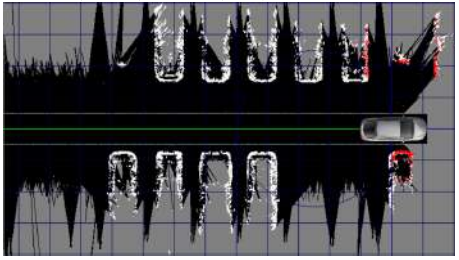


Figure 40: Example of Occupancy Grid Map, which contains obstacle cells (white), free-space cells (black), and unknown cells (grey). Red cell are reflecting the current update of cells of obstacle points.

Based on different purposes, the analysis of OGM to extract useful information is investigated. Reliable and efficient approaches are derived depending on types of functions. For instance, for obstacle avoidance in navigation, it is practically efficient and much more meaningful to compare the outcome of obstacle polygons extracted between OGMs. This is because the polygons are fundamentally used for calculating time-to-collision and planning the local path to avoid possible collision. On the other hand, for parking applications, it is much more meaningful to measure the free-space and obstacle's boundaries within surrounding environment to examine whether or not a possible parking-slot is detected and an optimal trajectory can be planned accordingly. More detailed information on the extraction of the evaluation metrics were provided in D6.4.

2.2.2.7.2 Cyber Attack Mitigation- Evaluation Protocol

The Key performance indicators for evaluating the output of the cyber security use cases have been defined in D6.4 deliverable of CPSoSAware.

2.3.1 Co-Operative Awareness Scenarios

Evaluation concept of the targeted components and use case

2.3.1.1 *Multimodal object detection solutions*

The developed 2D object detection solutions incorporate 2D object detection described in D3.1[6] Furthermore, the developed 3D object detection solutions incorporate PVRCNN and PointPillar LIDAR based detection networks[7] also described in D3.1 . A late fusion strategy takes into account both outcomes to improve the accuracy. The developed solutions were tested in both CARLA and KITTI testing environments. The following images present a qualitative evaluation visualising the outcome of the 2D and 3D object detection fusion.

Multimodal fusion approach receives as input 3D and 2D measurements. The 3D position of the detected objects is derived through the LIDAR data. 3D points are projected to the 2D plane. The points encapsulated by 2D bounding boxes are taken into account to define the detected object position in the 3D space. The calculated positions are used as range measurements for cooperative localization.

[6] Nousias, Stavros, et al. "Accelerating deep neural networks for efficient scene understanding in automotive cyber-physical systems." *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2021.

[7] Nousias, Stavros, et al. "Accelerating 3D scene analysis for autonomous driving on embedded AI computing platforms." *2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2021.

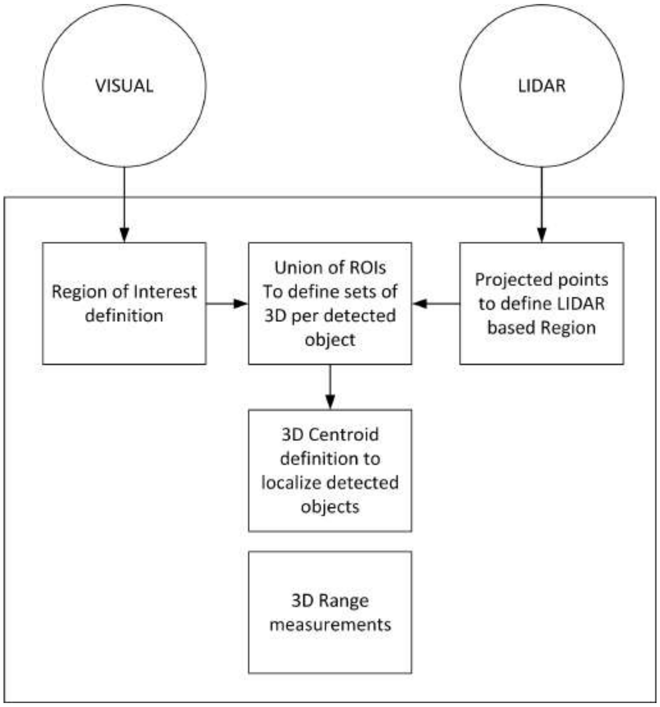

VISUAL
LIDAR
Region of Interest definition
Union of ROIs To define sets of 3D per detected object
Projected points to define LIDAR based Region
3D Centroid definition to localize detected objects
3D Range measurements

Figure 41:: Multimodal fusion approach and generation of range measurements

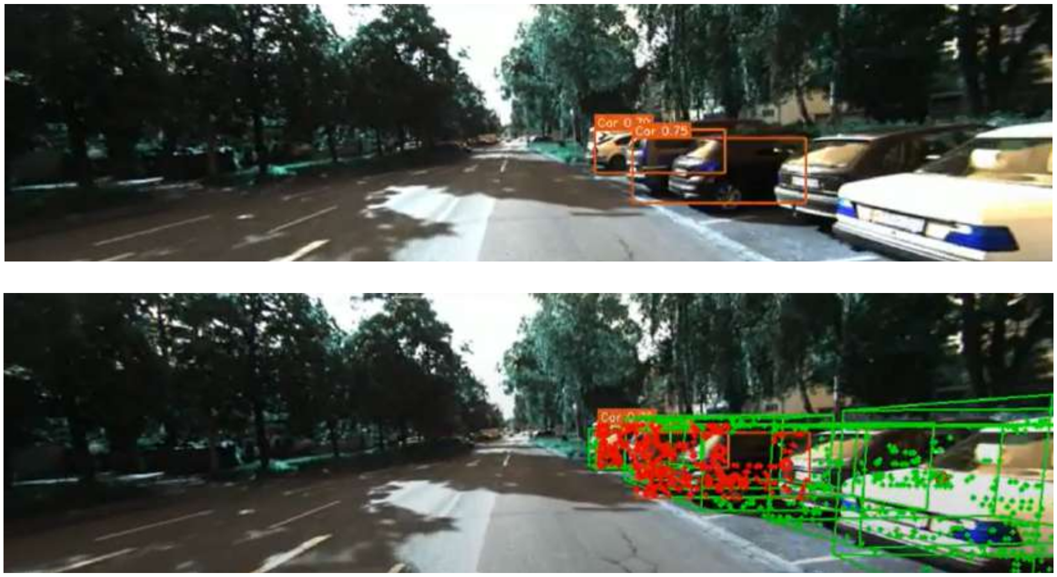


Figure 42: Fusion outcome qualitative evaluation

2.3.1.1.1 Cooperative Localization, Tracking and Awareness tasks of WP3

For the Cooperative Localization, Tracking and Awareness tasks of WP3, we have assumed that vehicles in an urban environment through V2V communication exchange their measurements so as to estimate more accurately their positions. For that purpose, a graph-based approach has been utilised, which couples together the vehicles' connectivity through the graph Laplacian operator, with the multi-modal inter-vehicular measurements. Two general methodologies have been developed: (i) information diffusion based, in which neighbouring vehicles broadcast and receive in an iterative manner the estimated location vectors, (ii) Kalman Filter based, where the state vector that needs to be tracked contains the self and neighbouring vehicles' positions. The latter approach is able to address the dynamic nature of connectivity topologies highly efficiently. In both cases, the key step of extracting the range measurements (relative distance and angle) towards other vehicles has been skipped. These approaches will be analytically presented in deliverable D3.1. Our goal is to explicitly integrate the previously discussed object detectors in the specific framework of Cooperative Awareness, so as to employ realistic traffic data.

Another limitation of the developed methodology is related to the apparent in vehicular applications network delays. Since vehicles frequently exchange measurements and estimations, it is expected that the delay introduced by V2V communication impacts on the performance of Cooperative Awareness. However, we have realistically simulated the effect of delays in our framework, so as to initially evaluate their footprints. Each vehicle broadcasts CAM messages at least every 100 ms, while the maximum delay introduced by V2V communication can reach 300 ms at heavy traffic density of 0.1 vehicles/meter. Therefore, for every iteration round of the proposed diffusion-based algorithm we have at most 400 ms delay, which implies that vehicle i receives the location vector of its neighbours estimated 4 iterations before. Integrating a network and communications simulator in the environment of CARLA, alongside the object detectors will be another major goal of ours.

In terms of the odometry robustification and assessment for the purposes of T4.5, we have derived the proposed system architecture diagram of Figure 43, through the combination of a Visual and LIDAR based SLAM solution:

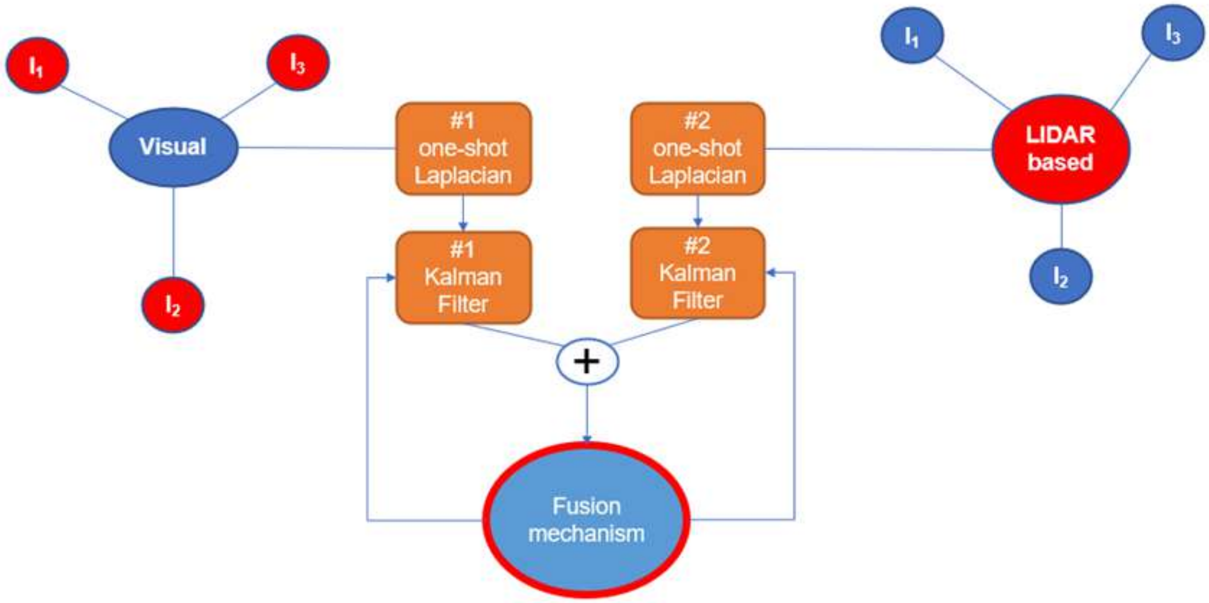


Figure 43: Proposed multi-modal fusion architecture

Its core idea is to couple the extracted pose from Visual (LIDAR based) approach with the landmarks detected by the LIDAR based (Visual) solution.

The simulation framework must provide data both for the object detection and the cooperative localization algorithms. Even though Carla is ideal for providing realistic data generated from visual and lidar sensors, it cannot simulate the CAM messages which are used for the V2V and V2X communications. To overcome this difficulty, we have utilised Artery which enables V2X simulations based on the ETSI ITS-G5 protocols. Artery started as an extension of Veins, which is an open-source framework for running vehicular network simulations, but now can be used independently. Like Veins, Artery is based on two popular open-source frameworks, Omnet++ which is an event-based network simulator and SUMO which is a road traffic simulator. SUMO, which acts as an intermediate between Carla and Artery provides the Traffic Control Interface (TraCI) which uses a TCP based client/server architecture to provide access to SUMO.

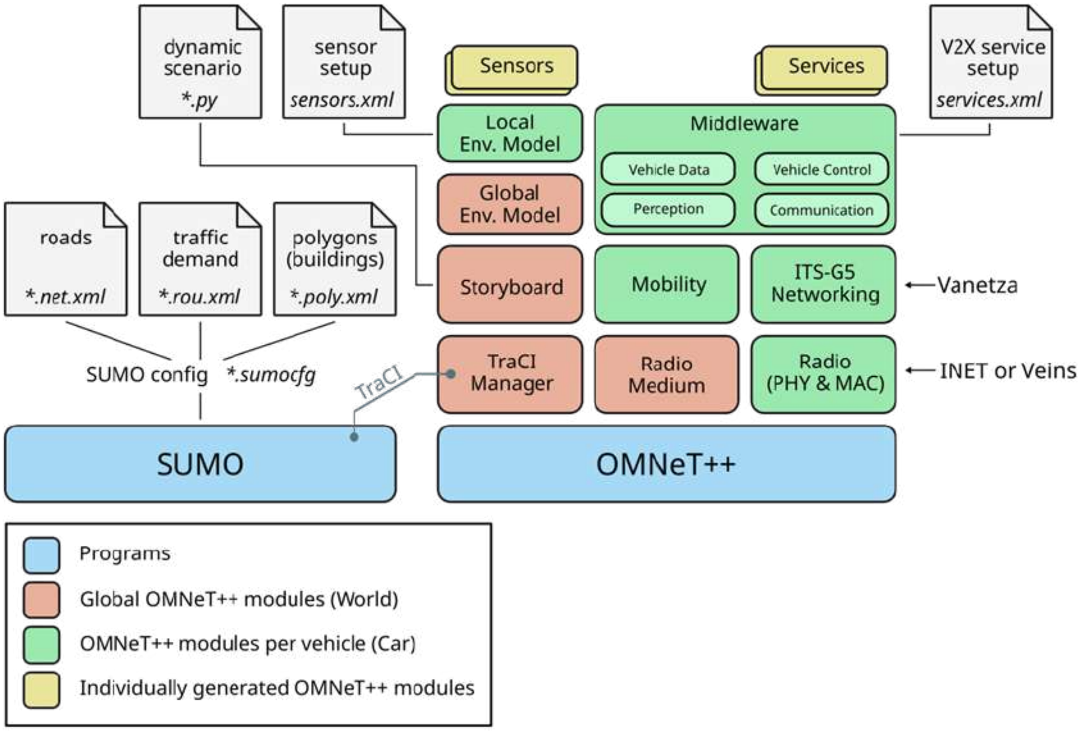


Figure 44: Artery architecture

The following figure demonstrates the whole setup of the simulation framework and how the object detection, multi-modal fusion and cooperative localization algorithms are integrated. After the initialization of the SUMO simulation server, the ego vehicle and other actors are spawned in Carla. A python script is executed for the synchronisation of the entities between the two simulation environments. Artery via the TraCI interface acquires all the relevant data about the position and the routes of the vehicles. The simulation is controlled by Artery which ticks after the completion of every discrete step and publishes the CAM messages from the neighbouring vehicles of the ego under the ROS topic */artery/carla0/NeighCAM*. Carla-ros bridge also publishes sensor data to the ROS environment and more specifically, rgb images, lidar data and the ground truth position of the ego vehicle. The odometry algorithms subscribe to the relevant topics, depending on their input modality, and produce estimated poses and point clouds which are published under the corresponding topics. The multi-modal fusion ROS node implements a fusion algorithm which utilises the output of the odometry algorithms aiming at the robustification of the ego vehicle's pose estimation. The cooperative localization ROS node uses the CAM messages from Artery and in combination with the output of the object detector node estimates the position of the ego-vehicle. Finally, a logger ROS node is used for capturing all the outputs of the algorithms along with the ground truth acquired directly from Carla.

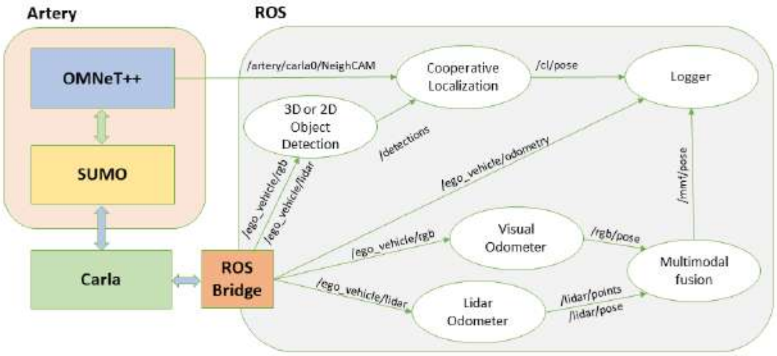
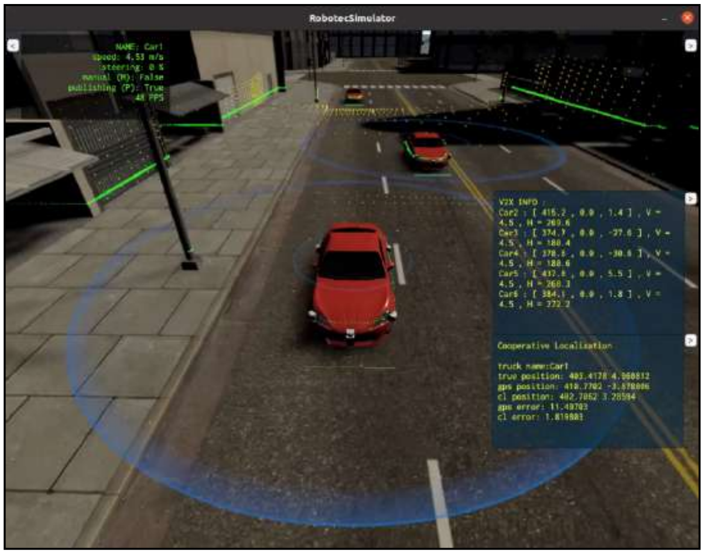


Figure 45: Simulation framework setup

2.3.1.1.2 Robotec Simulation Framework

In parallel to the Artery-CARLA simulation setup for validation of cooperative awareness scenarios, Robotec has been working on an alternative simulation solution enabling real-time simulation of multiple vehicles communicating with each other through the V2X communication interface. The solution is based on a Simulator - Unity based ROS2 enabled simulator, simulating multiple ego-vehicles equipped with required sensors. RoSi works as the central simulator, managing and executing scenarios, generating sensors data, and storing results of the experiments.



```
NAME: Car1
speed: 4.53 m/s
steering: 0 %
manual (M): False
publishing (P): True
48 FPS
```

```
V2X INFO :
Car2 : [ 415.2 , 0.0 , 1.4 ] , V =
4.5 , H = 269.6
Car3 : [ 374.7 , 0.0 , -27.6 ] , V =
4.5 , H = 180.4
Car4 : [ 378.5 , 0.0 , -30.6 ] , V =
4.5 , H = 180.6
Car5 : [ 437.8 , 0.0 , 5.5 ] , V =
4.5 , H = 268.3
Car6 : [ 384.1 , 0.0 , 1.8 ] , V =
4.5 , H = 272.2

Cooperative Localization

truck name:Car1
true position: 403.4178 4.060812
gps position: 410.7702 -3.578006
cl position: 402.7062 3.28594
gps error: 11.49703
cl error: 1.819883
```

Figure 46: RoSi view of V2X messages together with GPS and Cooperative Localization errors.

An additional component of the simulation setup prepared by Robotec is the V2X co-simulator integrated with RoSi via ROS2 interface.

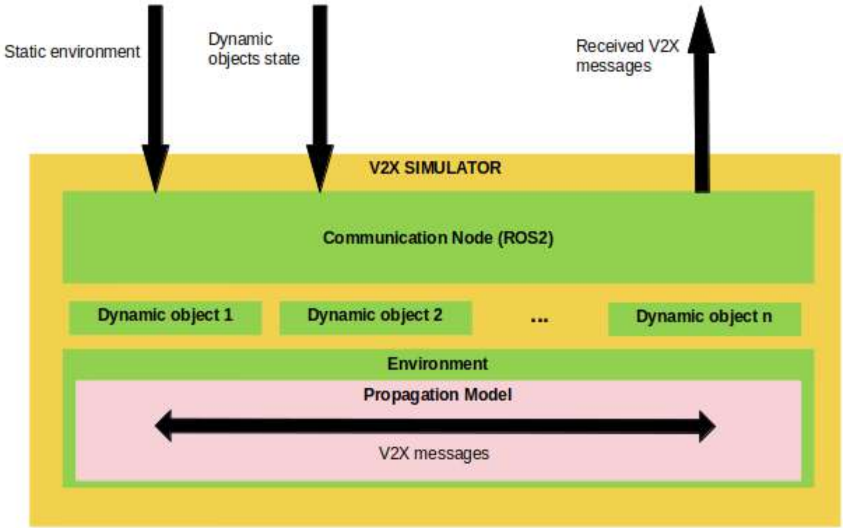


Figure 47: High level architecture of V2X simulator.

The general concept of the V2X simulator is that it creates a copy of the world used in the AV Simulator (In this case RoSi) and simulates the propagation of signals in a separate program that can be deployed on a different machine. The V2X Simulator can be integrated with any simulator with ROS2 support. Messages received by each traffic object supporting V2X communication are sent back to AV Simulator enables further processing.

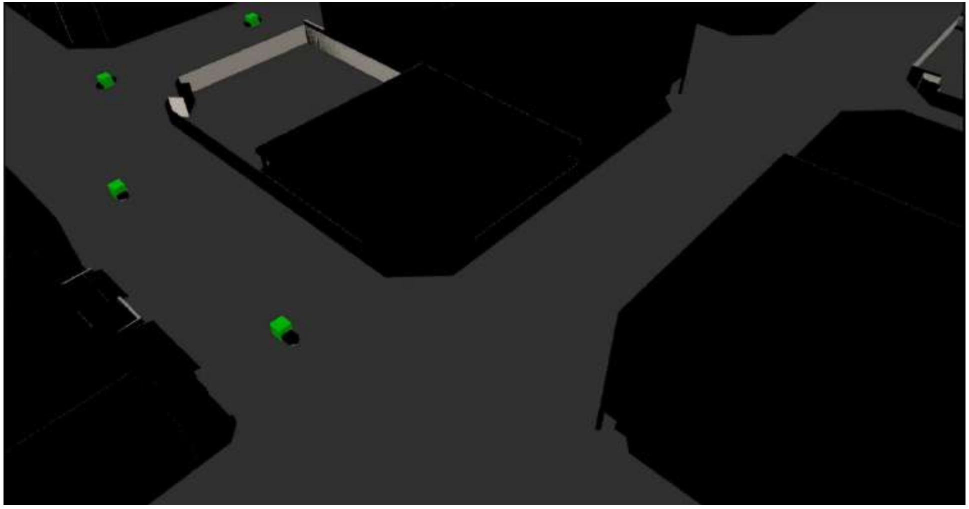


Figure 48: Visualization of the copy of simulated world in V2X simulator

Robotec Simulation framework will be used for validation of the same set of scenarios and use cases as setup with CARLA and Artery.

2.3.1.2 *Testing environments*

Carla vehicle simulator constitutes an ideal solution for gathering synthetic data from multiple and diverse sensors attached onto vehicles performing various scenarios under controllable lighting and weather solutions. In this chapter we describe different scenarios under various environmental conditions implemented in the simulator focusing on generating data under conditions that presumably would result in degrading the performance of the aforementioned solutions that deal with the estimation of the position of the ego vehicle and the detection of objects of interest. There is a huge variety of odometry and mapping techniques, aiming at achieving low-drift in motion estimation. However, most of them provide robust results only on a few use cases with specific sensor configurations. Hence, the goal is to provide data that would challenge these solutions. A brief description of these scenarios is given below.

1) Roads with positive or negative slope

The most complex town, with a 5-lane junction, unevenness, a tunnel, and more have been evaluated in this section. The SoA odometry approaches in extreme conditions such as tunnels or roads with a positive or negative slope will be examined here.

2) Rural environment

A rural environment with narrow roads, barns and hardly any traffic lights has been evaluated here. This scenario investigates how SoA approaches can handle an environment without many buildings or other static features like poles, traffic lights, etc. However, only a few clear objects are available for landmarks, especially in real cases. One of the challenges of the odometer here is to detect sufficiently discriminative features.

3) Sequence with an infinite loop

A dedicated scenario with an infinite loop has been generated here to examine loop closure, which is very important for global mapping. More specifically, the loop closure is related to the drift correction of the ego-vehicle based on the recognition of a previously visited place.

4) Modify weather and lighting conditions

A basic town layout consisting of “T junctions” and multiple weather conditions has been evaluated here. The weather and lighting conditions can be chosen from a set of predefined settings (ClearNoon, CloudyNoon, WetNoon, WetCloudyNoon, MidRainyNoon, HardRainNoon, SoftRain-Noon, ClearSunset, CloudySunset, Wet-Sunset, WetCloudySunset, MidRainSunset, HardRainSunset and SoftRainSunset). These weather conditions can model seasonal changes. Hence, this scenario investigates whether SoA odometry and slam methods are capable of handling dynamic environments and extracting robust features against those changes. Two cases with dynamic weather conditions have been generated here and they differ in the point cloud generation. The first one consists of original point clouds. In the second case, points in the cloud have been dropped off to simulate noise due to external perturbations.

5) Existence of moving objects on the 50% focus expansion of cameras Data from a basic town with multiple vehicles that cover the 50% focus expansion of cameras have been generated for this scenario. The scope of this experiment is to examine the solutions in areas with many dynamic objects in the scene. For instance, some algorithms may fail to detect whether the front vehicle is moving or not. For this task the images have been cropped from 1280x960 to 1280x640.

6) Complex city environment

A city environment with different environments such as an avenue or promenade and more realistic textures have been evaluated here. The experiments, in this case, will examine whether odometry and slam approaches can extract robust landmarks in a map of superb visual quality, with detailed buildings and realistic roads.

7) Long highways

An environment with long highways with many entrances, exits and roundabouts has been assessed in this scenario. The extracted landmarks in this case have larger shifts due to the high speed of the vehicle and sometimes move out of the field of view. Other landmarks may exist at high distances and as a result, their shifts in the image plane are noisy. Hence, the SoA approaches will be examined whether they can track the extracted landmarks and provide robust results.

2.3.1.3 *Simulation Based Evaluation*

2.3.1.3.1 Dataset Acquisition Process (CARLA)

Carla vehicle simulator is an ideal solution for gathering synthetic data for multiple and diverse sensors attached onto vehicles performing versatile scenarios under controllable lighting and weather solutions. Unreal Engine, on which Carla is based, provides realistic and high-fidelity images. Except from the ability to create easily custom environmental conditions another advantage of the simulator is the ability to easily track and store the ground truth values

Furthermore, each sensor has specific attributes which can influence their behaviour and the generated data. Sensors must be attached to a parent actor, in our case the vehicle, to follow it and gather data. The attachment to the parent can be of two types:

- "*Rigid attachmen*t": the relative movement of the sensor regarding the vehicle is strict and this is the type used in the data gathering process.

- "*Springarm"* attachment: movement is not so strict and includes little accelerations and decelerations.

Every sensor has a listen function which is called every time the sensor gathers data. The period between the successive callbacks, which defines the time when the sensor produces data, can be easily configured.

Generated data differ between the several sensor types. However, they share common properties like the frame number, the timestamp and their position in regard to their parent actor.

The sensors recorded for this dataset are shown in Table X. All of them use the Unreal Engine coordinate system (x-forward, y-right, z-up) and return coordinates in local space. Also, intrinsic, and extrinsic matrices are provided as well as timestamp files to allow synchronisation of the data. Finally, all data were collected using 30 fps. The specifications for each one of them are shown in XXXX.

Table 5: Sensors' configuration Overview

Type	Dimensions	Description
RGB camera	1280x960	Get images from the scene
Semantic segmentation camera	1280x960	Every object is classified in a different colour according to its tags
Depth camera	1280x960	Get depth values
Lidar	Velodyne 16/64	Get the 3d coordinates and intensity values.
Semantic Lidar	Velodyne 16/64	Get the index of the Carla object hit and its semantic tag
Imu	-	Provides measurements from accelerometer, gyroscope, and compass
GNSS	-	Provides current GNSS position in reference to a predefined map reference point

In addition to the sensor, CARLA provides a range of measurements associated with the state of the agent and compliance with traffic rules. Measurements of the agent's state include vehicle location and orientation with respect to the world coordinate system, speed, acceleration vector, and accumulated impact from collisions. Measurements taken for traffic rules include the percentage of the vehicle's footprint that impinges on wrong-way lanes or sidewalks, as well as states of the traffic lights and the speed limit at the current location of the vehicle. Moreover, CARLA provides access to exact locations and bounding boxes of all dynamic objects in the corresponding environment. Examples of outputs of sensors are illustrated in the next figure.

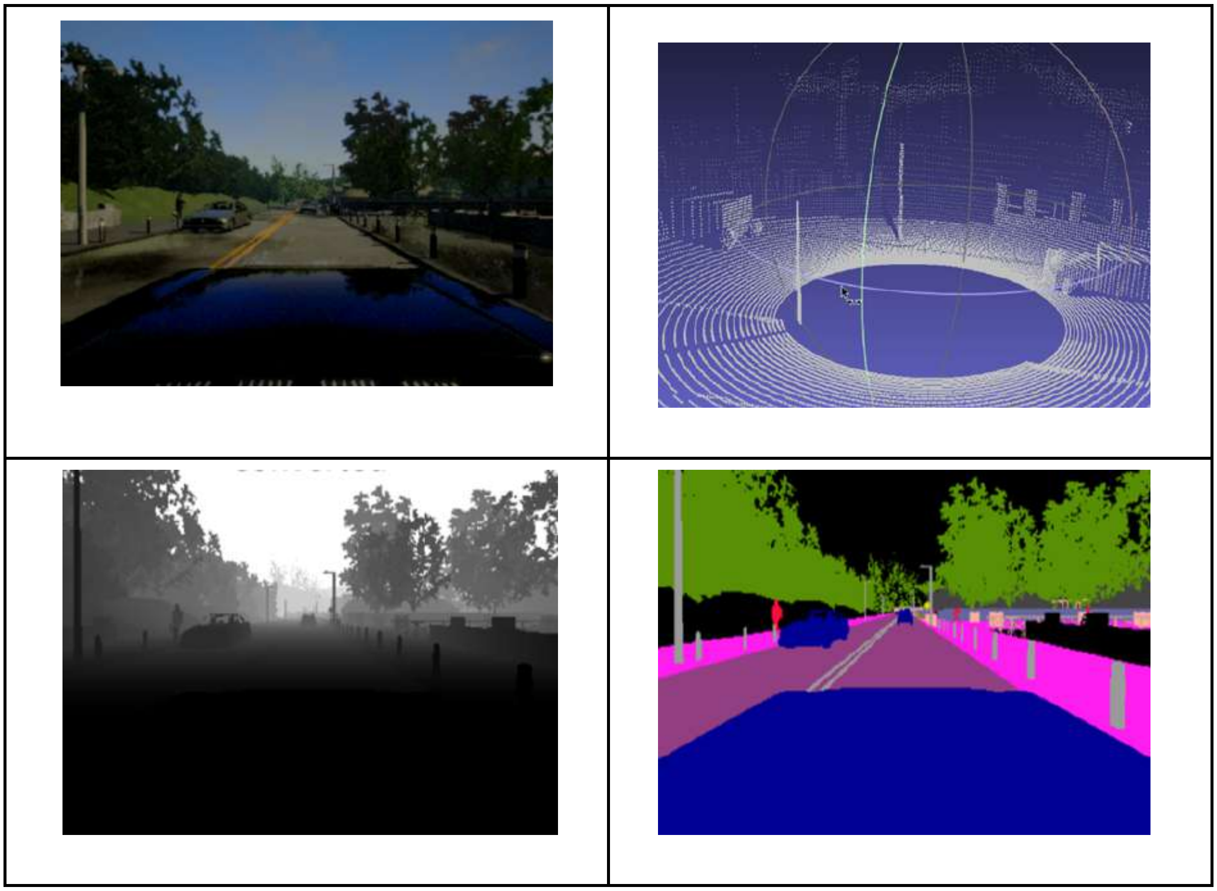


Figure 49: From Top to bottom and left to right RGB camera, Depth Camera, Lidar, Semantic Segmentation Camera

As already mentioned, we can easily gather from CARLA annotated data concerning the properties of the objects inside the virtual world. These readings can be extracted either by a relevant sensor (e.g., semantic lidar, semantic segmentation camera) or by using the Python API. In the following figure, the 3D bounding boxes of the neighbouring objects to the ego vehicle are rendered. These objects may include dynamic components (e.g., vehicles, pedestrians) or static components (e.g., traffic lights)

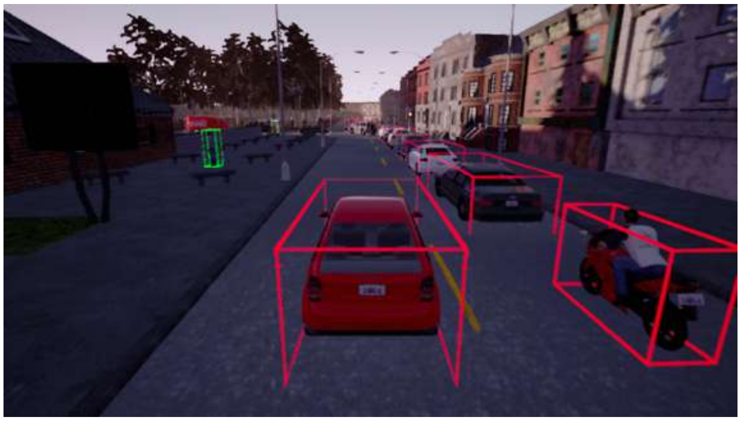


Figure 50: Rendered 3D Bounding Boxes

The following tables contain all the necessary configuration parameters for the deployment of the sensors on the ego vehicle. Besides sensor specific parameters it also includes the information about the positioning of the sensor on the vehicle.

Table 6: CAMERA SPECIFICATIONS

	Position(m)	Fov(degrees)	Size	FPS
Camera (rgb/depth/semantic)	(x,y,z) = (0,0,2)	100	1280x960	30

Table 7: LIDAR SPECIFICATIONS

	Position	Range	Channels	Points per second	Horizontal fov	Upper fov	Lower fov	FPS
Lidar semantic	(0,0,2.4)	120	64/16	100k/25k	360	10	-30	30

Table 8: IMU AND GNSS SPECIFICATIONS

	Position	Roll, Pitch, Yaw
IMU	(2,0,2)	0,0,0
GNSS	(1,0,2)	0,0,0

2.3.1.3.2 Dataset Acquisition Process (Real Vehicle)

The Set Up for dataset acquisition, Data Synchronization, recording and storage is the same as in 2.2.2.7.

2.3.1.3.2.1 *Structure of the Testing Dataset*

Cooperative Localization, Tracking and Awareness

During the experiments in the CARLA simulator, we have extracted an excel file containing all the necessary information about 150 vehicles. More specifically, each record of the file has the following format: time stamp, vehicle id, ground truth position.x, ground truth position.y, ground truth position.z, yaw, pitch, roll, yaw rate, pitch rate, roll rate, linear velocity.x, linear velocity.y, linear velocity.z. Therefore, each vehicle is associated with 14 parameters. The simulation horizon was set T = 448 time instances, with ΔT = 0.4sec. Thus, the excel file had in total 150*448 = 67200 records that had to be processed. Obviously, the post processing and evaluation procedure was done offline. The integration of the proposed schemes in the CARLA-ROS framework so as to perform online the experimental evaluation is our next goal.

Multimodal object detection

For the multimodal object detection evaluation a new CARLA dataset is employed. Since the simulator, allows to create datasets that capture different weather and lighting conditions(including wet surfaces), as well as complex city environments with traffic elements such as multiple intersections, complex lane roundabouts, or tunnels we generated a synthetic dataset where the player drives through 4 cities on day or night under clear or cloudy sky. The latter weather condition randomly generates rain that subsequently makes the road wet. Fog and low visibility effects also appear randomly. ***The following Figure*** presents examples of images picked from the compiled dataset. The dataset that is utilised in this evaluation, consists of 4423 frames in low and high traffic conditions. This set of frames is split into Ntr= 3538 and Nval= 885 frames for training and validation purposes, respectively. The objects are labelled in two categories, namely, cars and pedestrian

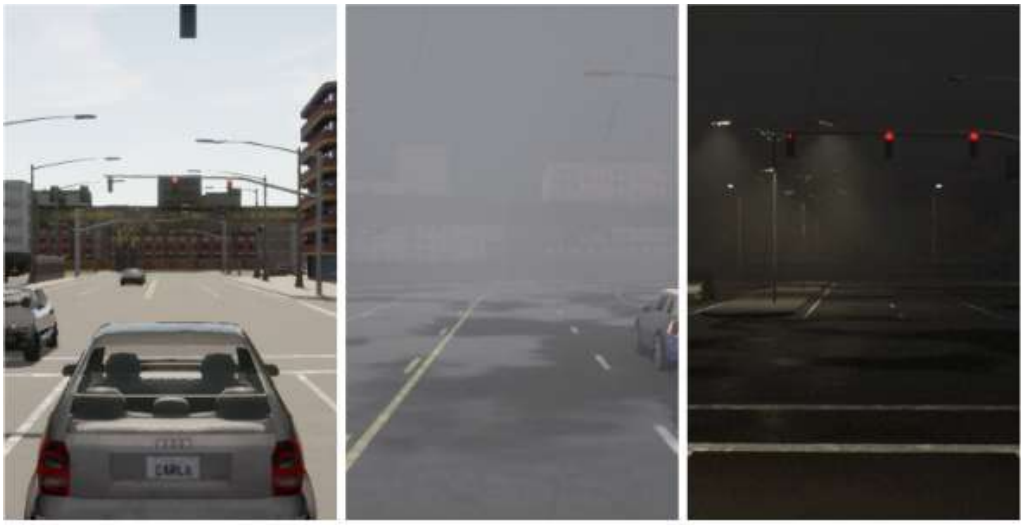


Figure 51: CARLA scenes examples

2.3.1.3.2.2 Statistical Evaluation Process

Cooperative Localization, Tracking and Awareness

For the time instant t, the proposed algorithms provide the estimated 2D position $\boldsymbol{\widehat{p}_i}$ of vehicle i which corresponds to the ground truth position $\boldsymbol{p_i}$. Therefore, the individual localization error at time t, is equal to:

$$LE(i) = \sqrt{(\boldsymbol{\widehat{p}_i} - \boldsymbol{p_i})^2}$$

Note that ego vehicle i estimates also the position $\boldsymbol{\widehat{p}_{i->j}}$ of its neighbour j. Thus, the individual neighbour localization error, which corresponds to the estimation accuracy of ego vehicle i can be used too:

$$LE(j) = \sqrt{(\boldsymbol{\widehat{p}_{i->j}} - \boldsymbol{p_j})^2}$$

In the following figures, the evaluation of the proposed schemes in the case of individual localization error is presented. As the reader may see, Laplacian Extended Kalman Filter for Cooperative Awareness (LEKF-CA) succeeds in significantly reducing both self GPS error (0.7 m instead of 6.86 m) as well as the GPS of neighbours (1.18 m and 0.85 m instead of 5.62 m and 2.8 m, respectively). Similar behaviour is also apparent in the bottom subfigure, where the Cooperative Awareness has been realised though the information diffusion-based scheme. The latter achieved once again high enough accuracy for the self and neighbouring locations.

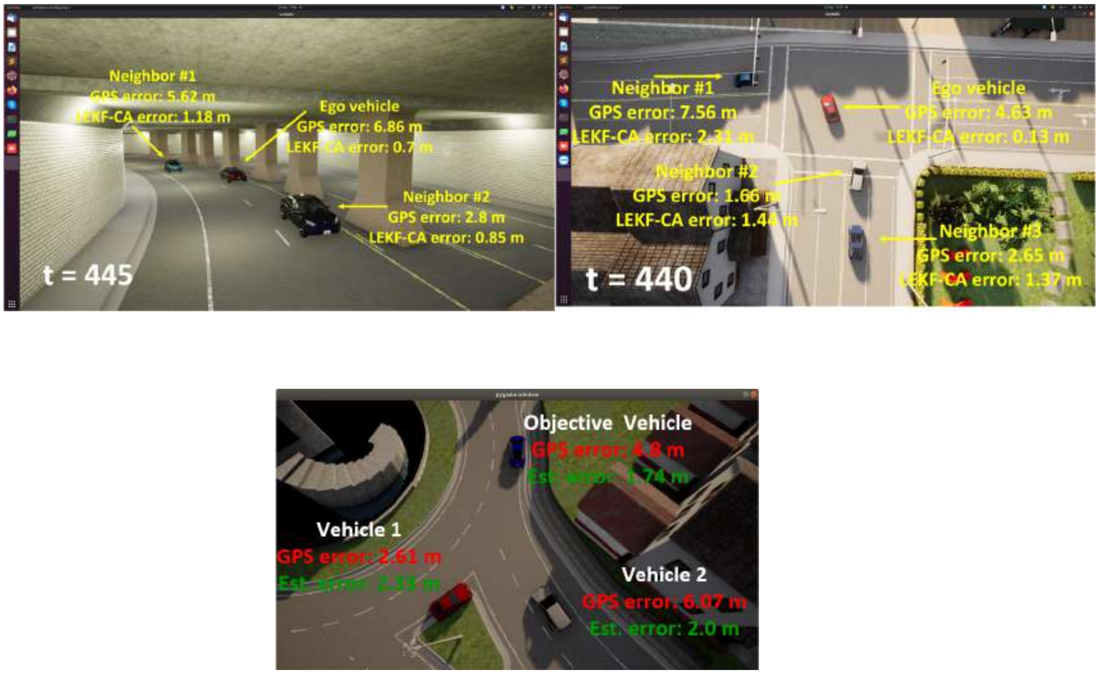


Figure 52: Evaluation of Cooperative Awareness solutions in the simulated framework

Multimodal object detection

Multimodal object detection combines the outcome of 2D and 3D object detectors in a late fusion process taking as output the union of both detectors. For the evaluation process the outcome of each detector is evaluated separately using standard KITTI metrics and compared to the fusion outcome . The following figure presents the fusion outcome

The official KITTI evaluation detection metrics include birds eye view (BEV), 3D, 2D, and average orientation similarity (AOS). The 2D detection is done in the image plane and average orientation similarity assesses the average orientation (measured in BEV) similarity for 2D detections [7]. The KITTI dataset is categorised into easy, moderate, and hard difficulties, and the official KITTI leaderboard is ranked by performance on moderate. For the sake of self-completeness, easy difficulty refers to a fully visible object with a minimum bounding height box of 40px and max truncation of 15%, moderate difficulty refers to a partially occluded object with a minimum bounding box height of 25px and max truncation of 30% and hard difficulty refers to a difficult to see an object with a minimum bounding box height of 40px and max truncation of 50%. Each 3D ground truth detection box is assigned to one out of three difficulty classes (easy, moderate, hard), and the used 40-point Interpolated Average Precision metric is separately computed on each difficulty class. It formulated the shape of the Precision/Recall curve as

$$\mathbf{AP}|_R = \frac{1}{|R|} \sum_{r \in R} \rho_{interp}(r)$$

averaging the precision values provided by ρ_interp (r), according to [21]. In our setting, we employ forty equally spaced recall levels,

$$R_{40} = \{1/40, 2/40, 3/40, \ldots, 1\}$$

The interpolation function is defined as

$$\rho_{interp}(r) = \max_{r' : r' \geq r} \rho(r')$$

where ρ(r) gives the precision at recall r, meaning that instead of averaging over the actually observed precision values per point r, the maximum precision at recall value greater or equal than r is taken.

2.3.1.3.2.3 Inter communication simulation (V2X simulator)

As it was defined in D2.4, the CPSoSAware inter communication model for the cooperative awareness scenario is based in V2X messages interchange following the ETSI G5 ITS communication protocol architecture. Nowadays, this architecture is prepared to work with two radio technologies, IEEE 802.11p and LTE-PC5 Mode 4.

Additionally, in D6.3 and D6.4 we planned to study cooperative awareness scenarios which include situations with potential risks of collisions between vehicles or between vehicles and pedestrians. These situations can be mitigated using information obtained from V2X message transmission between vehicles. Nevertheless, in a real situation, the transmission of radio messages suffers from multiple factors as the attenuation produced by obstacles in the propagation path, reflexion and refraction propagation in situations of Non-Line-of-Sight and, mostly, from interferences produced by other vehicles in the vicinity transmitting at the same time or hidden terminal situations. It is for these reason that the use cases to analyse, apart from the vehicles and Vulnerable Road Users (VRUs) involved in the potential collision, require to contemplate multiple vehicles in the area producing radio interferences as it would be in a real situation.

This section describes the first tests produced with the V2X communications software simulator developed in T4.4, taking into account scenarios with multiple vehicles and comparing the two current radio technologies, IEEE 802.11p and LTE-PC5 which we presented in the 2021 IEEE International Mediterranean Conference on Communications and Networking [1].

Description of the used radio technologies: IEEE 802.11p and LTE-PC5 Mode 4

The IEEE pioneered the first radio technology aimed at V2X communications with the IEEE 802.11p standard published in 2010, also known as Direct Short-Range Communications (DSRC). This technology operates in the Intelligent Transport Systems (ITS) band at 5.9 GHz and is based on the standard IEEE 802.11a. On the other side, the 3rd Generation Partnership Project (3GPP) specified in Release 14, in 2017, an evolution of the Long-Term Evolution (LTE) PC5 interface, known as Cellular V2X (C-V2X), which in its Mode 4 variant, can operate at the ITS 5.9 GHz band in an infrastructureless decentralized manner. A clear technical advantage of LTE-PC5 Mode-4 over IEEE 802.11p is a more advanced physical layer, which results in improved range under high mobility conditions, as verified in [2].

On the technical side, both the IEEE and the 3GPP, continue to enhance their support for vehicular communications. The IEEE launched the 802.11bd working group, which will update the physical layer of 802.11p to the one used in 802.11ac (Very High Throughput). On the other hand, the 3GPP defined in Release 16 C-V2X extensions for the 5G-New Radio technology, known as NR-V2X. This technology does not target day-1 safety services in the ITS band (5.9 GHz), but will focus on value added services using licensed spectrum [3]. Therefore, the candidate technologies to support day-1 safety services, as cooperative awareness, are IEEE 802.11p and its successor 802.11bd, and 3GPP LTE-PC5 Mode 4.

A. IEEE 802.11p, IEEE 802.11bd

IEEE 802.11p is based on the widely used IEEE 802.11a, or OFDM (Orthogonal Frequency - Division Multiplexing) Physical (PHY) layer in standard IEEE Std 802.11-2020, which removes the requirement of having to be associated to a Basic Service Set (BSS) to be able to transmit or receive data, this is why IEEE 802.11p is called "Outside the Context of a BSS" (OCB).

The PHY layer uses the same modulation and coding schemes as IEEE 802.11a, but the channel bandwidth is 10 MHz instead of 20 MHz. Therefore, data rates are halved, beginning at 3 Mbps and reaching up to

27 Mbps, being 6 Mbps the default data rate (QPSK modulation with a coding rate of 0,5). In addition, while IEEE 802.11a works at frequencies between [5150 - 5730] MHz, IEEE 802.11p uses the ITS band [5855 - 5925] MHz. The newer version of this standard, IEEE 802.bd, has as main goals to define, at least, i) one working mode that achieves twice the MAC throughput of 802.11p with relative velocities up to 500 kmph, ii) one mode that achieves twice the communication range of 802.11p, and iii) one mode that is compatible with 802.11p devices. Overall, it is expected that the spectrum efficiency and coverage of 802.11bd will be similar to the physical layer of LTE-PC5 Mode-4 [3]. In any case, the Medium Access Control (MAC) mechanisms of IEEE 802.11p and 802.11bd have been continually based on the well-known Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism for backwards compatibility purposes, and works as follows. Before initiating a transmission, a station senses the channel to determine whether it is busy. If the medium is sensed idle during a period of time called the Arbitrary Inter-frame Space (AIFS), the station is allowed to transmit. If the medium is sensed busy, the transmission is delayed until the channel is idle again. In this case, a slotted binary exponential backoff interval is uniformly chosen in [0, CW - 1], where CW is the contention window. The backoff timer is decreased as long as the channel is sensed idle, paused when a transmission is in progress, and resumed when the channel is sensed idle again for more than the AIFS. When the backoff timer expires, the station attempts transmission. After each unicast data frame successfully received, the receiver transmits an acknowledgment (ACK) frame after a Short Inter-frame Space (SIFS) period. In case the frame is transmitted in multicast or broadcast mode, as it happens in the transmission of Cooperative Awareness Messages, the ACK is not transmitted. The value of CW is set to its minimum value, *CWmin*, in the first transmission attempt and after each successful transmission; and increases in integer powers of 2 at each retransmission, up to a pre-determined value *CWmax*.

B. LTE-PC5

Figure 53 illustrates the resource grid structure of the LTE-PC5 interface used to allow direct vehicle to vehicle communications. In the time domain, the resource grid is divided into subframes of 1 ms. In the frequency domain, a subframe is divided into physical Resource Blocks (RBs) that span 180 kHz. The number of RBs per subframe depends on the available channel bandwidth, with 50 RBs used in a 10 MHz carrier. QPSK and 16QAM modulation schemes are supported for data transmissions.

LTE-PC5 groups RBs in the same subframe into subchannels, and allows to allocate resources with a subchannel granularity. A subchannel is composed of control information and user data. The control portion occupies 2 RBs and is known as the Sidelink Control Information (SCI). The group of RBs carrying user data in a subchannel is known as a Transport Block (TB) and occupies a variable number of RBs depending on the message size. In this paper we assume that the SCI and the TB are configured to be adjacent in the resource grid and that 5 RBs per subchannel are used, resulting in 10 subchannels per subframe in a 10 MHz carrier. Other configurations are possible, and its choice is relevant for the system's efficiency [4].

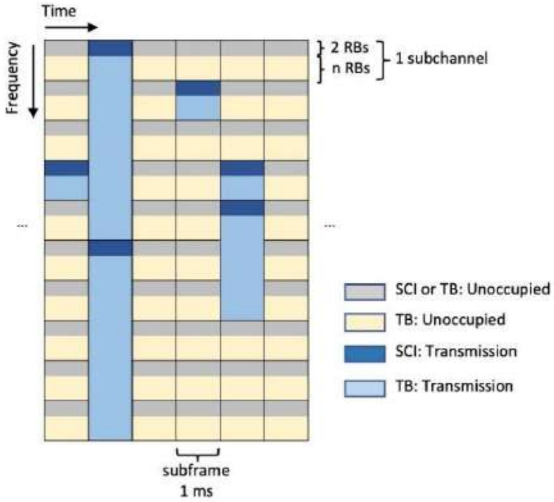


Figure 53: **Structure of the LTE-PC5 resource grid with 10 subchannels per subframe**

LTE-PC5 provides two mechanisms to implement resource allocation, namely Mode 3 and Mode 4. In Mode 3 vehicles are assumed to be under the coverage of an eNB, and it is the eNB who allocates the resources that are used in the sidelink or PC5 interface. In Mode 4 vehicles use the Sensing-based Semi-Persistent Scheduling (SPS) distributed channel access mechanism to decide on the resource allocation. Its main idea is to proactively avoid collisions by sensing the channel during a time window. Given our interest on the ITS safety band at 5.9 GHz we hereafter focus, on C-V2X Mode 4.

Figure 4 illustrates the SPS algorithm. A sensing window of typically 1000 ms is considered, where the User Equipment (UE) decodes the received SCIs, and keeps track of the Sidelink Received Signal Strength Indicator (S-RSSI) in each subchannel. When a transmission is requested, a set of potential candidate subchannels is considered within a resource selection window, which has a maximum of 100 ms to limit channel access delay. For each candidate subchannel in the resource selection window, a corresponding set of subchannels in the sensing window is established assuming a periodicity of 100 ms. Consequently, the set of candidate subchannels is pruned removing all the subchannels whose corresponding set in the sensing window was not monitored, or was received with an average S-RSSI exceeding a preconfigured power threshold, which indicates that this subchannel is occupied by another transmitter. If after the pruning process less than 20% of the subchannels in the resource selection window remain, the power threshold is increased by 3 dB and the process is repeated. Finally, the UE selects the subchannel to be used with a random uniform distribution among the best 20% subchannels, in terms of lowest average S-RSSI, remaining in the resource selection window.

The selected resource is maintained for a number of consecutive packet transmissions known as the Resource Counter (RC), which is randomly selected between (*RCmin*, *RCmax*). The RC is decremented by one on each packet transmission. Finally, when the RC goes to zero a keep probability, p_{keep} = (0, 0.8), is used to determine if a new resource selection needs to be triggered. Previous studies have shown that p_{keep} has a small impact on the performance of the SPS algorithm [5].

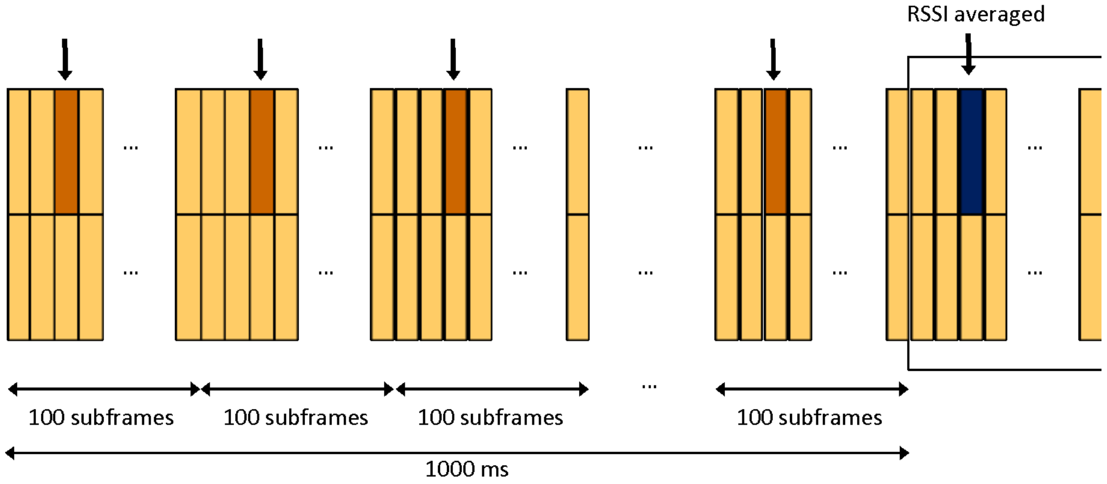


Figure 54: **Sensing Based Semi-Persistent Scheduling**

In the rest of this study, we assume the following configuration parameters of the SPS algorithm: initial pre-configured power threshold of 95 dBm, p_{keep} = 0.8, and (*RCmin*, *RCmax*) = (5, 15), according to the application generation rate of the Cooperative Awareness Service.

<u>The Cooperative Awareness Service</u>

The Cooperative Awareness Basic Service has been standardized by the ETSI in Europe [6]. The goal of the Cooperative Awareness Service is to allow vehicles to maintain awareness of other vehicles in proximity. This is achieved by means of the Cooperative Awareness Messages (CAM) that are transmitted periodically, and contain information such as time, position, and motion state of the vehicle. The default transmission interval of the CAM message is 0.1 seconds but this interval can be increased up to 1 second depending on the channel congestion status using the Decentralized Congestion Control (DCC) mechanism that operates on top of the MAC layer. In this first study, DCC has not been explicitly considered as we focus on comparing MAC algorithms of IEEE 802.11p and LTE-PC5 Mode-4 without the interference of any other layer. However, a representative set of CAM intervals is considered to model the effect of DCC. A detailed study of the behaviour of DCC can be found in [7].

After a CAM is authenticated, the receiving vehicle updates the Local Dynamic Map (LDM) to record the position of the vehicle that transmitted the CAM. In the following results we assume that an entry in the LDM is erased after 2 seconds without receiving any CAM from the corresponding vehicle.

<u>KPIs of the Cooperative Awareness service: NAR and Position Error</u>

To understand the performance of the CAM service we propose two Key Performance Indicators (PKI) that capture the efficiency and the accuracy of the CAM service.

The Neighborhood Awareness Ratio (NAR(r)), first introduced in [9], describes the portion of cars registered in a given vehicle's LDM from all the cars that are present at a given radius *r* from that vehicle. Thus, in a system with *N* vehicles we can define the system wide NAR at distance *r* as:

$$NAR(r) = \frac{1}{N}\sum_{i\in N} \frac{|LDM(i)_r|}{|S(i,r)|}$$

, where *LDM(i)*$_r$ is the number of LDM entries in vehicle *i* located at a distance below *r* from vehicle *i* and *S(i,r)* is a function that counts all vehicles within distance *r* of vehicle *i*. Thus, the NAR metric captures the efficiency of the CAM service in terms of the probability of missing neighboring cars. This metric is critical for the cooperative awareness service because missing neighboring cars affects the performance of safety applications relying on this service.

Another critical metric for the performance of the applications using the cooperative awareness service is to understand how reliable are the entries of LDM database in a given vehicle. Notice that having an entry in the LDM database does not guarantee that the actual position of the represented vehicle corresponds to the one recorded in the LDM, since neighboring cars move between CAM updates. Thus, we can define an average Position Error (PE) from the perspective of a given vehicle *i* applied to its neighbours inside a radius *r* as:

$$PE(i,r) = \frac{1}{|LDM(i)_r|}\sum_{j\in LDM(i)_r} \sqrt{|LDM(i,j) - pos(j)|^2}$$

, where *LDM(i,j)* represents the stored coordinates of vehicle *j* in the LDM of vehicle *i*, and *pos(j)* represents the coordinates of the actual position of vehicle *j* in the scenario. Hence, we can define the average PE from the system perspective as:

$$PE^{avg}(r) = \frac{1}{N}\sum_{i\in N} PE(i,r)$$

Notice that while NAR captures the efficiency of the CAM service, the Position Error captures the accuracy of the service.

Performance evaluation

The following results have been obtained with a first version of the V2X simulator that will be continuously improved during the third year of the project.

For IEEE 802.11p simulator we extended the Veins simulation platform, which uses 802.11p models available in OMNeT++ and the SUMO traffic simulator following a Krauss vehicle mobility model [8], to include the two new metrics (NAR and PE_{avg}). To model LTE-PC5 Mode-4 we also departed from the Inet module of OMNeT++ and developed a detailed implementation of the LTE-PC5 frame structure and of the SPS algorithm described previously. As our first intention is to evaluate the MAC algorithms of 802.11p/bd and LTE-PC5, we calibrated the physical layers of the 802.11p and LTE-PC5 models to deliver similar coverage ranges and Packet Reception Ratio (PRR) curves when operating with a 10 MHz carrier at 5.9 GHz, transmission power of 200 mW, propagation factor of 3, Rayleigh fading and background noise of -110dBm.

The considered CAM messages have a size of 185 bytes including GeoNetworking and Basic Transport protocols, which translate into an airtime of 344 microseconds in 802.11p using the default data rate of 6 Mbps implemented with a QPSK modulation and 0,5 coding rate, and span 3 subchannels in LTE-PC5, being our configuration 5 RBs per subchannel, MCS index of 7 with QPSK modulation. Additionally, 802.11p is configured with AIFS of 110 microseconds and a *CWmin* of 15. LTE-PC5 does not use HARQ retransmission.

Our goal is to understand the impact of the LTE-PC5 Mode-4 and 802.11p MAC layers on application-level metrics. It is thus essential to analyze both technologies under varying road congestion and vehicle mobility. For this purpose, we define two basic scenarios: i) Fast Highway scenario, representing 1 km highway with 8 lanes (4 lanes on each direction), where on average we have approximately 320 vehicles moving at 100 kmph (1 car every 25 meters, in each lane), and ii) Slow Highway with the same properties, where on average we have 700 vehicles moving at 10 kmph (1 car every 11.4 meters, in each lane). The mobility models in SUMO impose larger inter-car distances when driving faster.

In addition to vehicle speed and density, the other key factor impacting Cooperative Awareness Service is the CAM interval, which we set to 0.1 seconds, 0.2 seconds and 1 second. These CAM interval values are selected to reflect the minimum interval defined by the standard, as well as representative intervals allowed by DCC. Note that analysing 802.11p and LTE-PC5 under a set of fixed CAM interval allows to extrapolate the behavior under a DCC controlled CAM interval.

This analysis is structured as follows. First, we look at PRR over distance under different CAM interval settings. These results allow us to lay the necessary foundation to understand the behavior of the application layer KPIs. Subsequently, we evaluate the Cooperative Awareness Service investigating NAR and Position Error metrics while looking separately at the Fast Highway and Slow Highway scenarios, and at the impact of the CAM interval.

All metrics reported are obtained averaging the results of multiple simulation runs over distance buckets defined as the distance between transmitter and receiver.

A. Packet Reception Ratio (PRR)

Figure 5 reports the PRR over distance for the LTE-PC5 Mode-4 (blue solid lines) and 802.11p (red dashed lines) technologies in the Slow Highway scenario for a varying CAM interval of 0.1 seconds, 0.2 seconds and 1 second. We choose the Slow Highway scenario because higher levels of network congestion are attained. Looking at the case when CAM interval is 1 second, we can see that both technologies deliver a similar PRR, since their PHY layers have been calibrated to have the same coverage range and channel congestion is light. In this case, all error transmissions are due to propagation loses. Decreasing the CAM interval to 0.2 seconds and 0.1 seconds, increases channel congestion and results in an increasing gain of LTE-PC5 Mode-4 over 802.11p. The reason for the increased performance of LTE-PC5 Mode-4 lies in the design of the SPS algorithm, which, unlike CSMA/CA, is guaranteed to avoid collisions when there is enough capacity and vehicles can sense each other, and can even deliver a better performance in the case of hidden nodes, if within the last sensing window (1000 milliseconds) two hidden nodes happened to be in coverage of each other.

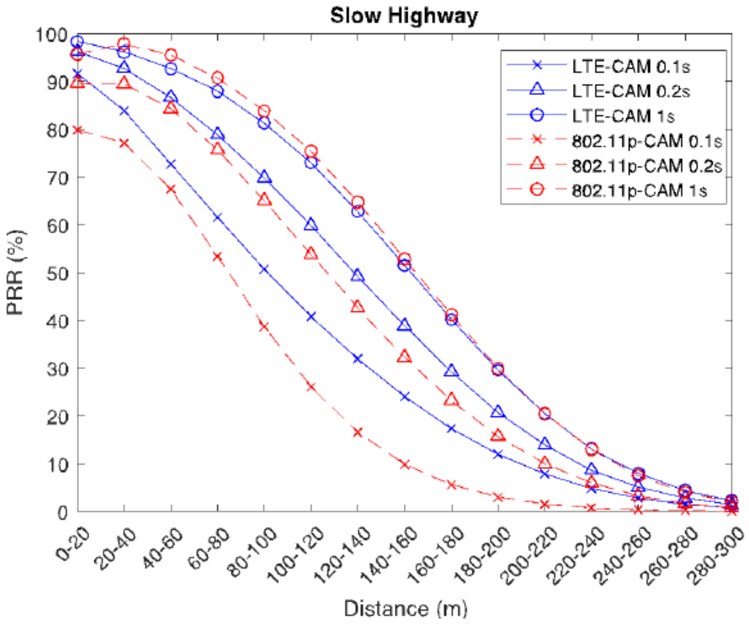


Figure 55: **PRR over distance with a varying CAM interval for LTE-PC5 Mode-4 (blue solid lines) and 802.11p (red dashed lines)**

B. Increasing vehicle density: Fast and Slow Highway

Figure 56 and Figure 57 depict respectively the results of NAR and Position Error metrics for LTE-PC5 Mode-4 and 802.11p when comparing the Fast Highway and Slow Highway scenarios, using the default CAM interval of 0.1 seconds.

Looking at Figure 56 we can see how the NAR is significantly better than the PRR. The reason is that the application layer of the Cooperative Awareness Service declares a vehicle as lost after not hearing a CAM for 2 seconds, which allows the service to recover from sporadic CAM reception losses. LTE-PC5 Mode-4 outperforms 802.11p in both scenarios, although the gain is significantly higher in the Slow Highway scenario, given that congestion is higher in this case. In this scenario neighbours are reliable tracked (NAR > 90%) up to 200 meters in LTE-PC5 Mode-4, and up to 160 meters in 802.11p.

Figure 57 depicts the position error of the Cooperative Awareness Service, namely the error between the recorded position of a vehicle in the LDM versus the actual position of this vehicle. Boxplots are used to depict this metric with the box and whiskers representing respectively the 25-75 and the 5-95 percentiles of the distribution, and the crosses representing outliers. Unlike NAR, we can see in this metric that the Fast Highway scenario results in higher errors than the Slow Highway, which is as expected because vehicles move faster and traverse more distance between successive CAM updates. It is also worth noting than in this case 802.11p results in slightly lower errors than LTE-PC5 Mode-4, being the reason that 802.11p packets are sent immediately if the channel is empty whereas LTE-PC5 Mode-4 may introduce a delay of up to 100 ms due to the SPS resource selection window. However, lower errors of 802.11p translate only in a moderate advantage in practice. For example, for vehicles located between 0-20 meters in the Fast Highway scenario, LTE-PC5 Mode-4 results in median and worst-case errors of 2 and 11 meters, and 802.11p of 1 and 10 meters. Worst case errors are caused by consecutive CAM losses due to collisions or channel errors. Despite being infrequent, these worst-case errors deserve further study as they may result in dangerous situations on the road. Moreover, in the presented results we assume that vehicles use ideal positioning, but real applications will have to cope with additional position errors introduced by localization systems whose precision range from several centimetres to several meters.

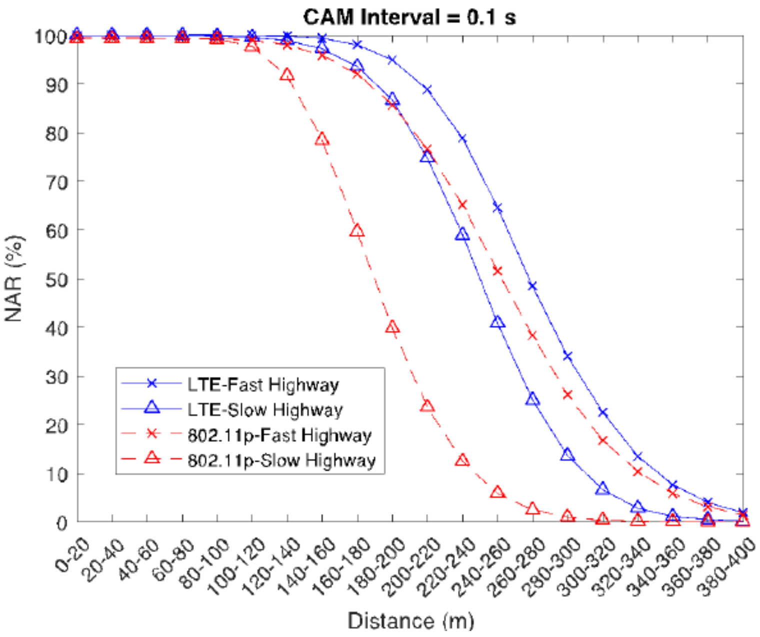


Figure 56: **NAR for LTE-PC5 Mode-4 and 802.11p in Fast and Slow Highway scenarios**

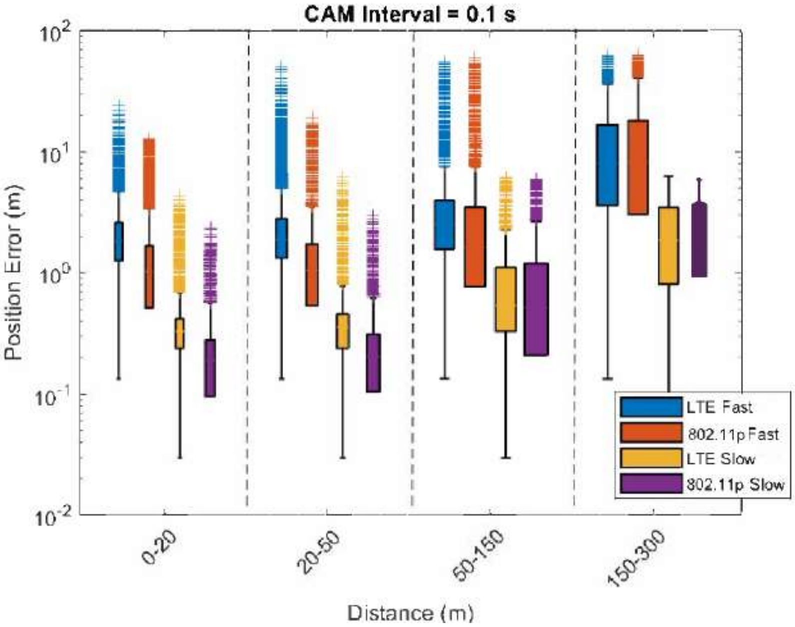


Figure 57: **Position Error for LTE-PC5 Mode-4 and 802.11p in Fast and Slow Highway scenarios.**

Figure 58 displays an interesting interaction between the NAR metric and the CAM interval. Using a longer CAM interval reduces congestion, and hence results in a higher PRR. However, having a fixed memory of 2 seconds, the NAR metric has less chances to recover from a packet loss if a longer CAM interval is used. The previous trade-off results in LTE-PC5 Mode-4 having the best NAR performance with a CAM interval of 0.1 seconds, followed closely by the 0.2 seconds one. However, at distances larger than 120m, 802.11p suffers from severe levels of loses when using CAM intervals of 0.1 seconds and this results in a much better NAR performance with a CAM interval of 0.2 seconds for these distances, although being in both cases inferior to LTE-PC5 Mode-4. Using a CAM interval of 1 second results in both cases in a very low NAR, despite being the setting that delivered the highest PRR, since only 2 CAMs are sent before erasing an entry in the LDM. These results illustrate the importance of optimizing MAC settings looking at application layer performance, and not only at MAC metrics. Comparing the NAR results with the PRR ones depicted in Figure 3, we can see that for CAM intervals of 0.1 and 0.2 seconds, the gain of LTE-PC5 Mode-4 over 802.11p is higher when looking at NAR than when looking at PRR, i.e.

$$\boxed{\frac{NAR_{LTE-PC5}}{NAR_{802.11P}} > \frac{PRR_{LTE-PC5}}{PRR_{802.11P}}}$$

which demonstrates how the application layer design can amplify the benefits of a more robust MAC layer.

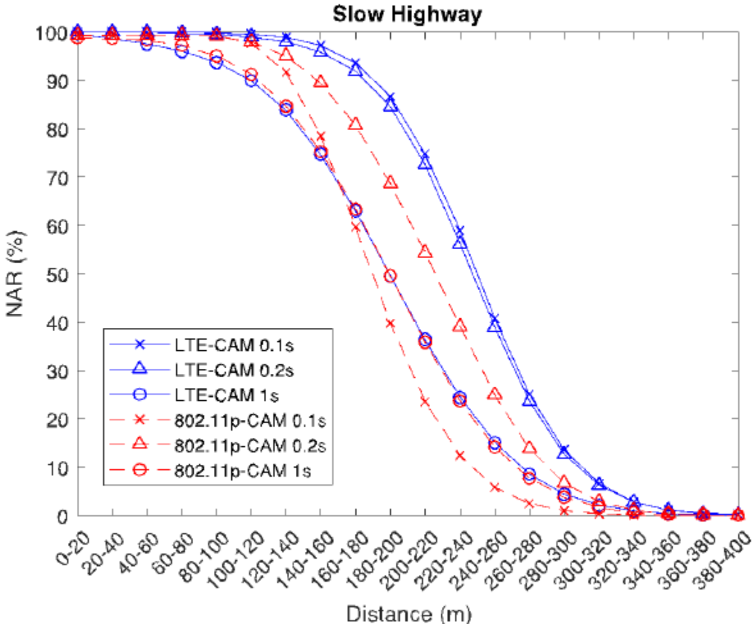


Figure 58: **NAR for LTE-PC5 Mode-4 and 802.11p with varying CAM interval.**

Finally, Figure 59 depicts using boxplots the obtained Position Error in this experiment. Again, we see that at small distances, 802.11p results in slightly lower errors than LTE-PC5 Mode-4, due to its shorter channel access delay. As expected, Position Error increases for both technologies with the CAM interval for neighboring vehicles below 50 meters, but this dependence becomes less clear as distance increases, since far away vehicles are subject to higher losses. A CAM interval of 0.1 seconds is the setting delivering the smallest Position Errors for both technologies, but in the case of 802.11p a CAM interval of 0.2 seconds becomes slightly better for far away vehicles, due to the high losses, because of congestion, when a CAM interval of 0.1 seconds is used. A worst-case Position Error of 6 meters is obtained in this experiment, since vehicles move at 3 m/s and an LDM memory of 2 seconds is used. These worst-case errors though may still be very dangerous when considering vehicles in proximity (0-20 meters bucket).

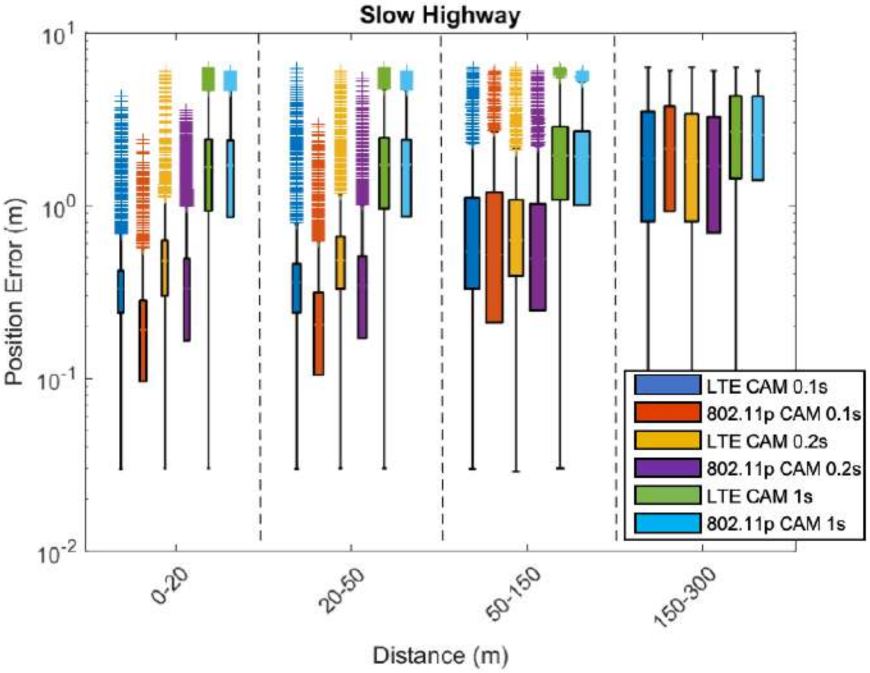


Figure 59: **Position Error for LTE-PC5 Mode-4 and 802.11p with varying CAM interval.**

Conclusions

There is a current debate among regulators about the appropriate technology to support future ITS safety services in the 5.9 GHz band. Among these services, Cooperative Awareness, which delivers the position of surrounding vehicles in real time, is of special interest to build the first safety applications such as anticipated collision avoidance, or cooperative manoeuvring.

The two commercially ready candidate technologies to support ITS safety services are 802.11p, promoted by the IEEE, and LTE-PC5 Mode-4, promoted by the 3GPP. Although 802.11p enjoys a well-established ecosystem, it is based on an older PHY layer technology. On the other hand, LTE-PC5 Mode-4 has a superior PHY performance but a not so mature ecosystem. Nevertheless, IEEE is currently developing the standard 802.11bd whose backwards compatible enhancements are expected to bring the PHY layer performance on-par with LTE-PC5. It is therefore of interest to understand, once PHY layers will perform similar, the impact that the MAC layer mechanisms of both technologies have on ITS safety services like cooperative awareness.

This analysis provides an in-depth simulative study addressing the previous question, while benchmarking the Cooperative Awareness service with two KPIs: NAR and Position Error. Our study concludes that, even with an equivalent PHY performance, the MAC layer of LTE-PC5 Mode-4 will outperform the MAC layer of 802.11p (or its enhanced version 802.11bd) when road congestion increases, which translates into an improved range to reliably track surrounding vehicles. However, 802.11p/bd results in slightly better vehicle positioning accuracy at lower distances, although for both technologies additional work is needed to limit worst case vehicle positioning accuracy, which can be as high as 11 meters when moving at 100 kmph for vehicles at 0-20 meters distance.

3. Conclusions

D6.2 provided a detailed report of the design, implementation, demonstration and evaluation scheme literature and industrial standards regarding the selected use-cases investigated in CPSoSAWARE. It also analyzes the dataset acquisition, usage, the structure, while it also reports the planning of the pilot studies. The evaluation trials use cases regard (1)Human-robot interaction in manufacturing environment and (2) Connected and autonomous L3-L4 vehicles along with its sub-usecases analyzed in 2.2.1, 2.2.2, 2.3.1. The objective of the evaluation/ demonstration scenarios for each use case is to study the potential and limitation of the CPSoSaware system components in dedicated environments to ensure the best system's performance and to provide informed experimental feedback for future enhancement of the technology.

4. References

[1] Lorenzo Vianello, Jean-Baptiste Mouret, Eloïse Dalin, Alexis Aubry, Serena Ivaldi. Human posture prediction during physical human-robot interaction. IEEE Robotics and Automation Letters, IEEE 2021, 6 (3), pp.6046-6053. 10.1109/LRA.2021.3086666. hal-03115242v2

[2] Marike K. van den Broek and Thomas B. Moeslund. 2020. Ergonomic Adaptation of Robotic Movements in Human-Robot Collaboration. In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI '20). Association for Computing Machinery, New York, NY, USA, 499–501. DOI:https://doi.org/10.1145/3371382.3378304

[3] Jan Stenum, Cristina Rossi, Ryan T. Roemmich, Two-dimensional video-based analysis of human gait using pose estimation, bioRxiv 2020.07.24.218776; doi: https://doi.org/10.1101/2020.07.24.218776

[4] S. Alliney, Digital filters as absolute norm regularizers, IEEE Transactions on Signal Processing,40 (1992), pp. 1548–1562.

[5] Recursive median filters of increasing order : a variational approach, IEEE Transactions on Signal Processing, 44 (1996), pp. 1346–1354.

[6] M. Nikolova, A variational approach to remove outliers and impulse noise, JMIV, 20 (2004), pp. 99–120.

[7] T.F. Chan and S. Esedoglu, Aspects of total variation regularized L1 function approximation, SIAM Journal on Applied Mathematics, 65 (2005), pp. 1817–1837.

[8] L. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation-based noise removal algorithms, Physica D, 60 (1992), pp. 259–268.

[9] W.K. Allard. “Total variation regularization for image denoising II: Examples”. In: SIAM Journal on Imaging Sciences 1.4 (2008), pp. 400–417.

[10] W.K. Allard. “Total variation regularization for image denoising I: Geometric theory”. In: SIAM Journal on Mathematical Analysis 39.4 (2007), pp. 1150–1190.

[11] F. Alter, V. Caselles, and A. Chambolle, A characterization of convex calibrable sets in Rn, Math. Ann., 7 (2005), pp. 29–53.

[12] G. Bellettini, V. Caselles, and M. Novaga, The total variation flow in rn, J. Differential Equations, 184 (2002), pp. 475–525.

[13]V. Caselles, A. Chambolle, and M. Novaga, Uniqueness of the cheeger set of a convex body. 2006.

[14]W. Yin, W. Goldfarb, and S. Osher, The total variation regularized L1 model for multiscale decomposition, SIAM Journal on Multiscale Modeling and Simulation, 6 (2007), pp. 190–211.

[15]J. Darbon, Total variation minimization with L1 data fidelity as a contrast invariant filter, 4th International Symposium on Image and Signal Processing and Analysis (ISPA 2005), (2005), pp. 221–226.

[16]P Perona and J Malik. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990.

[17]Jerome Fehrenbach and Jean-Marie Mirebeau. Sparse Non-negative Stencils for Anisotropic Diffusion. Journal of Mathematical Imaging and Vision, 2013.

[18] Gonzalez, R., & Woods, R. (2001). Digital Image Processing. New York: PrenticeHall.

[19] Barbu, T. (2013). Variational Image Denoising Approach with Diffusion Porous Media Flow. Abstract and Applied Analysis, Hindawi(SI17), 1-8

[20]https://en.wikipedia.org/wiki/USB_video_device_class

[21]https://velodynelidar.com/products/hdl-32e/

[22]https://www.applanix.com/products/poslv.htm

[23]https://www.spectra.de/produkte/industrie-pc-systeme/mini-pc-systeme/1-slot/spectra-powerbox-12c0.html?listtype=search&searchparam=Spectra-PowerBox%201280#aHR0cHM6Ly93d3cuc3BlY3RyYS5kZS9pbmRleC5waHA/bGFuZz0wJmNsPXNlYXJjaCZzZWFyY2hwYXJhbT1TcGVjdHJhLVBvd2VyQm94KzEyODA=

[24]https://www.dspace.com/de/gmb/home/products/hw/micautob/microautobox2.cfm

[25] Felzenszwalb, P., & Huttenlocher, D. (2012, September 2). Distance Transforms of Sampled Functions. Theory of computing, pp. 415-428.

[26]Li, W., Cao, H., Liao, J., Xia, J., Cao, L., & Knoll, A. (2020). Parking Slot Detection on Around View Images Using DCNN. Frontiers in Neurorobotics, 14, 46.

[27] Li, W., Cao, L., Yan, L., Li, C., Feng, X., & Zhao, P. (2020). Vacant Parking Slot Detection in the Around View Image Based on Deep Learning. Sensors, 20(7), 2138.

[28] Tanimoto, T. (1958). An Elementary Mathematical theory of Classification and Prediction. Internal IBM Technical Report.

[29] Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT Press, Cambridge.

[30] Tsardoulias, E., Protopapas, M., Symeonidis, A., & Petrou, L. (2019). A Comparative Analysis of Pattern Matching Techniques Towards. Journal of Intelligent & Robotic Systems, 98(3-4), 733–758

[28].F. Mendoza, L.M. Lopez, D. Camps-Mur, J. Casademont. "Benchmarking the Cooperative Awareness Service at Application Layer with IEEE 802.11p and LTE-PC5 Mode-4". IEEE International Mediterranean Conference on Communications and Networking. 7-10 September 2021. Athens (Greece).

[29]5GAA. "V2X Functional and Performance Test Report; Test Procedures and Results". 2018. Available online, accessed on January 2021: https://5gaa.org/wp-content/uploads/2018/11/5GAA_P-190033_V2X-Functional-and-Performance-Test-Report_final-1.pdf

[30] G. Naik, B. Choudhury, J.M. Park. "IEEE 802.11 bd & 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications". IEEE Access, vol.7, pp:70169-70184, 2019.

[31]L.M.W. Lopez, C.F.U. Mendoza, J. Casademont, D. Camps-Mur. "Understanding the impact of the PC5 resource grid design on the capacity and efficiency of LTE-V2X in vehicular networks". Wireless Communications and Mobile Computing, vol.2020, Article ID 8156908, 2020.

[32]A. Bazzi, G. Cecchini, A. Zanella, B.M. Masini. "Study of the Impact of PHY and MAC Parameters in 3GPP C-V2V Mode 4". IEEE Access, vol.6, pp:71685-71698, 2018.

[33]ETSI. EN 302 637-2 V1.4.1 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. April 2019.

[34]A. Autolitano, C. Campolo, A. Molinaro, R.M. Scopigno, A. Vesco. "An insight into Decentralized Congestion Control techniques for VANETs from ETSI TS 102 687 V1. 1.1". IFIP Wireless Days (WD), 2013.

[35]J. Song, Y. Wu, Z. Xu, X. Lin. "Research on car-following model based on SUMO". International Conference on Advanced Infocomm Technology, 2014.

[36]M. Boban, P.M. d'Orey. "Exploring the practical limits of cooperative awareness in vehicular communications". IEEE Transactions on Vehicular Technology, 65(6), pp:3904-3916, 2016.