Development and Demonstrations of Cooperative Perception for Connected and Automated Vehicles

Mao Shan, Stewart Worrall and Eduardo Nebot
The University of Sydney
{mao.shan, stewart.worrall, eduardo.nebot}@sydney.edu.au
September 3, 2021

This is a report that satisfies Milestone 13 of iMOVE CRC Project 1-006: Cooperative Perception
Contents

1 Introduction 3

2 Related Work 6

3 Development of Intelligent Platforms for Cooperative Perception 9
   3.1 Intelligent Roadside Unit: Prototype A ........................................ 9
   3.2 Intelligent Roadside Unit: Prototype B ........................................ 11
   3.3 Connected and Automated Vehicles ............................................. 14

4 Development of Cooperative Perception Framework 16
   4.1 Overview .................................................................................. 16
   4.2 Handling of ETSI CPMs ............................................................. 19
   4.3 Coordinate Transformation of Perceived Objects with Uncertainty .... 20
      4.3.1 Problem Formulation ............................................................ 21
      4.3.2 Numerical Simulation ........................................................... 23
   4.4 Probabilistic Cross-Platform Data Fusion ....................................... 27
      4.4.1 GMPHD Filter .................................................................. 27
      4.4.2 Enhancing Pedestrian Tracking with Person Re-Identification .... 29
      4.4.3 Track-To-Track Fusion ......................................................... 31

5 Demonstrations of CAV Operation Using IRSU Information 36
   5.1 Results in An Urban Traffic Environment ...................................... 36
   5.2 Results in CARLA Simulator ...................................................... 40
   5.3 Results in A Lab Environment .................................................... 43

6 Demonstrations of CAV Operation Using Cooperative Perception Information 47
   6.1 Results in CARLA Simulator ...................................................... 47
   6.2 Results in A Lab Environment .................................................... 55
   6.3 Results in USYD Campus ........................................................... 59
      6.3.1 Demonstration Setup ............................................................. 59
      6.3.2 Scenario A ........................................................................ 65
      6.3.3 Scenario B ........................................................................ 67
      6.3.4 Scenario C ........................................................................ 69
      6.3.5 Remarks ............................................................................ 71

7 Conclusions and Future Work 72
Abstract

Cooperative perception, or collective perception (CP) is an emerging and promising technology for intelligent transportation systems (ITS). It enables an ITS station (ITS-S) to share its local perception information with others by means of vehicle-to-X (V2X) communication, thereby achieving improved efficiency and safety in road transportation. In this report, we summarise our recent work on the development of a CP framework and ITS-S prototypes, including connected and automated vehicles (CAVs) and intelligent roadside units (IRSUs). We present a collection of experiments to demonstrate the use of CP service to improve awareness of vulnerable road users (VRU) and thus safety for CAVs in various traffic scenarios. We also demonstrate how CAVs can autonomously and safely interact with walking and running pedestrians, relying only on the CP information received from other ITS-Ss through V2X communication. This is one of the first demonstrations of urban vehicle automation using only CP information. We also address in the report the handling of collective perception messages (CPMs) received from multiple ITS-Ss, and passing them through a pipeline of CP information coordinate transformation with uncertainty, probabilistic cross-platform data fusion, multiple road user tracking, and eventually path planning/decision making within the CAV. The experimental results were obtained in simulated and real-world traffic environments using manually driven CV, fully autonomous CAV, and IRSU platforms retrofitted with vision and laser sensors and a road user tracking system.
Chapter 1

Introduction

Autonomous vehicles (AVs) have received extensive attention in recent years as a rapidly emerging and disruptive technology to improve safety and efficiency of current road transportation systems. Most of the existing and under development AVs rely on local sensors, such as cameras and lidars, to perceive the environment and interact with other road users. Despite significant advances in sensor technology in recent years, the perception capability of these local sensors is ultimately bounded in range and field of view (FOV) due to their physical constraints. Besides, occluding objects in urban traffic environments such as buildings, trees, and other road users impose challenges in perception. There are also robustness related concerns, for instance, sensor degradation in adverse weather conditions, sensor interference, hardware malfunction and failure. Unfortunately, failing to maintain sufficient awareness of other road users, vulnerable road users (VRU) in particular, can cause catastrophic safety consequences for AVs.

In recent years, V2X communication has garnered increasing popularity among researchers in the field of intelligent transportation system (ITS) and with automobile manufacturers, as it enables a vehicle to share essential information with other road users in a V2X network. This can be a game changer for both human operated and autonomous vehicles, which would be referred to as connected vehicles (CVs) and connected and automated vehicles (CAVs), respectively. It will also open many doors to new possibilities with peer-to-peer connectivity. The connected agents within the cooperative ITS (C-ITS) network will be able to exploit the significant benefits that come from sharing information amongst the network. For instance, the standardised cooperative awareness messages (CAMs) enable mutual awareness between connected agents. Nevertheless, there are other types of road users such as non-connected vehicles, pedestrians, and cyclists that have not been included in the C-ITS services yet. The detection of these non-connected road users in this case becomes an important task for road safety.

The major standardisation organisations such as European Telecommunications Standard Institute (ETSI), SAE and IEEE have made a significant effort to standardise specifications regarding C-ITS services, V2X communication protocols, and security. This is essential to facilitate the deployment of C-ITS in road transportation network globally. The collective perception (CP) service is among those C-ITS services that are currently being standardised by ETSI. The CP service enables an ITS station (ITS-S), for instance, a CAV or an intelligent roadside unit (IRSU) to share its perception information with adjacent ITS-Ss by exchanging Collective Perception Messages (CPMs) via V2X communication. The ETSI CPMs convey abstract representations of perceived objects instead of raw sensory data, facilitating the interoperability between ITS-Ss of different types and from different manufactures. A CAV can benefit from the CP service in terms of improved awareness of surrounding road users, which is essential for ensuring road safety. Specifically, it facilitates a CAV to extend its sensing range and improve
sensing quality, redundancy, and robustness through cross-platform data fusion, i.e., fusing its local sensory data with other CAVs and IRSUs information. Figure 1.1 illustrates two simple CP scenarios where a CV/CAV can receive essential perception information through the CP service when approaching a visibility-limited intersection. Besides, the improved perception quality as a result of the data fusion potentially relaxes the accuracy and reliability requirements of onboard sensors. This could lower per vehicle cost to facilitate the massive deployment of CAV technology. As for traditional vehicles, CP also brings an attractive advantage of enabling perception capability without retrofitting the vehicle with perception sensors and the associated processing unit.

Over the last three years, we have worked closely with Cohda Wireless on CP and CAV. We are particularly interested in the safety implications the CP service is bringing into the current and future transportation network, and how the CP service will potentially shape the development of intelligent vehicles. To this end, we have developed IRSU prototypes and tested them with the Australian Centre for Field Robotics (ACFR) CAV platforms in a number of simulation and real-world experiments representing different traffic scenarios. This report presents a comprehensive summary of the CP related technology we have developed and the findings from experiments conducted using these real platforms. Essentially, the report showcases how a CAV achieves improved safety and robustness when perceiving and interacting with VRU using the CP information from other ITS-Ss, including intelligent infrastructure and other CVs with sensing capability, in different traffic environments and with different setups.

The report first presents the developed CAV and IRSU platforms in Chapter 3. This is followed by proposing in Chapter 4 a set of key approaches that enable a CV/CAV to consider the
CP information in its operation. These include 1) the coordinate transformation of perception information considering the respective uncertainties, and 2) a probabilistic cross-platform data fusion approach that can consider both types of information present in a CP system, i.e., local sensor observations and remote tracks received from other ITS-Ss via V2X communication.

The vehicle platforms used in the experiments are equipped with a suite of local perception sensors to implement full autonomy. Nevertheless, the CAV employed in the experiments does not use the internal perception capabilities in its automated operation so as to highlight the benefits of using the CP service in the traffic environments. The received perception data (in the form of ETSI CPMs) from other ITS-Ss is used as the only or main source of information for multiple road user tracking and path planning within the CAV.

The experiments conducted and presented in Chapter 5 and Chapter 6 have different levels of complexity in their setups. More specifically, the set of experiments in Chapter 5 focuses on the operation of the CAV/CV using the perceived information from an IRSU. The first experiment was conducted on a public road in an urban traffic environment and the CV was able to “see” a visually obstructed pedestrian before making a turn to an alleyway. It is demonstrated in the next two experiments the CAV navigated autonomously and safely when interacting with walking and running pedestrians in proximity in a simulated and real lab traffic environments, respectively.

In the set of experiments presented in Chapter 6, the CAV received the perception information from another CV or from both a CV and an IRSU through V2V or V2X communication, respectively, and considered it in its automated operation. In these experiments, a probabilistic cross-platform data fusion approach was employed for tracking pedestrians and vehicles, and two different fusion strategies were investigated. It is also demonstrated that the CAV considered the fused perception information in its path planning and decision making when interacting with other road users at a pedestrian crossing and at a T-junction in the real-world environments.

The remainder of the report is organised as follows. Chapter 2 will focus on the related work on CP and its use cases for CAV. Chapter 3 presents the IRSU and CAV platforms developed and used in the experiments. The development of the CP approaches is addressed in Chapter 4. The results from simulation and real-world experiments are presented in Chapter 5 and Chapter 6, followed by conclusions drawn in Chapter 7.

Some of the research outcome presented in this report has been published in [1].
Chapter 2

Related Work

The concept of CP has been extensively studied in the ITS research community over the last two decades. Initial CP related work proposes to share raw sensory data between two mobile agents, such as images [2], lidar point clouds [3], both combined [4–6], location and relative range measurements [7,8]. Those approaches however tend to require prohibitively high bandwidth for existing V2X communication technologies in a dense environment. Besides, raw sensor data is often vendor dependent and proprietary, causing interoperability issues among communicating ITS-Ss.

More theoretical and experimental work on CP was conducted as part of the Ko-FAS [9] research initiative. These include [10–12] based on the Ko-PER specified Cooperative Perception Message (CPM). It is a supplementary message to the standard ETSI ITS G5 CAMs, to support the abstract description of perceived dynamic and static objects. Experimental studies are conducted in [10] on the Ko-PER CPM transmission latency and range. In [11], a high level object fusion framework in CP is proposed, which combines the local sensor information with the perception data received from other V2X enabled vehicles or roadside units (RSUs). Reference [12] investigates the inter-vehicle data association and fusion for CP. More recent work in [13] proposes a variant of Ko-PER CPM and analyses the trade-off between message size as a result of enabling optional data fields in the CPM and global fusion accuracy.

Based on the work in [10], reference [14] proposes Environmental Perception Message (EPM) for CP with different information containers specifying sensor characteristics, originating station state, and parameters of perceived objects. It is also addressed in [14] high level object fusion using the perceived information in received EPMs. Both EPM and the earlier Ko-PER CPM are a separate message that contains all CP related data elements (DEs) and data frames (DFs), and has to be transmitted in parallel with an ETSI CAM. There is also work towards extended CAM. For instance, Cooperative Sensing Message (CSM) from AutoNet2030 [15–17] extends CAM specifications to include description of objects perceived by local or remote sensors. Following a similar concept of CP, Proxy CAM is presented in [18–20], where intelligent infrastructure generates standard CAMs for those perceived road users, while the work in [21] proposes a CPM comprised of a collection of CAMs, each describing a perceived object. The work in [22] and [23] evaluates different EPM dissemination variants under low and high traffic densities and proposes to attach the CP relevant DFs in EPM to CAM to minimise communication overhead. The CPM currently being specified at ETSI, as in [24], is derived from optimising the EPM and combining it with CAM. It is therefore more self-contained, no longer dependent on the reception of CAMs.

Similarly, there are early stage standardisation activities in SAE advanced application technical committee to standardise messages and protocols for sensor data sharing in SAE J3224 [25]. These messages and protocols are not yet defined and are thus not considered in this work.
Considering the limited communication bandwidth and avoiding congestion in the wireless channel, more recent studies in the CP area tend to focus on the communication aspect of the technology, weighing up provided CP service quality and the V2X network resources. The work in [26] investigates ETSI CPM generation rules that balance provided service quality and the V2X channel load. Reference [27] provides an in-depth study on the impact of different CPM generation rules from the perspectives of V2X communication performance and perception capabilities in low and high density traffic scenarios. The authors of [28] raise the concern of redundant data sharing in vehicle-to-vehicle (V2V) based CP with the increase of CAV penetration rate. To tackle the redundant transmission issue, a probabilistic data selection approach is presented in [29]. Reference [30] proposes an adaptive CPM generation rule considering change in perceived object’s state, and the authors of [31] propose to employ object filtering schemes in CPM, to improve communication performance while minimising the detriment to perception quality. Similarly, the work in [32] presents a deep reinforcement learning based approach that a vehicle can employ when selecting data to transmit in CP to alleviate communication congestion.

There is also work conducted to explore the use cases, benefits, and challenges of CP. Reference [33] provides early study of CP illustrating its potential in terms of improved vehicle awareness and extended perception range and field of view. The work presented in [14] evaluates EPM for obstacle avoidance of two manually driven CAVs, showing that the CP helps gain extra reaction time for the vehicles to avoid obstacles. Reference [28] analyses the performance gain in extending horizon of CAVs by leveraging V2V based CP. The work in [34] and [35] analytically evaluates the enhancement of environmental perception for CVs at different CP service penetration rates and with different traffic densities. The authors of [36] discuss the security threats in CP and propose possible countermeasures in V2X network protocols, while the work in [37] focuses on using CP for detecting vehicle misbehaviour due to adversarial attacks in V2X communication. Most of CP related use cases studied in the literature are safety related, including cooperative driving [6,17], cooperative advisory warnings [38,39], cooperative collision avoidance [5,14,40], intersection assistance [18,41], and vehicle misbehaviour detection [42], to name a few. It is presented in [43] quantitative comparison of V2V and vehicle-to-infrastructure (V2I) connectivity on improving sensing redundancy and collaborative sensing coverage for CAV applications. The work concludes that infrastructure support is crucial for safety related services such as CP, especially when the penetration rate of sensing vehicles is low. The authors of [18] demonstrate improved awareness of approaching vehicles at an intersection using the CP information from an IRSU. The CP in the work is achieved through Proxy CAM. Reference [44] compares CAM and CPM and demonstrates IRSU assisted augmented perception through simulations. Recent work in [45] demonstrates the IRSU assisted CP for extending perception of CAVs on open roads. Infrastructure-assisted CP is also part of the scope of Managing Automated Vehicles Enhances Network (MAVEN) [46], an EU funded project targeting traffic management solutions where CAVs are guided at signalised cooperative intersections in urban traffic environments [41]. Other CP related joint research projects include TransAID [47] and IMAGinE [48].

A significant proportion of the existing work conducts the analyses of V2X communication and CP in simulated environments. For instance, the work in [28,29] is carried out in an open source microscopic road traffic simulation package SUMO (Simulation of Urban Mobility) [49]. Another commonly used network and mobility simulator is Veins (Vehicles in Network Simulation) [50,51], which integrates SUMO and a discrete-event simulator OMNeT++ (Open Modular Network Testbed in C++) [52] for modelling realistic communication patterns. The authors of [22,23,26,31,33,53] conduct work in Artery [54,55] framework, which wraps SUMO and OMNet++, and enables V2X simulations based on ETSI ITS G5 protocol stack. There
are also simulators for advanced driving assistance systems (ADASs) and autonomous driving systems, which provide more realistic sensory level perception information. In recent years, they start to show their potential in testing and validating CP with CAVs. For instance, Pro-SiVIC is employed in [44] for CP related simulations, and CARLA [56,57] is combined with SUMO in a simulation platform developed for CP in [32].
Chapter 3

Development of Intelligent Platforms for Cooperative Perception

3.1 Intelligent Roadside Unit: Prototype A

The first IRSU prototype developed comprises of a sensor head, a processing workstation, and a Cohda Wireless MK5 RSU. This is the prototype A that is used in demonstrations presented in Chapter 5, Sections 6.1, and 6.2. The sensor head is mounted on a tripod for easy deployment in different testing environments, as shown in Figure 3.1. Specifically, the two Pointgrey Blackfly BFLY-PGE-23S6C-C cameras are mounted with an angle separation of 45°. The camera with its lens Fujinon CF12.5HA-1 has a horizontal field-of-view (FOV) of about 54°, and a vertical FOV of 42°. The setup of dual cameras achieves a combined FOV of approximately 100°, and the FOV can be further augmented by adding more cameras to the sensor head. A 16-beam lidar is also installed to the sensor head. The workstation has AMD Ryzen Threadripper 2950X CPU, 32GB memory, RTX2080Ti GPU, running Robot Operation System (ROS) Melodic on Ubuntu 18.04.2 long-term support (LTS).

Figure 3.1: The developed IRSU prototype A is equipped with multiple sensors including dual cameras and a 16-beam lidar. The sensors sit on a tripod for easy deployment in the field.
In terms of information processing, the workstation first processes the sensory data of images and lidar point clouds for pedestrian and vehicle detection. Specifically, the raw images from cameras are first rectified using camera intrinsic calibration parameters. As illustrated in Figure 3.2, the road users within the images are classified/detected using YOLOv3 [58] that runs on GPU. At the same time the lidar point clouds are projected to the image coordinate system with proper extrinsic sensor calibration parameters. The lidar points are then segmented, clustered and labelled by fusing the visual classifier results (in the form of bounding boxes in the image) and the projected lidar points.

The detection results are then encoded into ETSI CPMs and broadcast by Cohda MK5 at 10 Hz. Details are available in Section 4.2. We tested the working range of the developed IRSU for detecting common road users, such as pedestrians and vehicles. The maximum detection range is approximately 20 m for pedestrians and 40 m for cars. Also in the IRSU, a variant of Gaussian mixture probability hypothesis density (GMPHD) filter [59] is employed to track multiple road users and has its tracking results visualised in real time within the workstation. The same tracking algorithm is also employed on the CAV side. Details would be given in Section 3.3.

To assess the position tracking accuracy of the IRSU, an outdoor test was conducted at the Shepherd Street car park at the University of Sydney (USYD). Figure 3.3 illustrates the setup of the test at the car park, where a pedestrian was walking in front of the IRSU for approximately one minute. The ground truth positions of the target pedestrian were obtained at 1 Hz by an u-blox C94-M8P RTK receiver, which reports a standard deviation of 1.4 cm of positioning when in the RTK fixed mode.

As presented in Figure 3.4, the trajectory of the target reported by the local tracker is found close to the ground truth points. The root mean squared error (RMSE) in position is calculated by comparing the ground truth positions and the corresponding estimates from the tracker. As the two sources of positions were obtained at different rates (10 Hz from the tracker versus 1 Hz from RTK), each RTK reading is compared with the position estimate that has the nearest timestamp. It can be seen from Figure 3.4c that the distance of the target pedestrian to the IRSU varies from 5 to 22 metres, and the position RMSE values remain less than 0.4 m throughout the test.
Figure 3.3: Pedestrian tracking setup for the IRSU. (a) shows the detection of the target pedestrian within a camera image using YOLOv3. The lidar points are projected to the image frame with extrinsic calibration parameters. The projected points in (b) are colour coded based on their range to the sensor in 3D space. Those bold points indicate those hitting the ground plane. (c) demonstrates the tracking of the pedestrian in 3D space. The RTK antenna was hidden within the cap of the pedestrian to log GNSS positions as the ground truth.

Figure 3.4: Pedestrian tracking results for the IRSU. It can be seen in (a) that when the target pedestrian was walking in a figure eight pattern in front of the IRSU the tracked positions are found to be well aligned with the ground truth path obtained from a RTK receiver.

3.2 Intelligent Roadside Unit: Prototype B

A new IRSU, i.e., prototype B, was built and used for the CP system demonstration in the USYD campus, as presented in Section 6.3. The prototype B was built using Stereolabs ZED 2 stereo camera as the main perception sensor, NVIDIA Jetson Xavier as the data processing...
Figure 3.5: The new IRSU developed for the demonstration in Section 6.3. (a) shows the Jetson Xavier, an LCD monitor and other accessories housed in a protective case. (b) illustrates the full setup of the battery powered IRSU in the demonstration area. The ZED 2 camera and the MK5 are mounted on a tripod.

Figure 3.6: A portable RTK surveyor developed for calibrating the IRSU.

As shown in Figure 3.5a, the Jetson Xavier, an LCD monitor, and other accessories are housed inside a Pelican protective case. The ZED 2 camera and the MK5 RSU are installed on a tripod, as Figure 3.5b reveals. The entire setup can be battery powered. The full-load power is approximately 75 W. In terms of software, the ZED SDK provides a real-time deep network based 3D road user detection approach. This replaces the camera-lidar fusion based visual detector in the previous IRSU prototype, while other components in the developed cooperative perception software package, such as the coordinate transformation algorithm and the GMPHD based multi-target tracker, remain the same. As the road user detection and tracking in the new IRSU is running above 20 FPS, the results are downsampled to 10 Hz when converted to CPMs for communication.

As identified in real-world demonstrations, the accurate pose for the IRSU with respect to the world frame is considered critical information in cooperative perception. Thus, a portable RTK surveyor was developed as a calibration tool for the IRSU, to survey its global location and heading when deployed in a real-world urban environment. The portable RTK surveyor is based on a Raspberry Pi and u-blox ZED-F9P GNSS module. As Figure 3.6 illustrates, it has an LCD touchscreen showing information including current GNSS and UTM coordinates. The surveyor requires Internet access to obtain NTRIP correction data from commercial RTK service providers. It provides localisation with centimeter level accuracy when a fix is obtained.
Figure 3.7: The IRSU calibration process using the portable RTK surveyor.

in RTK mode. The surveyor also has data logging function.

Figure 3.7 explains how the IRSU calibration works. It requires the portable RTK surveyor and a mobile phone hotspot providing the Internet access. The calibration process mainly consists of two steps. The first step is to survey the global position of the IRSU, which is straightforward using the portable RTK surveyor. The second step is to calibrate the global heading of the IRSU perception sensors. As Figure 3.7a shows, the portable RTK surveyor can share its real-time RTK position with the IRSU at 10 Hz through the WLAN hosted by the mobile phone hotspot. On the IRSU side, the real-time position of the portable RTK surveyor will show up in RVIZ as a white vertical line. In the meantime, a person is needed in this process to walk in front of the IRSU sensor, holding the portable RTK surveyor with the GNSS antenna on the top of their heads, as Figure 3.7b illustrates. We can then use the ROS dynamic reconfigure GUI to obtain the yaw parameter of the IRSU with respect to the world frame. As Figure 3.7c reveals, a good calibration result is indicated when this white vertical line lines up with the person detection result (as a magenta pillar in the figure).
3.3 Connected and Automated Vehicles

Figure 3.8 presents an overview of hardware configuration of the CAV platform built by the ACFR ITS group. Images are captured onboard at 30 FPS by an NVIDIA Drive PX2 automotive computer with six gigabit multimedia serial link (GMSL) cameras with 1080p resolution and a 100° horizontal FOV each. They are arranged to cover a 360° horizontal FOV around the vehicle. One of the vehicles also has a 32-beam scanning lidar with 30° vertical FOV and 360° horizontal FOV for scanning the surrounding at 10 Hz, and the other vehicle is equipped with a 16-beam scanning lidar. Both the cameras and the lidar have been calibrated to the local coordinate system of the platform, using the automatic extrinsic calibration toolkit presented in [60]. Besides, the CAV platform has a GNSS receiver, a 6 degrees of freedom IMU, and four wheel encoders for odometry and localisation. The onboard Intel next unit of computing (NUC) has 32 GB of memory and a quad-core Intel i7-6670HQ processor, serving as the main processing computer within the CAV. The NUC is running ROS Melodic on Ubuntu 18.04.2 LTS. Last but not least, the CAV platform has been retrofitted with Cohda Wireless MK5 OBU to enable the V2X communication capability. Please refer to [61] for more details on the CAV platform and the USYD Campus data set collected using the platform.

The CAV did not use any of the retrofitted perception sensors for road user detection in all experiments but the one in Section 6.1. The multiple cameras were used for video recording purpose, and in some of the experiments the multibeam lidar was enabled only for aiding vehicle self-localisation within the map. Lidar feature maps of the experiment sites were built using a simultaneous localisation and mapping (SLAM) algorithm. The maps are based on pole and building corner features extracted from lidar point clouds, which are essential for localisation since GNSS cannot provide the desired level of accuracy in the experiment environments. Interested reader can refer to [62] for more information. In the meantime, a Lanelet2 map is built for every experiment site, which includes road network, lane layout and traffic rules such as speed limits, traffic lights and right-of-way rules.
Figure 3.9: Two examples in (a) and (b) showing the detection and tracking of multiple pedestrians walking in front of the CV using its onboard perception sensors. Note that while the point clouds from two mounted lidars are visualised simultaneously in the figure, only the data from the 16-beam VLP-16 is used in the perception stack.

For experiments in Sections 6.2 and 6.3, the same perception stack that is running in the IRSU prototype A had been ported to the CV to turn it into a perception car, using its onboard Velodyne VLP-16 lidar and the front GMSL camera. The real-time visual detection task is carried out by the YOLOv4 Tiny running on the onboard NVIDIA Drive PX2. Two examples of tracking multiple walking pedestrians in front of the CV can be found in Figure 3.9.
Chapter 4

Development of Cooperative Perception Framework

4.1 Overview

A CP framework has been developed and tested with three real ITS-Ss, i.e., an IRSU and two CAVs, as described in Chapter 3. Its hardware structure is illustrated in Figure 4.1. The CP system supports sharing and fusion of local perception data between two CAVs through V2V communication, on top of the V2I communication between the IRSU and the CAVs. The detected and tracked road user descriptions are encoded to ETSI CPMs and transmitted from the IRSU to the CAVs and between two CAVs through the Cohda MK5s at 10 Hz. Optionally, one of the CAVs can be temporarily set up as an IRSU so as to test in the other CAV the fusion of CP information received from the two IRSUs. It should be noted that with trivial extension work the developed CP framework can support more ITS-Ss.

Figure 4.1: The hardware structure of the developed CP framework. The IRSU broadcasts perceived objects information in the form of ETSI CPMs through Cohda Wireless MK5 RSU. The CAVs have V2X communication capability through Cohda Wireless MK5 OBU.
As previously mentioned in Chapter 1, the experiments presented in Chapter 5 employ a simplified version of the developed CP framework, which involves only a CAV and an IRSU. Its overall system diagram is illustrated in Figure 4.2. The experiment in Section 6.1 investigates the CP between two CAVs through V2V communication, and finally, the experiments in Sections 6.2 and 6.3 use the full configuration of the developed CP framework.

When a CAV receives an ETSI CPM through the onboard Cohda Wireless MK5 OBU, the received perceived objects information is first decoded from binary ASN.1 encoding, and transformed with its uncertainty to the local frame of reference of the CAV, as presented in Section 4.3, which also takes into account the estimated egocentric pose of the CAV in self-localisation.

Following the coordinate transformation with uncertainty, the perceived objects information from other ITS-Ss are fused into a multiple road user tracking algorithm that is a variant of GMPHD filter running within the local frame of the receiving CAV. The tracked states of road users include their position, heading, and speed. The general formulation of the GMPHD tracker is to use Gaussian mixture to represent the joint distribution of the group of tracked targets. The GMPHD approach is considered attractive due to its inherent convenience in handling track initiation, track termination, probabilistic data association, and clutter. Compared with the naive GMPHD algorithm, the road user tracker running in both the IRSU and CAV is improved with measurement-driven initiation of new tracks and track identity management. Also note that an instance of the tracker is required for each type of road users, which effectively reduces the overall computational cost. Please refer to Section 4.4.1 for details on the GMPHD filter.

Figure 4.3 illustrates the pipeline of how perception information received from multiple other ITS-Ss (i.e., the CPS providers in the figure) are processed before it is fused into the road user tracker together with the local perception data within the CAV. While Figure 4.3 only presents a general concept of fusing the two types of information, the way to incorporate the transformed tracks from other ITS-Ss in the cross-platform data fusion (refer to Section 4.4 for details) is not as simple as fusing local measurements from onboard sensors. Essentially, it requires further attention to handling the common prior information from a platform and across different platforms. The simplest example is when a node A passes a piece of information to a node B, which then sends the information back to the node A. If A were to fuse the “new” information from B with the “old” one it has under the assumption of independence, the covariance matrix, which indicates the uncertainty level of the tracked target, would be reduced while it should
The problem is more sophisticated in a V2X network scenario, where the tracks of perceived objects shared amongst ITS-Ss are not only dependent on previous estimates, but also become cross-correlated as they are typically exchanged amongst different platforms in the V2X network. To tackle the problem of double counting of common prior information, which leads to inconsistency in estimation, the developed CP framework employs the covariance intersection (CI) algorithm for fusing tracks information from two or more sources without knowing the cross-correlation information between them. Details on the CI and track-to-track fusion can be found in Section 4.4.3.

While the CI algorithm is introduced, the GMPHD tracker is still required to fuse independent observations from local sensors. For this reason, the GMPHD filter has been improved to be able to fuse both types of information, i.e., measurements from local sources and tracks it receives from remote sources. In general, the GMPHD filter based cross-platform data fusion scheme employed in the developed CP framework is summarised in Figure 4.4.

The navigation subsystem in the CAV is responsible for path planning, monitoring and controlling the motion of the vehicle from the current position to the goal. A hybrid A* path planner presented in our recent work [63] is running within the CAV to plan a path navigating around obstacles and the tracked pedestrians. It maintains a moving grid-based local cost map that considers 1) structural constraints, such as road and lane boundaries, present in the Lanelet2 map, 2) obstacles picked up by local perception sensors, and 3) current and future estimates of road users detected and broadcast by other ITS-Ss through V2X communication, such that it can plan a smooth, safe, and kinematically feasible path to avoid collision with any other road users the CAV becomes aware of. In the experiments presented in the report, the use of local perception information in the navigation was minimised or disabled.
4.2 Handling of ETSI CPMs

Each ETSI CPM consists of an ITS PDU header and five types of information containers accommodating mandatory and optional DEs and DFs. These containers are:

1. A **CPM Management Container**, which indicates station type, such as a vehicle or an IRSU, and reference position of the transmitting ITS-S.

2. An optional **Station Data Container**, which provides additional information about the originating station. This includes the heading, speed, and dimensions when the originating station is a vehicle.

3. Optional **Sensor Information Containers**, which describe the type and specifications of the equipped sensors of the transmitting ITS-S, including sensor IDs, types, and detection areas.

4. Optional **Perceived Object Containers**, each of which describes the dynamics and properties of a perceived object, such as type, position, speed, heading, and dimensions. These perceived object descriptions are registered in the coordinate system of the originating station.

5. Optional **Free Space Addendum Containers**, which describe different confidence levels for certain areas within the sensor detection areas.

The CPM generation rules support different abstraction levels for perceived object descriptions for the implementation flexibility, which can derive from low-level detections made by individual sensors, or the results of the high-level data fusion. For the V2I case, the road user detections produced through the camera-lidar fusion in the developed IRSU are considered in the encoding of CPMs, as depicted in Figure 4.2. For the V2V scenario, the vehicles are exchanging CPMs that represent tracked road users as a result of the cross-platform data fusion.

The overall perception information flow from one ITS-S to another is also illustrated in Figure 4.1. Specifically, the handling of the perception information within the CP system is mostly carried out in ROS realm. The ROS Kinetic is installed in both Cohda MK5s, with the workstation/NUC set as the ROS master to the transmitting/receiving MK5, respectively. The
perceived road users on the workstation side are first described in the form of a ROS CPM, which is used internally within the ROS system and brings convenience of message handling and diagnostics in ROS. Each ROS CPM is a self-contained message that contains all information required for an ETSI CPM. It is the ETSI CPM that is eventually transmitted in the V2X communication.

A ROS `msg_bridge` node running within each MK5 serves as a bridge between ROS CPMs and ETSI CPMs. It populates an ETSI CPM given information from a ROS CPM and based on the ASN.1 definition of the ETSI CPM. In the meantime, it decodes an ETSI CPM received from other ITS-Ss back into the ROS CPM format. The transmission and reception of ETSI CPM payload is handled by Cohda’s V2X stack in MK5. We keep upgrading the ROS `msg_bridge` node to support the features required in the CP demonstrations in accordance with the status quo of ETSI CPM standardisation.

4.3 Coordinate Transformation of Perceived Objects with Uncertainty

The perceived objects information broadcast through CP service is produced from the perspective of the sensing ITS-S. When a piece of perceived object information is received, it is not usable to the receiving agent until it is transformed into its local coordinate system. In [14], the coordinate transformation of perceived objects does not explicitly incorporate the accuracy fields accompanied with DEs in the EPMs. We stress that uncertainty bounds associated with the perception information are indispensable in successive data fusion and thus have to be considered. Besides, it is essential for the coordinate transformation to also incorporate the uncertainty contained in the pose estimation of both ITS-Ss. These are discussed in detail as follows.

1. Perception uncertainty in the sensing ITS-S. Every commonly used perception sensor in ITS area has its own strengths and limitation. For instance, RGB images are useful in detecting object instances and classification of road users with an estimated confidence level using a visual classifier algorithm. Doppler RADARs produce both position and velocity measurements but are prone to noise and interference from the environment. Lidars have a high range resolution to observe physical extent and shape of objects. Nevertheless, the point density decreases dramatically along with range. These sensors all produce measurements corrupted by noise, and thus should be modelled with uncertainty. Combining multi-modal sensory information not only improves robustness of the perception system, but also increases accuracy, which means a lower level of uncertainty. The produced estimates of perceived objects with uncertainty have been represented in `Perceived Object Container` of the CPM specification.

2. Self-localisation uncertainty of sensing and receiving ITS-Ss. A vehicle ITS-S has to constantly localise itself within a global frame of reference such as map or UTM frame for navigation and safety reasons. However, accurate self-localisation for a moving platform is known to be one of the existing challenges in ITS applications such as urban navigation. Using GNSS as the only source for localisation often cannot achieve satisfactory accuracy, in particular in GNSS-degraded or even GNSS-denied environments. There are various existing solutions that can provide higher localisation accuracy such as RTK, GNSS and inertial/encoder data fusion, feature based localisation based on existing map, etc. Nevertheless, even with the same localisation approach, the uncertainty magnitude in the localisation can vary significantly depending on certain external conditions, such as GNSS
satellite visibility in the sky, and the quantity of observable features and their qualities in the surroundings. The location of an IRSU, although deployed static in a traffic environment, is not immune from localisation error either when set either by GNSS or through a surveying process. The localisation uncertainties of both sensing and receiving ITS-Ss therefore have to be considered in the perceived objects coordinate transformation since it cannot be completely eliminated regardless of the self-localisation means employed. The originating ITS-S information including its pose with associated uncertainty has been contained within the CPM Management Container and Station Data Container in the CPM definition.

Consequently, the transformed object state estimates contain uncertainty that is mainly a combined result of uncertainty in the localisation of both ITS-Ss and that in the sensory perception. As illustrated in Figure 4.5, both sensing and receiving ITS-Ss contain uncertainties in their estimated egocentric states within the global frame $G$. The sensing ITS-S $S$ observes multiple road users within its local frame with some level of sensory measurement uncertainty labelled with purple dashed lines. The road users measurement uncertainties grow when transformed to a global frame $G$ due to self-localisation uncertainty of ITS-S $S$, as labelled with gray dotted lines. The road users measurement uncertainties further grow, as labelled by orange solid lines, when transformed to the local frame of the receiving ITS-S $R$ as its self-localisation uncertainty is also incorporated.

Both the work in [11] and [64] performs the coordinate transformation of perceived object states with their uncertainty through temporal and spatial alignment. The temporal alignment is introduced in the work mainly due to the time difference between the reception of a CAM and a Ko-PER CPM. It is however less of a concern for an ETSI CPM, which now can contain the originating station information required from a CAM.

### 4.3.1 Problem Formulation

This section presents mathematical formulation of the coordinate transformation of perceived objects information a recipient ITS-S receives from a sensing ITS-S. We propose to use unscented transform (UT) in the coordinate transformation with uncertainty, which here is recognised as a non-linear and non-deterministic process. The formulation presented is intended for 2D transformation, as it is reasonable to assume a planar road surface for the road user detection and tracking problem discussed in the report. Nevertheless, the formulation can be easily extended to transformation in 3D space. It is important to note the formulation presented here is intended
to provide a generic form for two arbitrary ITS-Ss in a V2X network, which includes both V2V and V2I scenarios. Before proceeding further, some definitions about coordinate systems are first given to facilitate discussions.

The global frame, which is the fixed coordinate system attached on the ground, is represented as \( \{G\} = \{\vec{x}_G, \vec{y}_G\} \). Local frame of the sensing ITS-S \( S \) is attached on the platform body and is represented as \( \{S\} = \{\vec{x}_S, \vec{y}_S\} \), with \( \vec{x}_S \) pointing to the east direction if it is an IRSU and pointing towards its moving direction for a vehicle. Local frame of the recipient ITS-S \( R \) is attached on the platform body and moving with the platform. It is denoted as \( \{R\} = \{\vec{x}_R, \vec{y}_R\} \) with \( \vec{x}_R \) pointing towards its moving direction.

The pre-superscript is employed to describe the coordinate frame in which the corresponding variable is expressed. For instance, \( \bar{v}_{x_t} \) denotes the position of an object \( p \) in \( x \) direction with respect to \( \{V\} \) at time \( t \). A variable without pre-superscript is defined in a generic way to not specify a particular coordinate frame.

The state of the receiving ITS-S \( R \) at time \( t \) and its estimate are denoted as
\[
G_{x_t}^R = \begin{bmatrix} G_{x_t}^R & G_{y_t}^R & G_{\theta_t}^R \end{bmatrix}^T ,
\]
where \( G_{x_t}^R, G_{y_t}^R, \) and \( G_{\theta_t}^R \) denote 2D coordinates and the heading, respectively, in \( \{G\} \) at time \( t \). The state estimate is represented by a multivariate Gaussian as
\[
G_{x_t}^R \sim \mathcal{N}(G_{\bar{x}_t}^R, \Sigma_{\bar{x}_t}^R) ,
\]
where \( G_{\bar{x}_t}^R \) and \( \Sigma_{\bar{x}_t}^R \) denote the mean vector and covariance matrix, respectively.

Likewise, the estimated pose of the perceiving ITS-S \( S \) at time \( t \) is too represented as a Gaussian variable:
\[
G_{x_t}^S = \begin{bmatrix} G_{x_t}^S & G_{y_t}^S & G_{\theta_t}^S \end{bmatrix}^T \sim \mathcal{N}(G_{\bar{x}_t}^S, \Sigma_{\bar{x}_t}^S) ,
\]
where \( G_{x_t}^S \) and \( G_{y_t}^S \) are the location of \( S \), \( G_{\theta_t}^S \) is its heading with respect to \( \{G\} \) at time \( t \), \( G_{\bar{x}_t}^S \) and \( \Sigma_{\bar{x}_t}^S \) denote the mean vector and covariance matrix, respectively. In the special case of an IRSU serving as the perceiving ITS-S, the subscription \( t \) in (4.3) can be dropped as \( G_{x_t}^S \) and \( G_{y_t}^S \) are time invariant and \( G_{\theta_t}^S \) is towards due east for an IRSU. Nevertheless, \( \Sigma_{\bar{x}_t}^S \) is still applicable to an IRSU due to the presence of uncertainty in the surveying process.

The state vector of a perceived object \( p \) at time \( t \) is represented from the sensing ITS-S’s perspective as
\[
S_{x_t}^p = \begin{bmatrix} S_{x_t}^p & S_{y_t}^p & S_{\theta_t}^p \end{bmatrix}^T \sim \mathcal{N}(S_{\bar{x}_t}^p, \Sigma_{\bar{x}_t}^p) ,
\]
where \( S_{x_t}^p \) and \( S_{y_t}^p \) are object coordinates and \( S_{\theta_t}^p \) is heading within the sensing ITS-S’s local frame at time \( t \).

We obtain the transformed state of perceived object \( p \) with respect to \( \{R\} \) by
\[
R_{x_t}^p = \text{trans}(G_{x_t}^R, G_{x_t}^S, S_{x_t}^p) .
\]

Specifically, in \( \text{trans}(\cdot) \), we have
\[
\begin{bmatrix}
R_{x_t}^p \\
R_{y_t}^p \\
1
\end{bmatrix} = T(G_{x_t}^R)^{-1} T(G_{x_t}^S) \begin{bmatrix}
S_{x_t}^p \\
S_{y_t}^p \\
1
\end{bmatrix} ,
\]
where
\[
T(\begin{bmatrix} x & y & \theta \end{bmatrix}^T) = \begin{bmatrix}
\cos \theta & -\sin \theta & x \\
\sin \theta & \cos \theta & y \\
0 & 0 & 1
\end{bmatrix}
\]
is the homogeneous transformation matrix.
Given a Gaussian representation of perceived object state pdf in the form of mean vector $\bar{x}$ and covariance matrix $\Sigma$ with respect to $\{S\}$, the nonlinear transformation to receiving ITS-S’s frame $\{R\}$ with uncertainty is achieved through the UT process.

An augmented state vector is constructed by concatenating the state vector of the receiving and sensing ITS-Ss and that of the perceived object $p$ before transformation. Its Gaussian estimate is written as

$$x_a^t \sim \mathcal{N}(\bar{x}_a^t, \Sigma_a^t), \quad (4.7)$$

where $\bar{x}_a^t = \left[ (G\bar{x}_R)^T \ (G\bar{x}_S)^T \ (S\bar{x}_p)^T \right]^T$, and $\Sigma_a^t = \text{blkdiag}\{G\Sigma_R, G\Sigma_S, S\Sigma_p\}$.

A collection of sigma points $\{X_i, w^m_i, w^c_i\}_{i=0}^{2d}$ are obtained based on the augmented state estimate prior to the transformation:

$$X_0 = \bar{x}_a^t$$

$$X_i = \bar{x}_a^t + \left( \sqrt{(d + \lambda) \Sigma_i^a} \right)_i \quad \text{for } i = 1, \cdots, d$$

$$X_i = \bar{x}_a^t - \left( \sqrt{(d + \lambda) \Sigma_i^a} \right)_i \quad \text{for } i = d + 1, \cdots, 2d$$

$$w^m_0 = \frac{\lambda}{d + \lambda}$$

$$w^c_0 = \frac{\lambda}{d + \lambda} + (1 - \alpha^2 + \beta)$$

$$w^m_i = w^c_i = \frac{1}{2 (d + \lambda)} \quad \text{for } i = 1, \cdots, 2d \quad (4.8)$$

where $\lambda = \alpha^2 (d + \kappa) - d$, $d = \text{dim} (x)$ is the dimension of state $x$, scaling parameters $\kappa \geq 0$, $\alpha \in (0, 1]$, and $\beta = 2$ is optimal for Gaussian distributions, $(\sqrt{\Sigma_i^a})_i$ is to obtain the $i$th column of the matrix square root $R = \sqrt{\Sigma_i^a}$, which can be computed by Cholesky decomposition such that we have $\Sigma_i^a = RR^T$. Note that each sigma point can be decomposed in accordance with the concatenation sequence in (4.7), i.e., $X_i = \left[ (X_i^R)^T \ (X_i^S)^T \ (X_i^p)^T \right]$.

This is followed by passing each sigma point through the frame transformation function $\text{trans} (\cdot)$ in (4.5), which yields a set of transformed sigma points. For $i = 0, 1, \cdots, 2d$,

$$Y_i = \text{trans} (X_i^R, X_i^S, X_i^p) \quad (4.9)$$

Lastly, the transformed state of perceived object $p$ in $\{R\}$ is recovered by

$$R\bar{x}_p^t = \sum_{i=0}^{2d} w^m_i Y_i$$

$$R\Sigma_p^t = \sum_{i=0}^{2d} w^c_i (Y_i - R\bar{x}_i^t) (Y_i - R\bar{x}_i^t)^T \quad (4.10)$$

### 4.3.2 Numerical Simulation

The coordinate transformation of perceived objects in ETSI CPMs is validated through numerical simulation. As illustrated in Figure 4.6, the simulation is setup with two ITS-Ss acting as
a publishing/sensing ITS-S and a receiving ITS-S of CPMs. The sensing ITS-S is static at a position of \((100m, 100m)\) on the map, facing the east direction. It is perceiving road users within its local frame with some level of uncertainty in its sensory measurements. The receiving ITS-S in the simulation is a CAV moving at \(2m/s\) from its initial position at \((0m, 75m)\) with the same heading of the sensing ITS-S, which can be either an IRSU (V2I case) or another CAV (V2V case). Both V2V and V2I scenarios are considered in the simulation for a comparison. Both ITS-Ss are assumed to contain uncertainty in their self-localisation.

Within the sensing ITS-S, the perceived objects information and egocentric pose estimate are encoded into CPMs in the form of binary payloads and published at 10 Hz. The receiving CAV, as it moves on the map, decodes the CPMs received and transforms the perceived objects into its local coordinate system. Through coordinate transformation, the uncertainty in the transformed perceived information is a combined result of the sensing uncertainty and the self-positioning uncertainties of both ITS-Ss.

<table>
<thead>
<tr>
<th>Table 4.1: Simulation Parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard Deviation in State Estimate</strong></td>
</tr>
<tr>
<td>Position</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Perceived Objects</td>
</tr>
<tr>
<td>Sensing ITS-S</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Receiving CAV</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Figure 4.8: Confidence ellipses of perceived objects transformed to the local frame of the receiving CAV in Test 1, where different heading estimate standard deviations are tested for CAV(s). As the receiving CAV moves, (a), (c), and (e) show the results at different relative positions between IRSU and the receiving CAV, while (b), (d), and (f) present the results for the V2V case. Each ellipse represents 95% of position estimate confidence. The first ellipse on the left represents the uncertainty in the position estimate of the sensing ITS after transformed into receiving CAV’s frame.

In the simulation, the covariance matrix in the self-localisation estimate of the receiving CAV is denoted as $\Sigma_i^R = \text{blkdiag} \left\{ (\sigma_{pos})^2, (\sigma_{pos})^2, (\sigma_\theta)^2 \right\}$, where $\sigma_{pos}$ and $\sigma_\theta$ denote the standard deviation of position and heading estimate, respectively. The simulation consists of two tests for evaluating the effect of different $\sigma_\theta$ and $\sigma_{pos}$ values in coordinate transformation. Specifically, $\sigma_\theta$ is varied in Test 1 with $\sigma_{pos}$ kept constant, while a range of $\sigma_{pos}$ are tested for the CAV in Test 2. Each of the tests further includes both V2V and V2I cases. In the V2V case, the sensing CAV is setup with the same covariance matrix as for the receiving CAV, i.e., $\Sigma_i^S = \Sigma_i^R$. In the V2I case, where the sensing ITS-S is an IRSU, the localisation noise is assumed small yet non-zero. For simplicity, a group of 20 static road users are positioned in front of the sensing ITS-S in a line with identical perception uncertainty parameters, as demonstrated in Figure 4.7. Detailed parameters adopted in the simulation can be found in Table 4.1. Please note that standard deviations as low as 0.05° and 0.005m are tested in the simulation as they are roughly the minimum values supported in CPMs for representing the uncertainty of heading and
Figure 4.9: Confidence ellipses of perceived objects transformed to the local frame of the receiving CAV in Test 2, with different standard deviations in position estimate of CAV(s). (a), (c), and (e) demonstrate the results for the V2I case, while (b), (d), and (f) illustrate the results for the V2V case at different relative positions. Each ellipse represents 95% of position estimate confidence. The first ellipse on the left denotes the uncertainty in the transformed position estimate of the sensing ITS.

position estimates, respectively, for the originating ITS-S. This is mainly due to discretisation of confidence levels of the corresponding DEs defined in CPMs. Likewise, 0.005m is adopted as the standard deviation for the position estimate of the IRSU, and the heading estimate standard deviation is set to a close-to-zero value $\epsilon$ for preventing numerical errors in the transformation.

Figure 4.8 reveals the result of Test 1, i.e., the effect of different uncertainty levels in heading estimates of the CAV(s), along with the movement of the receiving CAV. Figure 4.9 depicts the result of Test 2, which is with different uncertainty in position estimates of the CAV(s). It can be seen from both figures that the confidence ellipses of those perceived objects, after transformed into the local coordinate system of the receiving CAV, are bloated and distorted to different extents, depending on their relative poses with respect to both ITS-Ss. Please note that in both figures, the first yellow ellipse on the left represents the uncertainty in the transformed position estimate of the sensing ITS.

In Figure 4.8, the bloating effect is found sensitive to the uncertainty contained in the heading estimate of both ITS-Ss. Also, the bloated ellipses are slanted along the tangential directions...
of both ITS-Ss, and the bloating is found more serious for ellipses that are further from both ITS-Ss. In theory, these thin and long confidence ellipses caused by the heading uncertainty are expected to be banana shaped due to the non-linear nature of the coordinate transformation. The result therefore indicates that with a larger uncertainty level in heading estimate and for further perceived objects, the Gaussian assumption starts to show its limitation for representing the perceived object estimates. One can choose to use Gaussian mixture or non-Gaussian representations in the transformation to alleviate the issue. As a comparison, the bloating effect as a result of the uncertainty in position estimates of the ITS-Ss is found less correlated to the relative distance, and the bloating happens in both x and y directions, as shown in Figure 4.9. This shows that reducing the estimate uncertainty of heading in ITS-S self-localisation is more effective than that of position, in suppressing the bloating of uncertainty during the coordinate transformation of perceived objects information.

Also, the two scenarios of an IRSU and a CAV acting as the sensing ITS-S are compared in each of Figure 4.8 and Figure 4.9. When the sensing ITS-S is an IRSU, it can be seen that the transformed confidence ellipses are still slanted but in general less bloated due to very low uncertainty in IRSU’s position.

4.4 Probabilistic Cross-Platform Data Fusion

4.4.1 GMPHD Filter

Please note that the perceived objects information is fed to the filtering algorithm as measurements of tracked targets. Therefore, the notation of perceived object state $x$ previously used in Section 4.3 is changed to $z$, and $x$ is used to denote the state of tracked targets in this section, following the convention for a filtering algorithm.

Before we proceed further, some background knowledge regarding Random Finite Set (RFS) formulation for multi-target filtering should be briefly explained. Please note that an intensity defined in RFS represents the hypotheses of the state of the entire set of targets. We consider the assumptions that 1) each target moves independently and is observed by sensors independently of one another, 2) the clutter follows Poisson distribution and is independent of measurements originating from targets, 3) there is no spawning of existing targets for simplicity.

Let $v_{t|t-1}(\cdot)$ and $v_t(\cdot)$ represent the intensities associated with the multi-target predicted and posterior probability density functions, respectively. It can be formulated using Bayes’ theorem such that the prediction and posterior intensities can be inferred recursively over time as

$$v_{t|t-1}(x) = \int p_{S,t}(\zeta) f_{t|t-1}(x|\zeta) v_{t-1}(\zeta) d\zeta$$

(4.11)

and

$$v_t(x) = (1 - p_{D,t}(x)) v_{t|t-1}(x)$$

$$+ \sum_{z \in Z_t} p_{D,t}(x) \psi_t(z|x) v_{t|t-1}(x)$$

$$+ \gamma_t(x).$$

(4.12)

In (4.11) and (4.12), $Z_t$ is the set of multi-target state measurements at time $t$; $f_{t|t-1}(\cdot|\zeta)$ is the kinematic model for a target state $\zeta$ evolving from time $t-1$ to time $t$; $\psi_t(z|x)$ is the likelihood function of state $x$ given an observation $z$; $\gamma_t(\cdot)$ is the intensity of the spontaneous
birth RFS at time $t$; $p_{S,t}(\zeta)$ is the survival probability that a target survives at time $t$ given its previous state $x$; $p_{D,t}(x)$ is the detection probability of detecting a target at time $t$ given its state $x$; and lastly, $\kappa_t(\cdot)$ is the intensity of clutter RFS at time $t$.

As a Gaussian mixture implementation of RFS for the multi-target tracking problem, the GMPHD filter represents each intensity by a mixture of Gaussian variables associated with weights. These include the birth intensity $\gamma_t(x)$, prior $v_{t-1}(x)$, prediction $v_{t|t-1}(x)$ and posterior $v_t(x)$.

The Gaussian mixture representation of the prior intensity is given in the form of

$$v_{t-1}(x) = \sum_{i=1}^{J_{t-1}} w_{i,t-1}^i N(\bar{x}^{i}_{t-1}, \Sigma^{i}_{t-1}),$$

where $J_{t-1}$ is the total number of Gaussian components at time $t-1$; $\bar{x}^{i}_{t-1}, \Sigma^{i}_{t-1}$, and $w_{i,t-1}$ refer to the mean and variance of the $i^{th}$ Gaussian component and its associated weight, respectively, for the prior states of targets.

Then the prediction intensity is denoted in the form of Gaussian mixture as

$$v_{t|t-1}(x) = \sum_{i=1}^{J_{t-1}} w_{i,t|t-1}^i N(\bar{x}^{i}_{t|t-1}, \Sigma^{i}_{t|t-1}),$$

where $w_{i,t|t-1} = p_{S,t}(\bar{x}^{i}_{t-1}) w_{i,t-1}$; $N(\bar{x}^{i}_{t|t-1}, \Sigma^{i}_{t|t-1})$ is the predicted estimate for the $i^{th}$ Gaussian through the kinematic model $f_{t|t-1}(\cdot|\cdot)$.

The Gaussian mixture representation for the birth intensity is written as

$$\gamma_t(x) = \sum_{i=1}^{J_{\gamma,t}} w_{\gamma,t}^i N(\bar{x}^{i}_{\gamma,t}, \Sigma^{i}_{\gamma,t}),$$

where $J_{\gamma,t}$ is the total number of new targets generated at time $t$; $w_{\gamma,t}^i$ is the initial weight assigned to the Gaussian representing the $i^{th}$ new target; $\bar{x}^{i}_{\gamma,t}$ and $\Sigma^{i}_{\gamma,t}$ refer to initial Gaussian mean and variance for the new target. In the GMPHD implementation for the project, $N(\bar{x}^{i}_{\gamma,t}, \Sigma^{i}_{\gamma,t})$ and $J_{\gamma,t}$ are set based on the measurements represented by Gaussian variables and their total number at time $t$, respectively. Also, a unique track ID can be assigned to each new target.

Lastly, the posterior intensity is also represented by a Gaussian mixture obtained by

$$v_t(x) = (1 - p_{D,t}(x)) v_{t|t-1}(x) + \sum_{z \in \{Z_t^l \cup Z_t^r\}} v_{D,t}(x; z) + \gamma_t(x)$$

where $Z_t^l$ is the set of object perception data from local sensors at time $t$; $Z_t^r$ is the set of transformed cooperative perception information from remote sources at time $t$; $v_{D,t}(x; z) = \sum_{i=1}^{J_{t-1}} w_{i,t}^i N(\bar{x}^{i}_{t}, \Sigma^{i}_{t}(z)); N(\bar{x}^{i}_{t}, \Sigma^{i}_{t}(z))$ is the posterior estimate for the $i^{th}$ Gaussian component updated with measurement $z$ through the likelihood function $\psi_t(\cdot|\cdot)$; $w_{i}^i(z) = \frac{p_{D,t}(x)w_{i,t-1}^i q_t^i(z)}{\kappa_t(x) + \sum_{i=1}^{J_{t-1}} p_{D,t}(x)w_{i,t-1}^i q_t^i(z)}$ is the normalised weight associated with the $i^{th}$ Gaussian after update; $q_t^i(z)$ is the data association likelihood of $z$ with the $i^{th}$ Gaussian component. After the measurement update, the number of Gaussian components representing $v_t(x)$ is denoted as $J_t$. 28
The computation of $N(\bar{x}_i^t(z), \Sigma_t^i(z))$ and $q_t^i(z)$ differs depending on the type of $z$. For $z \in Z^l_t$, a traditional detection-to-track fusion paradigm is adopted, for instance, Kalman filter and its variants, as in [59]. In the case where $z \in Z^r_t$ is a track from a remote source, track-to-track fusion is required and the details are given in Section 4.4.3.

It can be seen in the formulation from (4.13) to (4.16) that the value of $J_t$ follows a combinatorial growth over time. There are a few solutions one can adopt to effectively control the growth of the Gaussian components:

1. A pruning process can be introduced by removing Gaussian components with weights lower than a threshold.

2. A merging process can be used for combining close Gaussian components based on some distance metrics, for instance, Mahalanobis distance.

3. A maximum number of Gaussian components at each iteration can be set.

The total number of targets at time $t$ is estimated as $\hat{N}_t = \sum_{i=1}^{J_t} w_t^i$, which sums the weights of all Gaussian components in the $v_t(x)$.

Lastly, the extraction process of the mult-target state estimates is by sorting the Gaussian components by their weights and shortlisting the $\hat{N}_t$ highest weight Gaussians.

### 4.4.2 Enhancing Pedestrian Tracking with Person Re-Identification

![Image](image1.png)

Figure 4.10: An example of image labelled with bounding boxes from YOLOv3 visual classifier. Each bounding box represents an instance of detected pedestrian.

![Image](image2.png)

Figure 4.11: Image patches from bounding boxes in the Figure 4.10 are encoded to visual appearance descriptors using the CNN based visual feature extractor.
Road user detection using a popular visual classifier such as YOLOv3, as shown in Figure 4.10, usually does not provide more specific description about the detected object itself besides the object class it belongs to. It is therefore beneficial for a multitarget tracker to consider the object’s visual appearance as an additional clue, in particular, when to associate targets appearing in different image frames or to different platforms.

The visual appearance feature extractor from DeepSORT [65] has been added to the road user detection and tracking approaches developed for the project. The feature extractor is practically a convolutional neural network (CNN) based visual classifier that has the final classification layer stripped. The CNN is pretrained with MARS (Motion Analysis and Re-identification Set) Dataset for person re-identification.

Following the road user detection, the image patches within each bounding box are extracted and passed through the visual appearance feature extractor to obtain appearance descriptors, as Figure 4.11 illustrates. Given two descriptors A and B, the following simple cosine similarity function [66] is employed as a metric to quantify the visual similarity of two image patches, which correspond to two road users.
Cosine Similarity = \frac{A \cdot B}{\|A\| \|B\|} \quad (4.17)

It is found in tests that the pretrained feature extractor performs reasonably well in identifying the same person across two image frames. Two examples are presented in Figure 4.12 and Figure 4.13. To improve the tracking performance of pedestrians, the GMMPHLD filter has been updated with the Cosine Similarity of visual features, as formulated in (4.17), added to its score function for probabilistic data association.

4.4.3 Track-To-Track Fusion

A traditional Kalman based filter only works under the independent information assumption. However, cross-platform data fusion assuming independent information will lead to inconsistent estimate results. The use of CI algorithm can safely fuse estimates, in particular, in the case without the knowledge of cross-correlation between estimates to fuse. This is the strategy we employ for the track-to-track fusion in the proposed CP framework. This section mainly focuses on the track-to-track fusion, while the detection-to-track fusion has been well addressed previously in Section 4.4.1.

Covariance Intersection

Assume we would like to combine two Gaussian estimates \( \mathcal{N}(\alpha, A) \) and \( \mathcal{N}(\beta, B) \) without knowing their cross-correlation. In this work, we consider the CI formulation generalised for the case where the two estimates can possess different dimensions, as in [67], which means

\[
\dim(\beta) = \dim(H\alpha),
\]

where \( H \) is defined as an observation matrix.

The fused estimate \( \mathcal{N}(c, C) \), which will have the same dimension as \( \mathcal{N}(\alpha, A) \), is then computed using the CI algorithm as

\[
\mathcal{N}(c, C) = CI(\mathcal{N}(\alpha, A), \mathcal{N}(\beta, B)),
\]

where

\[
C^{-1} = \omega A^{-1} + (1 - \omega) H^T B^{-1} H
\]

\[
c = C(\omega A^{-1} \alpha + (1 - \omega) H^T B^{-1} \beta),
\]

and \( \omega \in [0, 1] \) is a scalar obtained by solving the optimisation problem of minimising the trace or determinant of \( C \). For the case of minimising the determinant, it can be written as

\[
\omega = \arg \min \omega \det(C).
\]

It is shown in Figure 4.14 a numerical example which fuses Gaussian estimates from multiple sources. It can be easily found out that the fused estimate produced by CI has a smaller covariance matrix than that of either estimate before fusion. Although not shown in the figure, the fused estimate remains the same when we continue to fuse the result with the third estimate recursively for 10 more times, and \( \omega = 1 \). This demonstrates one of attractive properties of CI algorithm, which is the immunity to double counting of information. When two input
The estimates considered are $N\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 9 & 5 & 7 \\ 5 & 9 & 4 \\ 7 & 4 & 9 \end{bmatrix}\right)$, $N\left(\begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 9 & 5 & 7 \\ 5 & 9 & 4 \\ 7 & 4 & 9 \end{bmatrix}\right)$, and $N\left(\begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 8 & -5 \\ -5 & 8 \end{bmatrix}\right)$. The observation matrix $H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. For the convenience of illustration, the three estimates are visualised in 2D as blue, red, and yellow solid ellipses, respectively, representing their 95% confidence ellipses. The CI first fuses the two estimates in blue and red and yields the fused estimate $N\left(\begin{bmatrix} 1.24 \\ 2.31 \\ 3.19 \end{bmatrix}, \begin{bmatrix} 6.30 & 1.62 & 4.87 \\ 1.62 & 6.30 & 1.36 \\ 4.87 & 1.36 & 8.25 \end{bmatrix}\right)$, which is shown in 2D as the purple dashed ellipse. $\omega = 0.791$ in the CI fusion. The fused estimate and the third estimate in yellow are then fused using CI to produce $N\left(\begin{bmatrix} 1.65 \\ 3.04 \\ 3.52 \end{bmatrix}, \begin{bmatrix} 4.10 & -1.62 & 3.12 \\ -1.62 & 4.10 & -1.17 \\ 3.12 & -1.17 & 8.95 \end{bmatrix}\right)$, which is visualised in 2D as the green dotted ellipse. $\omega = 0.683$ this time.

estimates are truly independent, the CI fused estimates are in general more conservative than Kalman filtered results. However, CI is the safer option when the correlation of information from different sources is unclear.

**Tracklet Fusion Management**

The remainder of the section will focus on the tracklet fusion management in the track-to-track fusion. Assume a CPM an ITS-S receives from another ITS-S contains $N$ remote tracks of road users, in the meantime, the host ITS-S maintains $M$ local tracks. It is assumed that the track-to-track fusion happens at time step $t$, therefore all the notations presented in the section has $t$ dropped for simplicity. To facilitate formulation and discussion, more specific notation should be introduced for the remote tracks $Z_r^t$ used in Section 4.4.1. The collection of remote tracks are now denoted as $\mathcal{T}_r = \{T_r^j\}_{j=1}^N$, where each $T_r$ is a tuple that can be decomposed to

$$T_r = \{\bar{x}_r, \Sigma_r, s_r, d_r\},$$

where $\bar{x}_r$ is track mean vector, $\Sigma_r$ is covariance matrix, $s_r$ refers to the unique ID of the transmitting ITS-S, and $d_r$ represents the track ID within the originating ITS-S. These elements are encoded into a CPM from the transmitter side.

Similarly, the set of local tracks are represented by $\mathcal{T}_l = \{T_l^i\}_{i=1}^M$, and each
\[ T_l = \{ \bar{x}_l, \Sigma_l, w, d_l, R \}, \] (4.23)

where \( w \) is the associated weight, \( d_l \) represents the track ID within the host ITS-S, and the alias ID list \( R = \{ [s_r^j, d_r^j] \}_{j=1}^P \) is introduced. Each pair in the list refers to the alias ID that the local track \( T_l \) is known as by another ITS-S, and the very same track can exist in \( P \) other ITS-Ss. Initially, the list is empty, i.e., \( P = 0 \). Details of the alias ID list will be given later. Unlike that in the remote track, the ID of the host ITS-S is not required in the local track. Please also note that in many scenarios the state vector of a remote track can be a subset of that of a local track maintained in the tracker. Hence, the CI formulation presented in (4.20) is used to accommodate the case of unmatched length of state vectors of the remote and local tracks. Like in (4.18), there exists an observation matrix \( H \) for the tracks, such that \( \dim(\bar{x}_r) = \dim(\bar{x}_l) \).

As shown in (4.22) and (4.23), a track not only contains state estimates, but is also assigned a track ID within the originating ITS-S, and different ITS-S have their own ways of initiating and managing track IDs. The same road user in the real world may be known by different track IDs within different ITS-Ss. When a remote track is received by an ITS-S, it would be impossible to determine if it matches up with a local track only by its track ID. To handle this issue, we introduce a simple yet effective track matching and fusion mechanism, which divides the track-to-track fusion into different subcases, which are detailed in the following.

Given each remote track \( T_r \in \mathcal{T}_R \), the mechanism first performs fusion only for its any matched local track \( T_l \in \mathcal{T}_L \) if it searches the alias ID list \( R \in \mathcal{T}_L \), finding that \( T_l \) has been fused with information from the particular transmitting ITS-S before, i.e., \( [s_r, d_r] \) exists in \( R \). For the remaining unmatched local tracks, they are fused only with those remote tracks received from the originating ITS-S for the first time. Specifically, it is determined based on the condition that \( s_r \in T_r \) is not found in \( R \in \mathcal{T}_L \). Once a local track is fused with a remote track, the pair of the remote track’s ID \( d_r \) and the transmitting ITS-S ID \( s_r \) is added into \( R \) for matching check at future time steps. It is also considered in the mechanism the update for the local tracks not detected by the transmitting ITS-S and the initiation of new local tracks based on the remote tracks.

Given a pair of \( T_l \) and \( T_r \) that are intended for fusion, the CI based track-to-track fusion is divided into two steps mathematically:

1. Solving the optimal \( \omega \in (0, 1) \) for the CI algorithm:

\[
\omega = \arg \min_{\omega} \det(\Sigma_{l\#}^\iota),
\] (4.24)

where

\[
(\Sigma_{l\#})^{-1} = \omega (\Sigma_l)^{-1} + (1 - \omega) H^T (\Sigma_r)^{-1} H.
\]

2. Estimate update:

\[
\begin{align*}
\bar{x}_{l+} &= \bar{x}_l + Ky \\
\Sigma_{l+} &= (I - K H) \Sigma_{l*} (I - K H)^T + K \Sigma_{r*} K^T \\
w_+ &= p_{D,r}(\bar{x}_l) q(T_l, T_r) w,
\end{align*}
\] (4.25)

where

\[
\Sigma_{l*} = \frac{\Sigma_l}{\omega} \quad \Sigma_{r*} = \frac{\Sigma_r}{1 - \omega} \quad y = \bar{x}_r - H \bar{x}_l
\]
\[ S = \Sigma_r + HH^T \quad \quad \quad \quad \quad K = \Sigma_s H^T S^{-1} \]

\[ q(T_l, T_r) = \mathcal{N}(y, S), \]

and lastly, \( p_{D,r}(\bar{x}_l) \) is the detection probability of the remote ITS-S detecting a local target given its state mean vector \( \bar{x}_l \).

The above (4.24) and (4.25) can be combined as a function denoted as \( \{ \bar{x}_l, \Sigma_l, w_+ \} = CI(T_l, T_r) \). The pseudocode for the track matching and fusion algorithm can be found in Algorithm 1. Finally, the set of updated local tracks, defined as \( T_{l+} \), is a union of different subsets of updated and new local tracks:

\[ T_{l+} = T_{ud} \cup T_{mt} \cup T_{um} \cup T_{nw}, \quad (4.26) \]

where

- \( T_{mt} \): The set of matched and fused local tracks, see Line 1 to 12 in Algorithm 1,
- \( T_{um} \): The set of unmatched local tracks after conditional fusion update, see Line 13 to 22 in Algorithm 1,
- \( T_{ud} \): The set of unmatched and updated local tracks that are not detected by the remote ITS-S, see Line 23 to 26 in Algorithm 1,
- \( T_{nw} \): The set of new local tracks initiated by the remote tracks, see Line 27 to 32 in Algorithm 1.
Algorithm 1: Cross-Platform Track-To-Track Fusion

Data: $\mathcal{T}_l, \mathcal{T}_r$

Result: $\mathcal{T}_{l+}, \mathcal{T}_{l+}^{mt}, \mathcal{T}_{l+}^{um}$

1: $\mathcal{T}_{l+}^{mt} \leftarrow \emptyset; \mathcal{T}_{l+}^{um} \leftarrow \mathcal{T}_l; \mathcal{T}_{r+}^{um} \leftarrow \mathcal{T}_r$

2: foreach $T_r \in \mathcal{T}_r$ do
3:     $\mathcal{T}_+ \leftarrow \emptyset$
4:     foreach $T_l \in \mathcal{T}_l$ do
5:         if $[s_r \ d_r] \in R$ then
6:             $T_{l+} \leftarrow T_l; \mathcal{T}_+ \leftarrow \mathcal{T}_+ \cup \{[s_r \ d_r]\}$
7:             $\{\bar{x}_{l+}, \Sigma_{l+}, w_+\} \leftarrow CI(T_l, T_r)$
8:             $\mathcal{T}_+ \leftarrow \mathcal{T}_+ \cup \{T_r\}$
9:             $\mathcal{T}_l^{um} \leftarrow \mathcal{T}_l^{um} \backslash \{T_l\}$
10:            $\mathcal{T}_r^{um} \leftarrow \mathcal{T}_r^{um} \backslash \{T_r\}$
11:     Normalise $\forall w_+ \in \mathcal{T}_+$
12:     $\mathcal{T}_{l+}^{mt} \leftarrow \mathcal{T}_{l+}^{mt} \cup \mathcal{T}_+$
13:     $\mathcal{T}_{l+}^{um} \leftarrow \emptyset$
14:     foreach $T_r \in \mathcal{T}_{l+}^{um}$ do
15:         $\mathcal{T}_+ \leftarrow \emptyset$
16:         foreach $T_l \in \mathcal{T}_{l+}^{um}$ do
17:             if $s_r \notin R$ then
18:                 $T_{l+} \leftarrow T_l; \mathcal{T}_+ \leftarrow \mathcal{T}_+ \cup \{[s_r \ d_r]\}$
19:                 $\{\bar{x}_{l+}, \Sigma_{l+}, w_+\} \leftarrow CI(T_l, T_r)$
20:                 $\mathcal{T}_+ \leftarrow \mathcal{T}_+ \cup \{T_r\}$
21:             Normalise $\forall w_+ \in \mathcal{T}_+$
22:             $\mathcal{T}_{l+}^{um} \leftarrow \mathcal{T}_{l+}^{um} \cup \mathcal{T}_+$
23:             $\mathcal{T}_{l+}^{ad} \leftarrow \emptyset$
24:     foreach $T_l \in \mathcal{T}_{l+}^{um}$ do
25:         $T_{l+} \leftarrow T_l; w_+ \leftarrow (1 - p_{D,r}(\bar{x}_l)) w$
26:         $\mathcal{T}_{l+}^{ad} \leftarrow \mathcal{T}_{l+}^{ad} \cup \{T_{l+}\}$
27:         $\mathcal{T}_{l+}^{nw} \leftarrow \emptyset$
28:     foreach $T_r \in \mathcal{T}_{l+}^{um}$ do
29:         $w_{nw} \leftarrow w_+; d_{l+}^{nw} \leftarrow$ New Track ID
30:         $\mathcal{R}_{l+}^{nw} \leftarrow \{[s_r \ d_r]\}$
31:         $T_{l+}^{nw} \leftarrow \{\bar{x}_r, \Sigma_r, w_{nw}, d_{l+}^{nw}, \mathcal{R}_{l+}^{nw}\}$
32:         $\mathcal{T}_{l+}^{nw} \leftarrow \mathcal{T}_{l+}^{nw} \cup \{T_{l+}^{nw}\}$
33:     $\mathcal{T}_{l+} \leftarrow \mathcal{T}_{l+}^{ad} \cup \mathcal{T}_{l+}^{mt} \cup \mathcal{T}_{l+}^{um} \cup \mathcal{T}_{l+}^{nw}$

Chapter 5

Demonstrations of CAV Operation Using IRSU Information

5.1 Results in An Urban Traffic Environment

The experiment was conducted in a real urban traffic environment next to the ACFR building located at Chippendale, Sydney. In the experiment, the IRSU was deployed near the intersection of Abercrombie St and Little Queen St, monitoring the traffic activity at the intersection and broadcasting the perception information in the form of ETSI CPMs in real time, as depicted in Figure 5.1. The intersection was chosen for the experiment mainly because turning from Abercrombie St to Little Queen St, which is a side road, requires extra attention from vehicle drivers due to the lack of traffic control and poor visibility of road users behind the corner building. The main purpose of the experiment is to demonstrate the improved VRU awareness and thus road safety for a vehicle ITS-S when it is able to learn pedestrian activity in blind spots through the CP information provided by IRSU.

Instead of autonomous driving, the CAV in the experiment was driven manually for ensuring public road safety in the real traffic scenario. The CAV platform therefore practically functioned as a CV in the experiment and is referred as the CV in the remainder of the section. While the CV was performing self-localisation within the experiment area, it was able to transform the perceived objects contained in CPMs into its local coordinate system with uncertainty and have the road users tracked in real time. The CV identifies itself and its driver based on the position of the transformed perceived objects in its local frame to avoid self-tracking.

In the experiment, as labelled in Figure 5.1, the CV first made a right turn from Meagher St to Abercrombie St, drove for about 60 m before turning to Little Queen St. As illustrated in Figure 5.2, the CV learnt the ongoing traffic activity at the intersection with the IRSU deployed far beyond the reach of its onboard perception sensors. This illustrates the extended sensing range for smart vehicles through the CP service. At a later time shown in Figure 5.3a and Figure 5.3b, when the CV was about to make a right turn to Little Queen St, the vehicle could “see” a visually occluded pedestrian behind a building with the information coming through the IRSU. Please note that it is achieved seconds before the pedestrian could actually be visually picked up from the CV perspective in Figure 5.3c and Figure 5.3d, which is crucial for safety in both manual driving and autonomous driving scenarios. This is considered another benefit of the CP service demonstrated in the experiment.

Besides, the CV localisation quality varied depending on the lidar feature quality at different locations of the experiment environment. It is illustrated by comparing Figure 5.3d and Figure 5.3f that the refinement in egocentric pose estimate as a result of a correction in the
Figure 5.1: Experiment setup at the intersection of Abercrombie St and Little Queen St. In (a), the IRSU was setup facing south of Abercrombie St. The car traffic flow was going north on this one-way street. The trajectory of the CV is denoted by a dashed red line, which indicates that the CV turned from Meagher St to Abercrombie St, followed by Little Queen St. In (b), CPMs were transmitted from the IRSU to the moving CV through Cohda MK5s.

self-localisation shown in the latter contributes to the improvement of tracking accuracy, which is consistent with the finding from the Section 4.3.2 that the level of ITS-S localisation estimate uncertainty correlates to the perceived objects detection uncertainty after coordinate transformation and thus tracking uncertainty.
Figure 5.2: The road user detection and tracking in the IRSU and CV in the Abercrombie St experiment. (a) and (b) show the image frames from Cam #1 and #2, respectively, overlaid with pedestrians and vehicles bounding boxes and projected lidar point cloud in the IRSU. The detection results from the dual cameras and the lidar were then fused within the IRSU for tracking in 3D space, as visualised in (c). The detection results were also transmitted to the CV, where they were transformed and used from the CV’s local frame. (d) illustrates the tracking results of the same group of road users within the CV at the same time. Due to change of visualisation colours in IRSU and CV, the vehicles in orange pillars in (d) correspond to red ones in (c), while the pedestrians in magenta pillars correspond to green ones in (c).
Figure 5.3: The CV became aware of a hidden pedestrian before turning from the main street to Little Queen St. The left column shows the images captured by the front-right camera on the CV at different times, the right column illustrates the corresponding road user tracking results within the CV. Also in the right column, the tracked pedestrians are represented as magenta pillars with their 2-sigma estimation confidence areas denoted as yellow ellipses, the IRSU is visualised in the CV as a tall white pillar. In (a) and (b), the CV could “see” through the building a visually occluded pedestrian (labelled with purple circles) around the corner with the help of CPMs received from the IRSU. This is seconds before the CV drove closer and could observe by itself the previously occluded pedestrian on Little Queen St, as depicted in (c) and (d). The localisation covariance is presented as purple disks along the vehicle trajectory in every right column figure. It can be seen in (f) that the uncertainty in road user tracking reduced after the CV received a correction in self-localisation.
5.2 Results in CARLA Simulator

The experiment demonstrates the autonomous operation of a CAV using the perception information received from an IRSU in CARLA simulator as the only source for pedestrian estimation. CARLA [57] is an open source game engine based simulator, which supports flexible configurations of sensors, road users, and urban traffic environments. One of the highlights of CARLA simulator is the various and realistic sensory data it provides, including RGB images, lidar point clouds, IMU and GNSS measurements. Another highlight is its ROS integration with the ros-bridge package. It is considered a safe, realistic, and repeatable environment for experimenting with autonomous driving systems, in particular, for conducting CAV operations that have safety concerns to test in the real world.

The experiment is setup with the IRSU detecting multiple walking pedestrians within its sensor range. The information of perceived pedestrians is then broadcast in the form of ETSI CPMs and is received by the CAV. The information is then transformed into its local frame of reference, and taken into account as the only perception data in the autonomous navigation of the CAV. The experiment is to demonstrate how a CAV interacts with VRU at non-designated crossing areas with the perception information from an IRSU. It is strongly recommended to conduct these experiments with higher risk CAV maneuvers in a simulator before performing them in real world.

The experiment was setup in the Town01 map of CARLA 0.9.8 simulator with participants labelled in Figure 5.4. These include an IRSU deployed static next to a straight road, a CAV navigating autonomously on the same road towards east, and two walking pedestrians. One of the pedestrians was crossing the road, while the other is walking along the road in the west direction. A Lanelet2 map was built for the experiment environment in the simulator, which is essential for the CAV to stay on the road and plan drivable paths for autonomous navigation. The map includes the lane markings and the network of drivable regions as per right hand traffic.
rules. Accordingly, the CAV respects right hand traffic rules in the demonstration. On the road where the CAV is navigating, the CAV has right of way over pedestrians and is allowed to cross the broken dividing line to overtake other road users when it is safe to do so.

The IRSU in the simulator is configured as close as possible to the real IRSU developed, which is equipped with two cameras and a 16-beam lidar, as revealed in Figure 5.4. It is visualised as a mini car parked on the roadside in the simulator. The same suite of sensory data processing algorithms for the real IRSU is adopted in the simulated IRSU. The road user perception information is encoded to ETSI CPMs in the form of binary payloads and broadcast by the IRSU. The CAV employed in the experiment is as also shown in Figure 5.4, which is able to position itself within the map using a combination of measurements from GNSS, IMU, and wheel encoder. The self-localization, however, is not assumed perfect but with noise added to the GNSS, IMU, and encoder measurements. A unscented Kalman filter (UKF) is used to achieve sub-meter accuracy. In the experiment, the CAV is also setup to receive ETSI CPMs from the IRSU. It should be noted that the CAV does not have any onboard sensors for detecting road users, instead, it becomes aware of the road users and infer their states purely relying on information received from the IRSU. The CAV navigates at a speed up to 18 km/h in the simulator.

In the experiment, one of the pedestrians was crossing the road at a non-designated crossing area while the other is walking along the road in the opposite direction to the CAV. Figure 5.5 demonstrates the road user detection and tracking in the IRSU at the beginning of the experiment. Specifically, Figure 5.5a shows the detection of the two pedestrians and the CAV by fusing the lidar point cloud and image frame from Cam #1 of the IRSU. The tracking of the same three road users in 3D space is shown in Figure 5.5b. The tracking of the walking pedestrians in the CAV is illustrated in Figure 5.6. Figure 5.6c depicts the planned path in the CAV to navigate around the pedestrians when approaching them. The tracked pedestrians are visualised with 95% confidence ellipses and arrows indicating their moving direction. When the grid-based cost map is updated in the hybrid A* path planner in the CAV, the current and future estimates of the pedestrian states account for a higher cost (darker colour) in the grid such that the path planner would consider them when searching for an optimal and kinematically feasible path to its goal. Please note that the physical dimension of the CAV has been considered by bloating the confidence ellipses in the cost map.
Figure 5.6: The pedestrian tracking in the CAV relying only on CPMs from the IRSU in the CARLA simulator. In (c), the grid-based cost map maintained by the path planner in the CAV is presented with white as free space and black as occupied areas, and the odometry trajectory is shown as a red line, and the planned path is presented as a dotted green line.

Figure 5.7: The CAV safely navigated around the crossing pedestrian autonomously based on the CPMs from the IRSU in the CARLA simulator.
In Figure 5.6, it is shown that the CAV responded to the pedestrian walking along the road by not altering its planned path as expected. Although the pedestrian was walking on the road, it was blocking only the opposite lane of the road. In Figure 5.6a, the CAV became aware of the crossing pedestrian even when it was approximately 35 meters away, with the information coming through the IRSU. This was not taken into account by the path planning since the CAV was still far from the pedestrian. As the CAV drove closer, it reacted to the event by planning a path that can safely navigate around the pedestrian while the pedestrian was crossing the road at a non-designated crossing area. As revealed in Figure 5.7, when the CAV was close to the pedestrian, it crossed the broken dividing lane to avoid the crossing pedestrian from behind. Following the avoidance maneuver, the CAV switched back to the right lane.

5.3 Results in A Lab Environment

Figure 5.8: Experiment setup in a lab environment. It is shown in the Lanelet2 map and lidar feature map built for the experiment site, which contains the road boundaries and lanes information. The light blue dots distributed at the bottom right corner represent the locations of observed features in the environment (trees, poles, corners of structures). The designated pedestrian crossing area is labelled as an orange rectangle.

The last experiment presented in the chapter investigates the CAV interacting with a pedestrian running towards a pedestrian crossing area. It was setup in a lab traffic environment with a 55m stretch of straight road, a virtual pedestrian crossing area, and the IRSU deployed next to the crossing, as illustrated in Figure 5.8. A Lanelet2 map was built for the environment and the CAV was able to localise itself within the pre-built map of lidar features, such as poles and building corners. The pedestrian crossing area was encoded in the Lanelet2 map which is stored locally in the CAV. The expected behaviour of the CAV is to drive through the crossing or stop before the crossing to give way depending on whether there is any pedestrian crossing activity detected by the IRSU.

Figure 5.9 shows the moment the pedestrian started running at the beginning of the experiment. The IRSU captured the road users in the scene, which include pedestrian and the CAV in 3D space. The CAV then received from the IRSU the information about the presence of a pedestrian running towards the crossing, as depicted in Figure 5.9d. When approaching him, the CAV kept tracking and predicting the pedestrian state based on the perception information.
from the IRSU. A constant velocity kinematic model was adopted for predicting the future state of the pedestrian. As shown in Figure 5.10b, the pedestrian was predicted by the CAV to occupy the crossing area, although he had not physically stepped onto the crossing yet. In the meantime, the CAV started to brake preemptively for the predicted crossing activity. The crossing area became a high cost region in the cost map within the path planner when the pedestrian was crossing. As a result, there was no feasible path found to drive through the crossing area for the CAV. It eventually stopped before the crossing line for the pedestrian, waiting until he finished crossing, as demonstrated in Figure 5.10d. When the pedestrian crossing area was clear a moment later, the CAV re-planed a path to the original goal and started to drive through the crossing at a slightly slower speed due to the road speed rule encoded in the map, as depicted in Figure 5.10f. As depicted in Figure 5.11, where the CAV speed measured by the wheel encoders is presented, the CAV managed to gain extra 1.5 s of response time to the pedestrian crossing activity. The braking behaviour was well planned and executed within the CAV for a safe and smooth deceleration.

The CAV onboard perception sensors were not used for pedestrian detection throughout the experiment. Yet, there are a few measures in place to ensure safety when conducting autonomous navigation related activities in real world experiments. First of all, the experiment was conducted in a closed, controlled environment and the CAV speed is limited to 11 km/h when in autonomous mode. Besides, there is always a safety driver sitting in the CAV closely monitoring any hazards and has the ability to trigger an e-stop of the CAV in case of an emergency. Furthermore, the virtual bumper subsystem based on lidar points obstacle detection is enabled on the CAV, which brings the vehicle to an e-stop if any obstacle breaches any of the pre-set distance thresholds to the vehicle. However, the virtual bumper treats every surrounding object as static obstacles and therefore did not take part in the decision making process of the CAV in Figure 5.10b, where it decided to give way based on the predicted future state of the pedestrian.
Figure 5.9: The CAV started tracking a pedestrian running toward a crossing line of a marked pedestrian crossing in a lab environment based on ETSI CPMs from an IRSU. (a) shows the image frame from Cam #1 of the IRSU with projected lidar point cloud. (b) and (d) illustrate the tracking of the road users in 3D space in the IRSU and the CAV, respectively, and (c) shows the image captured by the front-left camera on the CAV. The pedestrian and the safety driver of the CAV are tracked and denoted by magenta pillars with confidence ellipse and arrows indicating their moving direction, while the vehicle is labelled as an orange pillar in the IRSU’s tracker. Only the running pedestrian was tracked in the GMPHD tracker running within the CAV, as depicted in (d), as the CAV itself and its safety driver are identified and removed from tracking based on the location of the received perceived objects after transformed into its local frame. The Lanelet2 map is also visualised in (d), where the pedestrian crossing is denoted by an orange rectangle. Furthermore, the odometry trajectory of the CAV is shown as a red line, and its planned path is denoted as a dotted green line in (d).
Figure 5.10: The CAV gave way to a running pedestrian at the designated crossing area in a lab environment. Based on the ETSI CPMs broadcast by the IRSU, the CAV predicted the state of the running pedestrian up to 1.5 seconds into the future. The IRSU is visualised as a tall white pillar in the CAV in (b), (d), and (f). It is seen in (b) that the predicted states of the pedestrian, visualised as yellow ellipses, intersected with the orange crossing area on the Lanelet2 map. The CAV stopped for the pedestrian in front of the crossing in (d). The CAV then planned a new path forward after the pedestrian finished crossing in (f). In (a), (c), and (e), image frames from front-left, front-centre, front-right cameras on the CAV are presented, respectively.

Figure 5.11: The CAV speed between time 14 to 24 s in the experiment. It is shown that the CAV started to decelerate about 1.5 s prior to the moment the pedestrian physically stepped onto the crossing, as indicated by the black dashed line.
Chapter 6

Demonstrations of CAV Operation Using Cooperative Perception Information

6.1 Results in CARLA Simulator

The experiment was setup in the *Town01* map of CARLA 0.9.8 simulator with two moving CAVs and three walking/running pedestrians, as labelled in Figure 6.1. Both CAVs were approaching an intersection before making a right turn. A few cars were parked next to the intersection to create a visually blind spot for each CAV. It is assumed that there was no traffic lights setup at this intersection. It is also configured in the simulator that each CAV is able to position itself probabilistically within the map using measurements from GNSS, IMU, and wheel encoder, which are corrupted by noise. Both CAVs are equipped with a front facing camera and a 16-beam lidar for road user detection and tracking. Both CAVs are setup with the V2X communication capability and there are ETSI CPMs exchanged between them at 10 Hz. In the experiment, each CAV was fusing its local sensory information and the perception data transmitted by the other CAV through V2V communication. CAV1 was navigating autonomously on the road using the navigation stack developed by ACFR ITS group, while CAV2 followed a predetermined path.

In terms of perception sensory data processing, the 3D lidar point cloud is first projected to images of the front facing camera, followed by segmentation, clustering, and labelling of those points within the image bounding box frusta predicted by the visual classifier YOLOv3. As the CAVs are receiving perceived information from each other, the received road user tracks are first decoded, and transformed with its uncertainty to the vehicle’s local frame, taking into account the estimated egocentric poses of both CAVs in self-localisation.

There is an instance of GMPHD tracker running within each CAV for tracking the states of the road users within its local frame, including their position, heading, and speed, based on the detection results from the fusion of lidar and camera. The GMPHD tracker used in the experiment is capable of fusing both types of information, i.e., observations from local sources and tracks it receives from remote sources.

Please note that the CNN based visual feature extractor, as presented in Section 4.4.2, has been implemented as the `deep_encoder` ROS module. It is enabled on each CAV for pedestrian re-identification in the experiment, thereby improving pedestrian tracking performance.

In the experiment, CAV1 was moving up to 18 km/h in autonomous navigation, CAV2 was programmed to move at 10 km/h. The two walking pedestrians were moving at 2 m/s, while the running pedestrian moved at 2.8 m/s.

Figure 6.2 presents the perception and tracking results within two CAVs at the beginning of the experiment, precisely, at ROS time 6950.29 s. More specifically, CAV1 was observing and
tracking the running pedestrian using its local sensors, as depicted in Figure 6.2a, while CAV2 picked up the two walking pedestrians within its local sensor range, as illustrated in Figure 6.2b. In both Figure 6.2a and Figure 6.2b, the tracked pedestrians are visualised with 95% confidence ellipses and arrows indicating their moving direction. As the CAVs were sharing perception information through cooperative perception, CAV1 started to track the walking pedestrians, which were visually occluded, using the remote perception information in its local coordinate system, as shown in Figure 6.2c. CAV1 planned a path to move straightforward as the running pedestrian was on the other lane of the road, as illustrated in Figure 6.2d. Likewise, in the meantime, CAV2 were tracking all three pedestrians within its local frame by fusing local and remote perception information, although the pedestrian tracking within CAV2 is not shown in the figure. It is important to note that despite sharing of the common tracking information between two CAVs multiple times, the CI algorithm adopted for track-to-track fusion within the GMPHD trackers effectively prevented the inconsistent estimation issue from happening.

When approaching the intersection, CAV1 planned a feasible path to make a right turn at the intersection. In the meantime, the two walking pedestrians kept moving closer to the intersection, and CAV1 managed to continuously track their positions with the information from CAV2. The path planner within CAV1 kept updating its planned path based on the current estimated positions of the pedestrians. As illustrated in Figure 6.3, there was a path planned for turning right at the intersection up to ROS time 6953.89 s. At ROS time 6953.95
Figure 6.2: The RVIZ screenshot showing the beginning moment (at ROS time 6950.29 s). (a) and (b) illustrate the fusion of local sensors and detection of road users in CAV1 and CAV2, respectively. As shown in (a), CAV1 was able to observe the running pedestrian, but could not detect the two walking pedestrians behind the building. In (b), CAV2 were perceiving the two walking pedestrians while not able to detect the running pedestrian. In (c), CAV1 was tracking all three pedestrians using detection-to-track, and track-to-track fusion based on perception information from CAV2 through V2V communication. With the tracking results considered, CAV1 planed a path navigating forward, as shown in (d).

s, the two pedestrians were moving close to the intersection to a point that the path planner within CAV1 could not find a feasible path to make the turn, as shown in Figure 6.4d.

Later at ROS time 6958.47 s, as depicted in Figure 6.5, CAV1 decelerated and made a full stop to give way to the two crossing pedestrians. Please note that up to this point, CAV1 could not observe the two walking pedestrians using its local sensors due to visual occlusion. CAV1 responded to the crossing behaviour of the pedestrians based on cooperative perception information received from CAV2.

While CAV1 was waiting for the pedestrians to finish crossing, it began to observe the pedestrians using its onboard perception sensors when they walked out from the occlusion. As seen in Figure 6.6, at ROS time 6960.53 s and 6963.09 s, respectively, the tracking accuracy for the first and second walking pedestrians improved significantly in CAV1 with the corresponding local sensor detections becoming available for the GMPHD tracker.

In Figure 6.7, which shows the simulation at ROS time 6966.10 s, the intersection became clear for CAV1 to turn right. As Figure 6.7d shows, CAV1 planned a path and started turning. In the meantime, the running pedestrian was moving out of the local sensor range of CAV1, as Figure 6.7a reveals, and the tracking of the pedestrian was handed over to CAV2.

While CAV1 was turning, CAV2 started turning right too. As demonstrated in Figure 6.8, at ROS time 6969.95 s, CAV1 still managed to track the running pedestrian, although with increased uncertainty, relying on the perception information from CAV2.
Figure 6.3: The RVIZ screenshot showing the simulation at ROS time 6953.89 s. At this particular moment, a path was planned within CAV1 for turning right at the intersection.

Figure 6.4: The CARLA and RVIZ screenshot showing the simulation at ROS time 6953.95 s. Based on the tracking results of the two walking pedestrians behind the building, as visualised in (c), CAV1 could not find a feasible path to make a right turn at the intersection, as shown in (d).
Figure 6.5: The CARLA and RVIZ screenshot showing the simulation at ROS time 6958.47 s. CAV1 slowed and stopped in front of the intersection to give way to the walking pedestrian. As (a) shows, CAV1 still could not observe the pedestrians using its local sensors.
Figure 6.6: The RVIZ screenshot showing the simulation at ROS time 6960.53 s and 6963.09 s. In (a), CAV1 started to detect the first walking pedestrian using its own sensors at ROS time 6960.53 s. It managed to improve the tracking performance with the detection-to-track fusion, as shown in (c). In (d), CAV1 could detect both walking pedestrians using the onboard sensors at ROS time 6963.09 s. As a result, as shown in (f), the tracking uncertainty of the second walking pedestrian reduced significantly.
Figure 6.7: The simulation at ROS time 6966.10 s. As the area for turning right became clear at the intersection, CAV1 planned a path for turning in (d). In (a), CAV1 lost local detection of the running pedestrian. In (d), CAV1 managed to track the pedestrian using the information received from CAV2.
Figure 6.8: The screenshot showing the simulation at ROS time 6969.95 s. Both CAVs were making a right turn, as shown in (d). Although the running pedestrian was out of sensor range, CAV1 was still tracking the pedestrian with the help from CAV2. It is seen in (c) that the tracking uncertainty for the pedestrian grew slightly in CAV1, due to the absence of local sensor detection for the pedestrian.
6.2 Results in A Lab Environment

The section presents the real-world demonstration results of a CAV using cooperative perception information in a lab environment. The CAV performed fusion of perception information from a CV and an IRSU and considered it in its operation.

![Diagram of communication topology in the demonstrations. The CV and CAV were exchanging CPMs through V2V communication, while the IRSU was transmitting CPMs to both vehicles via V2I communication.](image)

Figure 6.9: Communication topology in the demonstrations. The CV and CAV were exchanging CPMs through V2V communication, while the IRSU was transmitting CPMs to both vehicles via V2I communication.

The communication topology of the participant ITS-Ss in the demonstrations is illustrated in Figure 6.9. As Figure 6.9 reveals, CPMs were exchanged between the two vehicles through V2V communication, while the IRSU was configured to send CPMs to the vehicles through V2I communication. A hybrid information fusion scheme was employed in the vehicles in both demonstrations. The IRSU broadcast object detection information, which requires detection-to-track fusion when received by the vehicles. At the same time, object tracks were shared between the vehicles, which require track-to-track fusion in the vehicles. The IRSU was setup in this demonstration so that it only shares its independent detection information instead of tracks, which are usually correlated with one another.

The demonstration site has been surveyed with a Lanelet2 map built. Both vehicles were using RTK to localise themselves during the demonstrations. Also, the vehicles had a self-exclusion mechanism in place to avoid themselves being tracked when they are detected by other ITS-Ss.

The demonstration at the lab was setup as shown in Figure 6.10 and Figure 6.11. It can be seen that there was an overlap of the perception areas of the CV and the IRSU. As depicted in Figure 6.12 and Figure 6.13, with the perception information from the two sources received, the CAV performed fusion and managed to track the walker crossing the road. Figure 6.14 provides an example comparing before and after the information fusion, and illustrating the improvement in estimation accuracy after the fusion. The behaviour of the CAV was as expected – it gave way to the crossing pedestrian when it had obtained the information from the IRSU and the CV, it then continued to move after the crossing was detected to be clear.
Figure 6.10: Satellite map of the lab annotated with the road boundaries, lane information, and a designated pedestrian crossing area. The map also shows the CV and the IRSU near the pedestrian crossing, and the CAV moving northeast towards the crossing.

Figure 6.11: Demonstration setup in the lab environment.
Figure 6.12: Screenshots of the demonstration video showing different time instances: (a) The CAV approaching the pedestrian crossing while it became aware of the crossing pedestrian. This information is provided through the fusion of information from the CV and the IRSU; (b) The CAV stopping in front of the crossing for the pedestrian; (c) The CAV planning a path moving forward when the crossing became clear; (d) The CAV slowly moving over the crossing.

Figure 6.13: RVIZ screenshot of the IRSU processing computer showing the same scene as in Figure 5(b) from the IRSU perspective.
Figure 6.14: Comparison of before and after fusion in the CAV. The perception information before fusion is shown in the top figure, where there are two purple pillars corresponding to the track and the detection of the same target received from the CV and the IRSU, respectively. The tall grey pillars represent the locations of the CV and the IRSU. The bottom figure shows the fused perception information in the CAV, which is presented as a single Gaussian with lower uncertainty (the inner yellow disc). The outer yellow hollow ellipses represent future motion prediction of the pedestrian.
6.3 Results in USYD Campus

This section presents the results of another real-world demonstration of autonomy of CAV using cooperative perception information. The IRSU prototype B, as described in Section 3.2, was built and used. The demonstration was conducted at the USYD campus, where a CAV received the fused perception information from a parked CV and an IRSU and considered it in its autonomous driving operation. In this case, the CV was practically functioning as a second IRSU. Unlike the case in previous demonstrations, the cross-platform data fusion in this demonstration was performed within the IRSU and the CV, while the CAV used the fused information directly. The perception information is shared as CPMs amongst the three ITS-Ss.

The experimental results show that the CAV, setup without any onboard sensing capability, managed to make a safe right turn at a T-junction in three different scenarios, based on the fused perception information from the CV and the IRSU. As Figure 6.15b will show, the three scenarios represent different traffic conditions at the T-junction. Specifically, they are:

- Scenario A: CAV giving way to a pedestrian blocking the turning lane at the T-junction.
- Scenario B: CAV giving way to a vehicle moving straight at the T-junction.
- Scenario C: CAV giving way to a vehicle turning left at the T-junction.

It is also demonstrated that in each scenario, the CAV considered the fused perception information in its path planning and decision making when navigating autonomously in the real-world environment.

6.3.1 Demonstration Setup

The setup of the demonstration conducted at the USYD campus is illustrated in Figure 6.15. The demonstration participants, i.e., the IRSU, the CV, and the CAV, are shown in Figure 6.16. As Figure 6.15 shows, the CV and the IRSU mainly monitor different regions at a T-junction, with a small overlapping area of their sensor detection areas. This means the CV sometimes can see road users that the IRSU cannot see, and vice versa. Yet, each of them has the combined knowledge through data fusion.

The demonstration consists of three scenarios, showing how the cooperative perception can improve the safety of the CAV operation under different traffic conditions. In each scenario, the CAV relied on the fused perception information from the IRSU and the CV to check if the T-junction was clear to making a right turn. In the CAV, a human-in-the-loop (HILT) safety feature has been added to let the CAV safety driver perform a final check of the safety before the turning. A GUI shown on the CAV touchscreen will print “Safe to go? (seems unsafe)” message on a button, warning the safety driver when the cooperative perception algorithm finds it unsafe to turn, and “Safe to go? (seems safe)” when the algorithm believes it is clear to turn. By clicking the button the safety driver confirms it is safe and the CAV will start turning.

Participant ITS Stations

The communication topology of the participant ITS-Ss in the demonstration is illustrated in Figure 6.17a. As shown in Figure 6.17b, the new IRSU and the CV detect and track road users, including pedestrians and cars, within their sensors' range and field-of-view (FOV). The real-time tracking information is then exchanged between the IRSU and the CV in the form of CPMs through V2I communication, and cross-platform data fusion is performed in each of
Figure 6.15: Demonstration site and setup at the USYD campus. The demonstration area is highlighted by the orange rectangle in (a). The three scenarios in the demonstration are summarised in (b).

them. The CAV can receive the CPMs from both the IRSU and the CV, and use it in its path planning and decision making processes in the autonomous driving. The CV was stationary in the demonstration, which means it was acting as another IRSU. For this reason, we can treat the IRSU and the CV holistically as intelligent infrastructure where global information fusion is performed. Essentially, the information broadcast by any of them is the combined knowledge from both ITS-Ss. Hence, although the CAV can receive CPMs from both the IRSU and the CV, it is configured in the demonstration to only need to consider the perception information
Figure 6.16: Demonstration participants. The IRSU deployed beside the main road, with its sensor facing the T-junction area, as shown in (a). The CV was parked, as revealed in (b), functioning as another IRSU. In (c), the CAV was waiting for a safety clearance signal to make a right turn.

from the IRSU. It is important to note that, apart from the coordinate transformation of the received perception information, there was no tracking or fusion taking place within the CAV. This configuration moves the computationally expensive data fusion work to the infrastructure, lowering the computation resource requirement on the CAV side.

The developed cooperative perception approach is also employed in the CV. The main difference compared to the IRSU is that for the CV, the perception relies on its onboard Velodyne VLP-16 lidar and the front GMSL camera. The real-time visual detection task is carried out by the YOLOv4 Tiny running on the onboard NVIDIA Drive PX2.

The onboard perception sensors in the CAV are disabled. In this demonstration, as mentioned earlier, there is no tracking of road users nor fusion of information after the CPMs are received through V2X communication. The CAV is configured only to use the perceived objects information from the IRSU.
Figure 6.17: Communication topology and system modules in the demonstrations. In (a), the CV and IRSU were exchanging CPMs through V2I communication. The CV and the IRSU were transmitting CPMs to the CAV via V2V and V2I communication, respectively. Note that in (b), the data fusion takes place between the CV and the IRSU, there is no fusion or road user tracking in the CAV.

**IRSU and CV Data Fusion**

As mentioned earlier, the demonstration is setup such that the cross-platform data fusion takes place only between the IRSU and the CV. The parked CV in this demonstration served as
another IRSU. Figure 6.18 and Figure 6.19 illustrate some examples of IRSU and CV data fusion results from the CV’s perspective. The IRSU is visualised as a white pillar in the figures. It is shown in Figure 6.18 the tracking of a pedestrian and a moving car. Figure 6.18a and Figure 6.18b are just a couple of seconds apart. The pedestrian, who was too far to be detected by the IRSU, was picked up by the local sensors of the CV, and thus a lower estimation uncertainty in the figure, as indicated by the red arrows. The moving car at a further distance, indicated by the yellow arrows, was beyond the reach of the CV’s sensors, but was observed and tracked by the IRSU. Through data fusion, the CV achieves an augmented sensing area, and can track both the pedestrian and the moving car. Although not shown in Figure 6.18, the same augmented sensing area is achieved in the IRSU.

In Figure 6.19, it is illustrated the continuous cooperative tracking of a pedestrian running back and forth between the CV and the IRSU, using the data fusion between the two ITS-Ss. The pedestrian was first out of sensor FOV of the CV in Figure 6.19a but could be tracked by the CV with the help of the IRSU. The pedestrian then was detected locally by the CV and thus tracked more precisely in Figure 6.19b and Figure 6.19c. Lastly, in Figure 6.19d, the CV was still tracking the pedestrian, even though he ran out of the scene again. When the CV could not detect the pedestrian using its perception sensors, the estimation uncertainty in tracking increased, as marked by the yellow circles in the figure, due to the extra uncertainty considered in the coordinate transformation.

Figure 6.18: Augmented sensing area of the IRSU and the parked CV. (a) and (b) show the same scene with only a couple of seconds apart.
Figure 6.19: Cooperative tracking of a pedestrian running back and forth between the parked CV and the IRSU. The yellow arrows indicate the running direction. The IRSU is visualised as a white pillar. In (a) and (d), the pedestrian ran out of the scene, yet the CV still managed to track the pedestrian continuously, as marked by the yellow circles, using the data fusion between the CV and the IRSU.
6.3.2 Scenario A

Figure 6.20: The demonstration setup in the Scenario A, where a pedestrian was blocking the turning lane of the CAV.

In this scenario, as shown in Figure 6.20, a standing pedestrian was first blocking the turning lane of the CAV. The pedestrian was beyond the detection range of the IRSU, and therefore was only detected by the CV. The perception information was shared between the CV and the IRSU. The CAV was receiving CPMs from both the CV and the IRSU, but it only used the fused information from the IRSU in its operation.

Figure 6.21 depicts the situation from the perspective of the CAV based on the fused perception information received from the IRSU. As Figure 6.21a show, the CAV became aware of the pedestrian when it was approaching the T-junction. In Figure 6.21b, the CAV stopped and gave way to the pedestrian. The HILT GUI was prompting unsafe to turn message to the safety driver of the CAV, as the bottom right corner in Figure 6.21b shows. After the pedestrian walked out of the T-junction region, as Figure 6.21c illustrates, the CAV deemed it safe to turn based on the received fused perception information and requested confirmation from the safety driver through the HILT GUI, as the bottom right corner in Figure 6.21c shows. Once the safety driver clicked the button to confirm, the CAV started to turn, which is shown in Figure 6.21d.
Figure 6.21: The Scenario A from the perspective of the CAV based on the fused perception information shared by the IRSU. In (a), the CAV was receiving the road user activity from the IRSU when approaching the T-junction. Note that the pedestrian was only detected by the sensors of the CV. However, the IRSU was able to track the pedestrian through data fusion between the CV and the IRSU. The positions of the IRSU and the CV are indicated by the blue arrows. In (b), the CAV was giving way to the pedestrian, who was blocking the turning lane. The pedestrian is marked by the red arrows. The HILT GUI, as shown at the button right corner in (b), prompted the unsafe to turn message to the safety driver. In (c), the CAV deemed it safe to turn after the pedestrian walked out of the T-junction area. In (d), the CAV made the right turn with the confirmation from the safety driver through the HILT GUI.
6.3.3 Scenario B

Figure 6.22: The demonstration setup in the Scenario B, where a vehicle was moving straight on the opposite lane. The CV is not shown in the figure.

In the second scenario, a vehicle was moving straight on the opposite lane, as Figure 6.22 demonstrates. The CAV in this case has to respect the traffic rule at this T-junction, by giving way to the vehicle. As Figure 6.23b illustrates, the IRSU was able to detect and track the vehicle before the vehicle reached the T-junction. The CAV became aware of the coming vehicle through the IRSU and decided to give way to the vehicle. The CAV waited for the vehicle to pass the T-junction, as Figure 6.23c and Figure 6.23d show. It finally made the right turn after the T-junction became clear and received confirmation from the safety driver through the HILT GUI, as shown in Figure 6.23e.
Figure 6.23: The Scenario B from the perspective of the CAV based on the fused perception information received from the IRSU. In (a), the CAV was receiving the road user activity from the IRSU when it was approaching the T-junction. In (b), the CAV became aware of the coming vehicle on the opposite lane through the IRSU and had to give way to the vehicle according to the traffic rule. The vehicle is marked by the yellow arrows. In (c), the CAV was waiting for the vehicle to pass the T-junction area. In (d), the CAV deemed it safe to turn based on the fused perceived information from the IRSU. With the confirmation from the safety driver through the HILT GUI, the CAV made the right turn, as shown in (e).
6.3.4 Scenario C

Figure 6.24: The demonstration setup in the Scenario C, where a vehicle on the opposite lane was turning left at the T-junction. The CV is not shown in the figure.

In the last scenario, instead of moving straight, the vehicle on the opposite lane made a left turn at the T-junction, as illustrated in Figure 6.24. The expected behaviour of the CAV in this scenario was the same as in the last scenario—respecting the traffic rule by yielding to the vehicle. The CAV stopped in front of the T-junction for an extended period of time in this scenario, because it took the vehicle a longer time to move out of the T-junction, and a public pedestrian entered the region right after the vehicle.

As Figure 6.25b shows, the CAV became aware of the coming vehicle through the IRSU when it was approaching the T-junction. When the vehicle made a left turn at the T-junction, it was first detected by the IRSU, as shown in Figure 6.25c. When the vehicle moved further away, it only could be detected by the CV, as illustrated in Figure 6.25d. However, the IRSU in this case still managed to track the vehicle through data fusion of the CV and the IRSU. Immediately after the vehicle moved away, a public pedestrian walked into the T-junction region, as depicted in Figure 6.25e, causing it still unsafe to turn for the CAV. After the pedestrian left the region, as shown in Figure 6.25f, the CAV was cleared to make the right turn, and it did so after receiving confirmation from the safety driver, as illustrated in Figure 6.25g.
Figure 6.25: The Scenario C from the perspective of the CAV based on the fused perception information from the IRSU. In (a), the CAV was approaching the T-junction. In (b), the CAV became aware of the coming vehicle on the opposite lane through the IRSU and had to yield. The vehicle is marked by the yellow arrows. In (c) and (d), the CAV was waiting for the vehicle to finish turning at the T-junction. The vehicle was first observable by the IRSU in (c), then only by the CV when it moved further away in (d). In (d) and (e), a public pedestrian was walking in the T-junction area, and the CAV had to wait a longer time. The pedestrian is marked by the red arrows. In (f), the CAV found it safe to turn after the pedestrian left the area. It then made the turn, as illustrated in (g).
6.3.5 Remarks

In the demonstration, the ZED SDK worked well to detect and track multiple road users, including pedestrians and vehicles. However, it is found that it struggled to classify the CAV platform as an object under the “vehicle” category due to its odd shape. For this reason, the detection of the CAV by the IRSU was intermittent, which causes problems in the tracking process. As Figure 15 reveals, there were phantom object instances reported by the IRSU besides the real position of the CAV in all three demonstration scenarios. We consider this an implementation issue that will not exist in most practical urban applications.

Figure 6.26: The phantom object instances (marked by the green arrows) reported by the IRSU when it was detecting and tracking the CAV platform in the demonstration. (a), (b), and (c) correspond to the moment when the CAV was turning right at the T-junction in the Scenario A, Scenario B, and Scenario C, respectively.
Chapter 7

Conclusions and Future Work

The report mainly investigates CP, an emerging and promising technology that can substantially improve the efficiency and safety of road transportation. Essentially, it focuses on safety and robustness implications the CP technology can bring to the operations of CAVs. The report presents a CP framework and IRSU and CAV platforms developed by the joint research team from the ACFR and Cohda Wireless over the last three years, and demonstrates through a collection of representative experiments the benefits associated with the use of CP service for CV and CAV operations alongside VRU in different traffic contexts.

In the experiment presented in Section 5.1, which is conducted on a public road, a CV was able to track a pedestrian visually obstructed by a building with CP information from an IRSU deployed on the road side. This was achieved seconds before its local perception sensors or the driver could possibly see the same pedestrian around the corner, providing extra time for the driver or the navigation stack to react to a safety hazard.

The experiment conducted in CARLA simulator, as presented in Section 5.2, shows the autonomy of a CAV using the CP service. The experiment highlights the autonomous navigation of the CAV and its safe interaction with walking pedestrians, purely responding on the perception information provided by the IRSU.

The experiment in Section 5.3 demonstrated the expected behaviour of a CAV when interacting with a pedestrian rushing towards the designated crossing area in a lab environment. The CAV managed to take preemptive action, that is, braking and stopping before the crossing area for the pedestrian based on the kinematic prediction of the pedestrian. The pedestrian tracking, prediction, path planning and decision making in the CAV in the experiment were operating based on the perception information received from the IRSU.

Section 6.1 presents another experiment setup in CARLA simulator, where two moving CAVs were sharing perception information of three walking/running pedestrians within their sensor range through V2V communication. The experiment validated the CI based track-to-track fusion, the multiple road user tracker combined with the CNN based visual feature extractor, the CAV navigation stack, etc., in a simulated traffic environment. It is demonstrated that CP plays an essential role in the safety of CAV operations, in particular, in scenarios where the CAV is unable to detect road users using local sensors.

It is demonstrated through the experiment in Section 6.2 that a CAV, setup without any onboard sensors turned on for object detection, managed to fuse the perception information of a crossing pedestrian from a CV and an IRSU. The experiment validates the efficacy of the cross-platform data fusion approach and the CAV navigation stack in a lab environment. It is highlighted in the experiment that the CAV managed to extend its perception range and increase perception quality by fusing the local sensor data and remote perception information received
from other ITS-Ss through V2X communication.

Lastly, the experiment conducted in the USYD campus, as presented in Section 6.3, shows that a CAV managed to make a safe right turn at a T-junction in three different scenarios, relying only on the fused perception information from a CV and an IRSU. Unlike the case in previous experiments, the cross-platform data fusion in this demonstration was performed within the IRSU and the CV, while the CAV used the fused information directly. It is also demonstrated that in each of the scenarios tested, the CAV considered the fused perception information in its path planning and decision making when navigating autonomously in the real-world environment.

We believe these demonstrations can assist engineers and researchers in the relevant fields in achieving a better understanding of the safety implications the CP is bringing to the current and future transportation systems. The CP enables the smart vehicles to break the physical and practical limitations of onboard perception sensors, and in the meantime, to embrace improved perception quality and robustness along with other expected benefits from the CP service and V2X communication. As importantly, the CP can also reduce the reliance on the vehicle’s local perception information, thereby lowering the requirement and cost for onboard sensing systems. Furthermore, it is demonstrated that when properly used, perception data from other ITS-Ss can be used as another reliable source of information to add additional robustness and integrity to autonomous operations. These potentially trigger more in-depth discussions in the research community and industry related to the future development directions of intelligent vehicles and other C-ITS services and applications, such as maneuver intention sharing and cooperative driving.

The future work will focus on developing more advanced and ready-to-deploy ITS-S platforms that can better suit the future road transportation and building an open standard high definition (HD) map for CP enabled autonomous driving purpose. A comprehensive set of algorithms will also be developed towards safer, lower cost, and more efficient operation of CAVs, exploiting the HD map, V2X, and CP technologies. Lastly, we will aim to test the developed CP and CAV technologies in more sophisticated traffic environments.
Funding

This research is funded by iMOVE CRC and supported by the Cooperative Research Centres program, an Australian Government initiative.
Acknowledgment

The authors would like to thank HxGN SmartNet for providing Corrections and Data Service in the research work.
Bibliography


77


[48] I. Llatser, T. Michalke, M. Dolgov, F. Wildschütte, and H. Fuchs, “Cooperative automated driving use cases for 5g v2x communication,” in Proc. the IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, Sep. 2019, pp. 120–125.


