Improving Automated Vehicle Safety with Collective Perception

by

Siqi Yan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

Automated Vehicle (AV) is a trending technology being developed with the promise to reduce traffic accidents caused by human errors. Perception plays a crucial role for Automated Driving Systems (ADS) to make safe decisions. However, local sensory data is insufficient to capture comprehensive information for ADS due to occlusion and limited perception range. Vehicle-to-vehicle (V2V) communication technology provides an alternative solution by enabling vehicles to exchange perception information (referred to as Collective Perception or CP), increasing environment awareness for all vehicles in the network, and thus improving driving safety. In this research, we demonstrate the safety benefits of CP through the simulation of three driving scenarios in the Carla simulator. We use pose graph optimization (PGO) to fuse high-level sensory data between the local sensor and the data coming from the V2V network to reduce state estimation uncertainty. After that, a connected ADS was developed to make driving decisions based on the fused information to prevent potential traffic hazards. We show that the time-to-collision (TTC) metrics between the ego vehicle and unconnected road users (URU) are improved for all three scenarios compared to the baseline, indicating that CP can improve safety for AV in the simulated scenarios.

Acknowledgments

I would like to express my deepest appreciation to my supervisor Dr. Hong Zhang for getting me involved in the field of computer vision and robotics, sharing his knowledge, and guiding me through graduate studies in this difficult time due to the Covid-19 pandemic. I wouldn't have made it without the support you gave me.

I would also like to extend my sincere thanks to my examiners, Dr. Zhi-Jun Qiu and Dr. Jessy Kang, who attended my exam at midnight in their timezone and gave constructive feedback to strengthen my thesis. A thank you to Dr. Vadim Bulitko for being my exam chair and presiding over a smooth and comfortable examination.

I am also grateful to my colleagues Dr. Xiaolong Wang and Ehsan Ahmadi for conducting the initial research and providing valuable ideas to get me started in my research. Thanks also to Dr. Shaochen Luo and Dr. Huifang Ma for giving me helpful advice during research meetings.

Last but not least, I am deeply indebted to my parents for supporting my life and encouraging me when I made my decision to study abroad.

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Objective 3

1.3 Thesis Organization 5

2 Background **6**

2.1 Collective Perception 6

2.2 SLAM 8

2.3 Safety Evaluation 11

2.4 Simulators 15

3 Methodology **21**

3.1 Information Fusion Framework 21

3.2 Implementation Approach 24

3.3 Experiment Design 27

4 Result and Interpretation **33**

4.1 Parameter Selection 33

4.2 Safety 36

5 Conclusion **46**

Bibliography **48**

List of Tables

2.1 NHTSA Pre-crash Scenarios in the Challenge 19

4.1 Safety Aggregate Indicators 40

4.2 Welch T-test 40

List of Figures

1.1 Illustration of Collective Perception. In the cartoon, two CAVs are GPS enabled, and they can share data with each other through V2V communication. Both vehicles are perceiving the same URU that falls in the FOV and perception range; at the same time, indicated by the purple arrow and green arrow, which denotes an indirect encounter. The two CAVs can also perceive each other, as shown in the blue line segment, which denotes a direct encounter. The CAVs also have odometry sensors to constrain their consecutive poses, as shown in gray lines. The fusion process estimates the surrounding state by considering all sensor measurements coming from all connected CAVs, taking advantage of observation redundancy. Future trajectories between CAV and URU are predicted (shown in orange dashed arrows) based on the state estimation to detect possible hazards and improve safety. 4

2.1 The circle algorithm [30] models vehicles as circles, and the collision between two vehicles can be detected if the distance between the centers of two circles equals the sum of their radii. Here, we assume two vehicles are moving in straight lines at constant velocities, so their TTC can be obtained by finding the time at which the two circles intersect. 14

2.2 A screenshot of the SUMO simulator [31] 16

2.3 Carla Autonomous Driving Challenge [33] Screenshots 18

2.4 A screenshot of the Visualizer. We developed the visualizer for inspecting the recorded driving data during the Carla Autonomous Driving Challenge. The figure shows a bird's-eye view of the traffic situation centered at the top of the ego vehicle, the two blue line segments for the FOV, and the yellow circle indicates the communication range between the ego vehicle and the surrounding CAVs. Orange dots and lines represent the detected position and predicted future trajectories of the nearby obstacles. Blue dots denote the local waypoints, and gray dots denote the global waypoints. Finally, the connectivity between CAVs is shown in purple. 20

3.1 Factor Graph illustrating the relation of variables and corresponding measurement constraints in time sequence. The blue circle denotes the unknown variable, and a black dot on an edge denotes the measurement constraint. p is the prior of the origin o, each pose variable is constrained to the origin by GPS and IMU measurement g, two consecutive pose variables are constrained by an odometry measurement u. If an encounter occurs, the two corresponding object poses are constrained by a perception measurement c (red line). 23

3.2 System Diagram for our connected ADS. The ADS consists of six sub-modules: detection, localization, data fusion, prediction, planning, and control, and the sub-modules form a computation pipeline, in which the output of one module is passed into the next module, indicated by the arrows. Outside the connected ADS are different components that belong to the simulated world, which provide input sensory data and generated state transition based on the output of the connected ADS. Additional detection and localization information will be provided to the data fusion module in CAM and CPM, in the perspective of CAV. 25

3.3 Illustration of simulated detection. We do not use real object detection algorithms in our experiment. To simulate the effect of occlusion, we attach a semantic camera to the same location as the imaginary camera for detection. Both the semantic camera and the detection camera have the same intrinsic matrix $\mathbf{K}$. We project the pose of the target on the camera pixel coordinate $x(u, v, 1)^T = \mathbf{K}(y, z, x)^T$, and check whether the pixel in the semantic image belongs to the same category of the target. 26

3.4 Custom Scenario 1. On the right-hand side, the black car is the ego vehicle operated by our connected ADS. The dashed arrows in pink, orange and green represent the moving trajectories of the ego vehicle, pedestrian, and the CAV, respectively. When the ego vehicle is doing a right turn, the pedestrian will start crossing the road, which leads to the potential accident that the ego vehicle needs to avoid. The ego vehicle will come to a full brake upon detecting the pedestrian. The detection of the pedestrian will happen earlier from the CAV's point of view. Therefore, with the CP from the CAV, the ego vehicle will stop at an earlier time, thus increasing the TTC to the pedestrian. . 28

3.5 Custom Scenario 2 (in different views). The dashed arrows in pink, orange, and green represent the moving trajectories of the ego vehicle, pedestrian, and the CAV, respectively. The pedestrian will start to cross the road if the ego vehicle is close enough, leading to a potential traffic accident. The ego vehicle will brake immediately upon the detection of the pedestrian. In Figure 3.5a, the pedestrian is invisible to the ego vehicle due to occlusion but is visible from the CAV's point of view. Therefore, with CP from the CAV, the ego vehicle will stop at an earlier time than using a local sensor only. This will increase the TTC between the ego vehicle and the pedestrian. 29

3.6 Custom Scenario 3. The dashed arrows in pink, orange, and green represent the moving trajectories of the ego vehicle, UV, and the CAV, respectively. The UV is driving at a constant velocity of 100 km/h. The ego vehicle has an initial velocity of 50 km/h and will attempt to make a left turn at the intersection. Both the UV and the ego vehicle will enter the intersection at roughly the same time, and the ego vehicle needs to stop and yield to the UV if the UV is detected. The stopping location of the ego vehicle depends on the timing of detection. In the meantime, a CAV driving on the right lane ahead of the ego vehicle can provide an extended detection range, allowing early detection of the UV and increased TTC. 31

4.1 Coordinate Systems. The origin o of the GPS coordinate is shown in the figure, and the local coordinate system of the vehicle camera is denoted by the red, green, and blue axes, with a rotation angle θ to the GPS coordinate o. 34

4.2 URU Localization Accuracy. The figure compares the difference between the baseline, Raw CP (un-fused CP) and CP method in localizing URU with different standard deviations of the camera detector. Raw CP has roughly the same ATE as the baseline, which is expected because both methods do not attempt to reduce localization error. The Raw CP curve shows a smaller variation compared to the baseline because the effect of Raw CP is equivalent to averaging two independent episodes of the baseline. CP outperforms both the baseline and Raw CP by a large margin, but the differences start to diminish as the detection uncertainty increases. This happened because the increased detection uncertainty level reduces the confidence in the corresponding measurement constraint during optimization. 35

4.3 Scenario 1 Safety. The Figure shows a comparison of TTC between the ego vehicle and the URU as a result of using different state estimation methods. The baseline here represents the worst situation when the TTC reached its lowest level at around 1.2 seconds. This happened because the URU was not visible due to occlusion at the beginning of the right turn, and there wasn't enough time for the ego vehicle to take action to reduce its speed. CP outperforms the baseline because, for the same situation, the CAV is capable of detecting the URU and informing the ego vehicle to reduce its speed early. The Raw CP has an even better TTC in this case because it did not attempt to optimize the URU pose, and that can lead to higher uncertainty in the URU localization and trigger the emergent brake earlier than CP. 37

4.4 Custom scenario 1. The top figures 4.4a and 4.4b are the bird's-eye views of the ego vehicles using the semantic camera images, where the ego vehicles are the dark blue cars at the center of the images, and the URUs (i.e., the pedestrians) are marked in red. The semantic images show the states where the ego vehicles were completely stopped for the URUs. We can see that in Figure 4.4a, the baseline agent was stopped at a distance slightly closer to the URU than the CP agent in Figure 4.4b, which leads to a higher risk in terms of a smaller value of TTC for the baseline than CP. The bottom two figures, 4.4c and 4.4c, are the corresponding state estimations of the environments for the two methods. Within the bottom two figures, pink triangles and green triangles denote the camera poses of the ego vehicle and the CAV, respectively, and the pink lines and the green lines denote the detection measurement with respect to each camera. Blue lines show the direct encounter between the ego vehicle and the CAV. Gray lines connect the two cameras' poses for the odometry measurements. Orange dots show the URU observations. The Cyan line denotes the predicted trajectory of the URU. Finally, the red circle denotes the predicted collision location. 41

4.5 Scenario 2 Safety. The Figure shows a comparison of TTC between ego vehicle and URU as a result of using different state estimation methods. The baseline here represents the worst situation when the TTC reached its lowest level at around 1.1 seconds. Because the URU is occluded by the red truck as the ego vehicle is passing through, as shown in Figure 3.5, and when the URU is visible, the ego vehicle does not have sufficient braking time. Both CP and Raw CP have roughly the same values in TTC, and both of them outperform the baseline because, for the same situation, there was a CAV capable of detecting the URU, and informing the ego vehicle to reduce its speed early. . . 42

4.6 Custom scenario 2. The top figures 4.6a and 4.6b are the bird's-eye views of the ego vehicles using the semantic camera images, where the ego vehicles are the dark blue cars at the center of the images, and the URUs (i.e., the pedestrians) are marked in red. The dark blue trucks beside the ego vehicles are static and only used to block the detection of the URUs in the ego vehicle's points of view. We can see that in Figure 4.6a, the baseline agent stopped at a distance much closer to the URU than the CP agent in Figure 4.6b. This leads to a higher risk in terms of a smaller value of TTC for the baseline than CP. The bottom two figures, 4.6c and 4.6c, are the corresponding state estimations of the environments of the two methods. Within the bottom two figures, pink triangles and green triangles denote the camera poses of the ego vehicle and the CAV, respectively, and the pink lines and green lines for the detection measurement with respect to each camera. Blue lines show the direct encounter between the ego vehicle and the CAV. Gray lines connect the two cameras' poses for the odometry measurements. Orange dots indicate the URU observations. The cyan line denotes the predicted trajectory of the URU. Finally, the red circle denotes the predicted collision location. 43

4.7 Scenario 3 Safety. The Figure shows the TTC curve of the baseline in scenario 3. The curve is roughly linear because the URU is approaching to the ego vehicle at a constant speed when the ego vehicle is stopped at the intersection in Figure 3.6. When the URU passes through, there is no chance for a collision between the ego vehicle and the URU. Both the CP agent and the Raw CP agent were able to stop early enough without entering the intersection so that the heading direction of the ego vehicle is parallel to the trajectory of the URU (see Figure 4.8b), and the TTC is not measurable in this situation. 44

4.8 Custom scenario 3. The top figures 4.8a and 4.8b are the bird's-eye views of the ego vehicles using the semantic camera images, where the ego vehicles are the dark blue cars at the center of the images, and the NVs are also the dark blue cars in front of the ego vehicles. We can see that in Figure 4.8a, the baseline agent has entered the intersection and stopped at a distance very close to the UV. This is a dangerous behaviour and may lead to a traffic accident. Compared to the baseline, the CP agent did not enter the intersection, as shown in Figure 4.8b. The bottom two figures, 4.8c and 4.8c, are the corresponding state estimations of the environments of the two methods. Within the bottom two figures, pink triangles and green triangles denote the camera poses of the ego vehicle and the CAV, respectively, and the pink lines and green lines denote the detection measurement with respect to each camera. Blue lines show the direct encounter between the ego vehicle and the CAV. Gray lines connect the two camera's poses for the odometry measurements. Orange dots show the URU observations. The cyan line denotes the predicted trajectory of the URU. Finally, the red circle denotes the predicted collision location. 45

Abbreviations & Acronyms

AV Automated vehicle.

CAM Collective awareness message.

CAV Connected automated vehicle.

CP Collective perception.

CPM Collective perception message.

FOV Field-of-view.

PGO Pose graph optimization.

SLAM Simultaneous localization and mapping.

TET Time exposed Time-to-collision.

TIT Time integrated Time-to-collision.

TTC Time-to-collision.

URU Unconnected road user.

UV Unconnected vehicle.

Chapter 1

Introduction

1.1 Motivation

Automated Vehicles (AV) is a trending technology that has attracted lots of attention in recent years. According to the National Highway Traffic Safety Administration (NHTSA), human drivers were responsible for 94 percent of 2,189,000 crashes between 2005 and 2007, where recognition errors, decision errors, and performance errors accounted for 41%, 33%, and 11% of crashes, respectively [1]. Automated Driving Systems (ADS) are being developed and have the potential to gradually replace human drivers and reduce traffic accidents in the near future.

Perception plays a crucial role for ADS to make safe driving decisions; inaccurate or failed perception of the state of the environment can lead to severe outcomes. Robust ADS relies on fusing different types of sensors to gather a comprehensive state estimation of the environment to make safe driving decisions because each type of sensor has its own strengths and weaknesses. For example, an RGB camera is a cheap solution to provide high-resolution color and spatial information, which is useful for detecting and localizing objects such as pedestrians, vehicles, and traffic signs, but the quality of the camera image is largely dependent on illumination and weather conditions [2] [3]. Radar is not affected by illumination and weather conditions and can measure the depth information and the speed of objects at great

distances, but the emission of radio signals can interfere with other systems, and it is low in data resolution and prone to false detection of metal objects [3] [4] [2]. LiDAR features accurate range measurements for objects under 200 meters, large field-of-view (FOV), and less interference against illumination and weather conditions compared to cameras, but has a low vertical resolution, sparse point measurement along with distance, poor detection of dark objects, and is expensive [2] [3] [4] [5].

However, even with sensor fusion to maximize the perception coverage, there still exist difficulties using only local sensory data, such as occlusion and limited perception range. Adding multiple sensors can help but is not sufficient to overcome these limitations and may significantly increase the cost of AVs. Vehicle-to-vehicle (V2V) communication technology provides an alternative solution to this problem by sharing each vehicle's perception information with other nearby vehicles to increase the perception capabilities of all vehicles in the network. The literature shows that as traffic density increases, the local sensors' effective FOV (eFOV) drops quickly due to occlusion. Instead, by sharing perception information, the collective FOV (cFOV) of each vehicle from V2V communication can be significantly increased, even for a small market penetration rate [6]. Therefore, sharing perception is a cost-effective solution to increase environment awareness as more AVs start to join the V2V network.

The technology of sharing perception through the V2V network with other vehicles is referred to as collective perception, collaborative perception, or cooperative perception in the literature [7] [8] [9] [5] [6]. Here for consistency, we call it collective perception (CP) for the rest of the thesis. Automated vehicles with information sharing capability are called connected automated vehicles (CAVs) for this thesis. CAVs can potentially have a much larger perception range than non-connected AVs with a large communication range, and they suffer less from occlusion, which enables early detection of unconnected road users (URU), such as pedestrians and unconnected vehicles (UV) and gives more

reaction time for the decision module to handle. Therefore, our research hypothesis is that CP can potentially improve safety for AVs.

1.2 Objective

The goal of this research is to demonstrate the safety benefits provided by CP to CAV compared with UVs. To do that, we conduct simulation-based experiments to get the safety measurements according to the RAND Corporation [10]. We designed three simple scenarios where the detection of URU is challenging for local sensors due to occlusion and sensor range limit. However, with the help of a nearby CAV, the potential obstacle state can be received through V2V communication and gives the ego vehicle enough time to take action and improve safety.

For our implementation of CP, we use high-level fusion and assume no communication delay. We use pose graph optimization (PGO) to fuse the local sensory data and the data coming from the V2V network and reduce the state estimation uncertainty. After that, a simple constant velocity predictor is applied to predict the future trajectory of the URUs and check for possible collision between the ego vehicle and URU. An emergency stop maneuver is issued based on the detection of a future collision. We implement these scenarios using the Carla simulator [11] and measure the safety by time-to-collision metrics. See Figure 1.1 for an illustration of our concept.

Our contributions for this work are summarized in the following:

- We designed three custom scenarios that exposed the limitation of local sensory data, i.e. occlusion and limited perception range. These limitations can potentially cause traffic accidents.
- We proposed a sensor fusion framework based on PGO to fuse information from

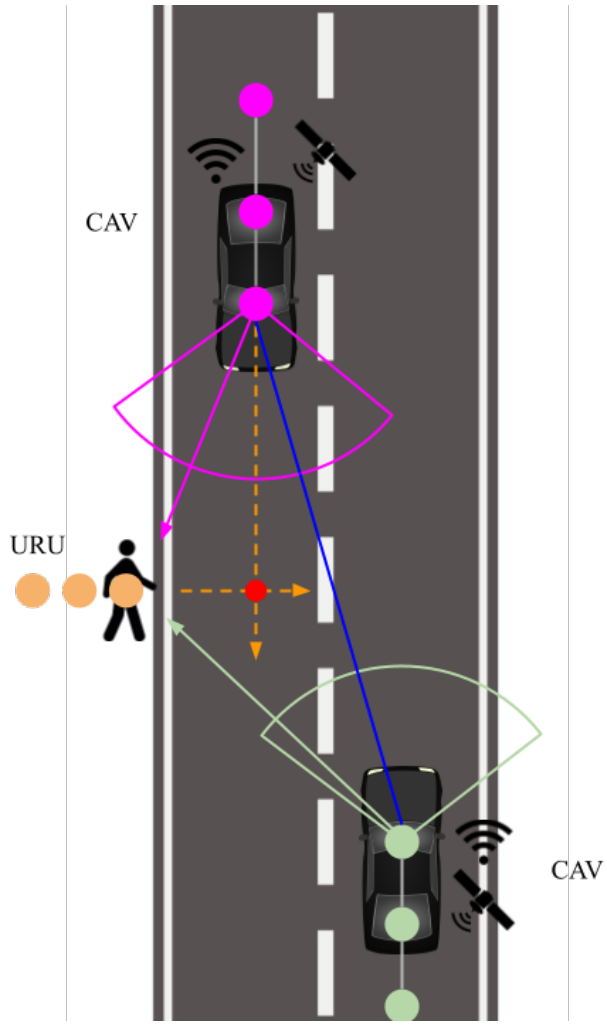


Figure 1.1: Illustration of Collective Perception. In the cartoon, two CAVs are GPS enabled, and they can share data with each other through V2V communication. Both vehicles are perceiving the same URU that falls in the FOV and perception range; at the same time, indicated by the purple arrow and green arrow, which denotes an indirect encounter. The two CAVs can also perceive each other, as shown in the blue line segment, which denotes a direct encounter. The CAVs also have odometry sensors to constrain their consecutive poses, as shown in gray lines. The fusion process estimates the surrounding state by considering all sensor measurements coming from all connected CAVs, taking advantage of observation redundancy. Future trajectories between CAV and URU are predicted (shown in orange dashed arrows) based on the state estimation to detect possible hazards and improve safety.

local sensory data and information from the V2V network.

- We demonstrate that it is possible to reduce the state estimation uncertainty and improve the safety of AVs using CP.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we discuss the background and related works. In Chapter 3, we talk about the details of our methodology and the experiment design. In Chapter 4, we demonstrate the results and interpret the data. In Chapter 5, we conclude our work based on the results and talk about possible future works.

Chapter 2

Background

In this chapter, we review the literature to provide background information for this research. First, we introduce the background of CP in Section 2.1. Then we briefly talk about Simultaneous Localization and Mapping (SLAM) in Section 2.2, as it forms the basis of our CP sensor fusion method. After that, the background for safety evaluation is discussed in Section 2.3. Finally, the simulators related to our experiment are discussed in Section 2.4.

2.1 Collective Perception

In the earlier project Ko-PER [12], the Cooperative Awareness Message (CAM) and Cooperative Perception Message (CPM) were defined for each connected vehicle and road infrastructure to communicate sensor information. The CAM contains the state of the CAV, including the position, orientation, speed, the state of headlights, the actuation of brakes, etc. The CPM, on the other hand, includes the information of the dynamic objects perceived by the CAV or connected road infrastructure. ETSI is working on a standardization of CP service, including CAM and CPM [7].

Perception data coming from multiple observers with uncertainty needs to be fused and transformed into a fixed coordinate frame. If the same object is observed by multiple observers, data redundancy also becomes an issue since each observation is

corrupted by the accumulated uncertainty with the noise of the sensor and the noise introduced by the observer's own state estimation. Therefore, to fully utilize CPM and CAM, an effective data fusion method is needed.

In the work proposed by Rauch et al.[9], a high-level fusion architecture for merging CPM and CAM is developed, where the object lists received from other connected vehicles are transformed into the local coordinate system, and the temporal alignment of the fusion is addressed by predicting the vehicle state to the current time using an augmented state unscented Kalman filter with a constant turn rate and acceleration motion model. Two approaches were tested for spatial alignment: linearizing the transformation function and unscented transformation (UT). The author shows that UT has a superior performance in reducing spatial uncertainty.

Gunther et al. [13] proposed environmental perception message (EPM) and a high-level fusion framework, where the EPM contains the information from local perception sensors and is shared through the V2X network with other CAVs. The author then demonstrated a collision avoidance application between two CAVs, with safety measured by the time-to-collision (TTC) metric. However, in their application, the vehicles were driven by human drivers, and the scenario involves only static obstacle avoidance.

More recently, Godoy et al. proposed a grid-based fusion framework for CPM, where the environment is modelled as an occupancy grid, and sensor measurements from CAVs are fused in the occupancy grid with uncertainties. After that, a particle filter is used to evolute the occupancy grid for object tracking [14]. Yuan et al. proposed FPV-RCNN, a keypoint-based deep learning fusion framework for CPM, which involves bounding box matching and keypoint selection for compressing CPM and addressing data association [15]. In the work published by Yoon et al., a

covariance intersection-based fusion scheme is proposed that not only considers the local FOV but also accounts for each vehicle's own state estimation. Besides, the Bhattacharyya distance filter is used for solving data association, and eventually, a single state estimation is derived for each vehicle [16].

In recent work, an Intelligent Road Side Unit (IRSU) and a CAV platform has been developed and demonstrated for safety. In their design, the IRSU with cameras and LiDAR is used to monitor road traffic conditions and broadcast the CP services to the surrounding CAV to extend the perception capability. The author then demonstrated through three experiments the improved VRU awareness for connected vehicles, and it is possible for IRSU perception as the only source for CAV navigation to interact with pedestrians on the road [5].

As mentioned in chapter 1, Huang et al. defined the FOV for ego-only perception as effective Field-of-view (eFOV) and the FOV for CP as collective Field-of-view (cFOV). The author shows that due to the occlusion of surrounding vehicles, the eFOV drops quickly as the traffic density increases, but the cFOV provides 95% more coverage with only a 30% market penetration rate (MPR) [6].

2.2 SLAM

Previous works in Section 2.1 reviewed multiple CP fusion frameworks. Graph-based Simultaneous Localization and Mapping (SLAM), on the other hand, can be an alternative way to model the CP fusion process, in which the poses of the CAV are modeled as vertices, and the sensor measurements represent edges that constraint the connected poses. Then the goal is to find the pose configurations that are consistent with the measurements as a non-linear least squares problem [17].

In the robotics field, SLAM aims to solve the mapping of the environment and localization of the robot at the same time, allowing robots to perform navigation-related tasks in GPS-denied environments. The map is a representation of the environment model, and it consists of landmarks that can be observed by the sensors of the robot. A comprehensive review of the SLAM problems was published by Cadena et al. [18]. In their work, the typical modern SLAM systems are divided into a front-end, which performs feature extraction and data-association tasks, and a back-end, which takes the information from the front-end and estimates the robot poses and the map by a Maximum a Posterior (MAP). The formulation of the back-end is described below:

Let X be the set of unknown variables, including the robot poses and the locations of the landmarks, and let $Z = \{z_k : k = 1, ..., m\}$ be the set of *independent* sensor measurements. We have a non-linear measurement model $h_k(\cdot)$ that maps a set of variables $X_k \subseteq X$ to a measurement z_k with some random noise ϵ_k:

$$z_k = h_k(X_k) + \epsilon_k \tag{2.1}$$

The goal is to find an optimal set X^* that maximizes the posterior $P(X|Z)$ such that:

$$\begin{aligned} X^* &= \arg\max_X \ \mathrm{p}(X|Z) = \arg\max_X \ \mathrm{p}(Z|X)\mathrm{p}(X) \\ &= \arg\max_X \ \mathrm{p}(X) \prod_{k=1}^{m} \mathrm{p}(z_k|X_k) \\ &= \arg\min_X \ -\log\left(\mathrm{p}(X) \prod_{k=1}^{m} \mathrm{p}(z_k|X_k)\right) \end{aligned} \tag{2.2}$$

The measurement likelihood can be written as:

$$\mathrm{p}(z_k|X_k) \propto \exp\left(-\frac{1}{2}||h_k(X_k) - z_k||^2_{\Omega_k}\right) \tag{2.3}$$

And the prior can be written as:

$$\mathrm{p}(X) \propto \exp\left(-\frac{1}{2}||h_0(X) - z_0||^2_{\Omega_0}\right) \tag{2.4}$$

where Ω_k represents the covariance matrix of the measurement k, assuming the measurement follows a Gaussian noise. Notation $||e||^2_\Omega = e^T\Omega^{-1}e$ denotes the Mahalanobis distance with covariance matrix Ω.

Substitute Equation (2.3) and (2.4) into (2.2), our objective becomes:

$$X^* = \arg\min_X \sum_{k=0}^{m} ||h_k(X_k) - z_k||^2_{\Omega_k} \tag{2.5}$$

subjects to error function:

$$F(X) = \sum_{k=0}^{m} ||h_k(X_k) - z_k||^2_{\Omega_k} \tag{2.6}$$

To solve the non-linear least squares problem (2.5) using standard techniques like Gauss-Newton or Levenberg-Marquardt algorithms, one needs to approximate the error function $F(X)$ by first-order Taylor expansion at an initial guess of X, and update X using an approximated linear solution from the error function [18] [17]. The approximation of the error function assumes that variable X is in the Euclidean space, but it is not true for SLAM, and this can lead to sub-optimal solutions [17]. Kummerle et al. [19] proposed a general framework (G2O) to solve non-linear least squares problems for SLAM, which represents the rotation component of the pose variables in over-parameterized non-Euclidean group SO(2) or SO(3), and defined a non-linear operator to perform pose composition between different representations and avoid the problem of breaking the over-constraint of the graph. G2O also utilizes multiple domain knowledge to accelerate the optimization, such as utilizing the sparsity of the pose graph and optimizing for modern CPU architectures.

For CP scenarios, one needs to consider multiple robots running SLAM in the same environment. Kim et al. proposed a probabilistic formulation of the multi-robot SLAM based on pose graphs, where the robot poses are connected by odometry measurements [20]. A direct encounter occurs when one robot observes another robot; an indirect encounter occurs when two robots observe the same part of the environment; both encounters [21] introduce additional constraints between local pose graphs of the robots. Since the relative pose between robots is unknown at the beginning, Kim et al. introduced anchor nodes that represent the offset of each robot's trajectory w.r.t. a global frame, and the poses are transformed to the global frame using the anchor nodes. This way, the pose graph can be solved using an incremental smoothing and mapping (iSAM) algorithm [22].

Since AVs are commonly equipped with GPS and IMU sensors, which can provide global localization and orientation estimation with some degree of uncertainty, the initialization is not a significant concern for AV. In addition, GPS and IMU data can also be fused into the CP by adding extra edges between the pose vertices and a virtual origin vertex [23]. Combining GPS and IMU data with Graph-based SLAM, where GPS and IMU are integrated into state estimation in long-distance localization, is proved to be robust against visual odometry drift [24].

2.3 Safety Evaluation

The goal of this work is to demonstrate a safety improvement of CP for AV. However, the definition of AV safety is not clear. In this section, we review the literature regarding safety evaluation in AV.

The RAND Corporation defined safety as "the overall ability of a vehicle to operate without harm to passengers or other road users within the roadway ecosystem."

[10]. They stated that the traditional approach for measuring vehicle safety based on exposure rates is problematic because the accumulation of vehicle miles traveled (VMT) is insufficient for AV to draw statistical evidence. They proposed a safety framework consisting of Setting, Stages and Measures. Settings refers to the context in which the AV operates, including simulations, closed courses, and public roads. Stages refers to life stages in which AV measurements can be acquired, including Development, Demonstration, and Deployment. During Development and Demonstration stages, simulated scenarios are developed and improved using data collected from human-driven vehicles, AV on simulation, AV on closed courses, and AV on public road experiences.

The categories of measures mentioned in the RAND Safety Framework begin with Standards, Processes, Procedures and Design of AV, followed by Leading Measures and Lagging Measures [10]. The Leading Measures serve as surrogates for Lagging measures and are not directly associated with safety outcomes; Lagging measures involve the actual crashes and their outcomes. Roadmanship is a Leading Measure that characterizes whether the AV interacts well with other road users without creating hazards. Examples of Roadmanship include near misses or near crashes, rapid acceleration or deceleration, time-to-collision (TTC), etc.

Responsibility Sensitive Safety (RSS) is a safety model proposed by Mobileye. RSS falls into the category of Roadmanship of the Leading Measures. The model formalizes five common rules that involve longitudinal and lateral safety, right-of-way, and limited perception visibility and provides formal proof for verification of these rules [25]. Kim et al. determined the variables of the RSS model for a variable focus function camera, and the safety distance and response time were derived [26]. Khayatian et al. proposed a generic version of the RSS model for CAVs to enable Cooperative Driving, capable of detecting and resolving deadlocks in a decentralized

manner [27].

We will be using the TTC as the main safety measure for our research based on the scenarios we designed. TTC is defined as the time until a collision between a leading and a following vehicle if the speed of the following vehicle is faster than the speed of the leading vehicle [28] [29].

For a 2D plane, there is no concept of leading and following vehicles, and the collision is based on estimated trajectories between two vehicles. Hou et al. [30] proposed new algorithms for computing TTC in a 2D plane, and for simplicity, we will be using the circle algorithm. The circle algorithm models each vehicle's shape as a circle and a collision between two vehicles can be detected if the distance between the centers is smaller than the sum of the radii, assuming two vehicles are moving on a straight line at constant velocities, as illustrated in Figure 2.1.

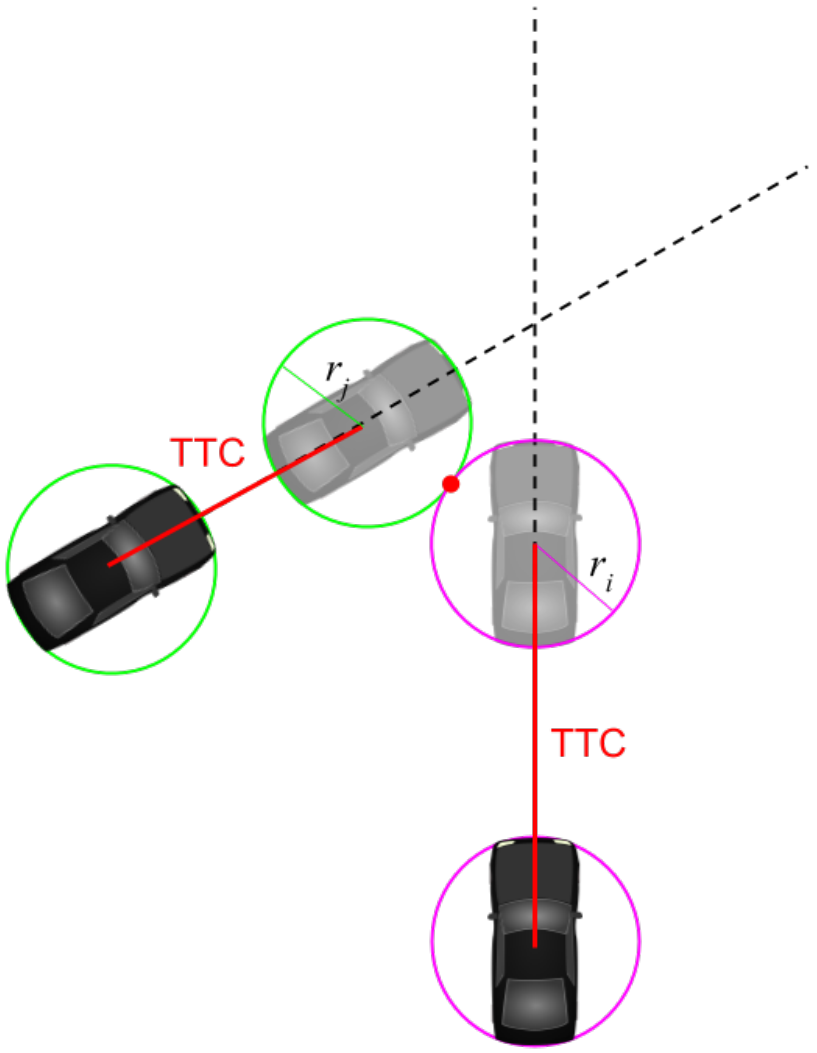


Figure 2.1: The circle algorithm [30] models vehicles as circles, and the collision between two vehicles can be detected if the distance between the centers of two circles equals the sum of their radii. Here, we assume two vehicles are moving in straight lines at constant velocities, so their TTC can be obtained by finding the time at which the two circles intersect.

Given the locations, velocities and lengths of two vehicles, their TTC can be obtained by solving the following equation and taking the smaller root of t:

$$[x_i + v_{xi}t - (x_j + v_{xj}t)]^2 + [y_i + v_{yi}t - (y_j + v_{yj}t)]^2 = (r_i + r_j)^2 \tag{2.7}$$

where

x_i, y_i :location of the object i

x_j, y_j :location of the object j

v_{xi}, v_{yi} :velocity of the object i

v_{xj}, v_{yj} :velocity of the object j

r_i, r_j :radii of the objects i and j

t :TTC between two objects i and j

If there is no solution, the two vehicles will not collide with each other based on the motion assumptions. Note that Equation (2.7) is modified from the original formula in [30], omitting the acceleration terms because the vehicles are moving at constant velocities in our experiment.

2.4 Simulators

Simulation is crucial for evaluating AV performance during the Development and Demonstration stages, without causing actual damage to the public, according to the RAND Safety Framework [10]. In this section, we review some existing simulators and discuss their pros and cons for the use case of CP and safety evaluation.

Simulation of Urban Mobility (SUMO) is a well-known 2D open-source traffic simulator. It is designed for simulating large road networks with different demand models [31]. The benefits for SUMO come with its speed and the design for traffic networks. It is capable of simulating large volumes of vehicles at the same time, supporting flexible network format for building and importing, and providing multiple car-following models and lane-changing models for realistic traffic flow simulation. SUMO also comes with a variety of safety evaluation metrics, including surrogate safety measures like TTC, headway, break times, etc. However, in SUMO, there is no

straightforward way to allow an ADS to manipulate a vehicle using a steering wheel, throttle, brake, gear shifting, etc. In addition, sensors such as camera, LiDAR, radar and IMU are not implemented in SUMO. Therefore AV cannot be easily simulated.

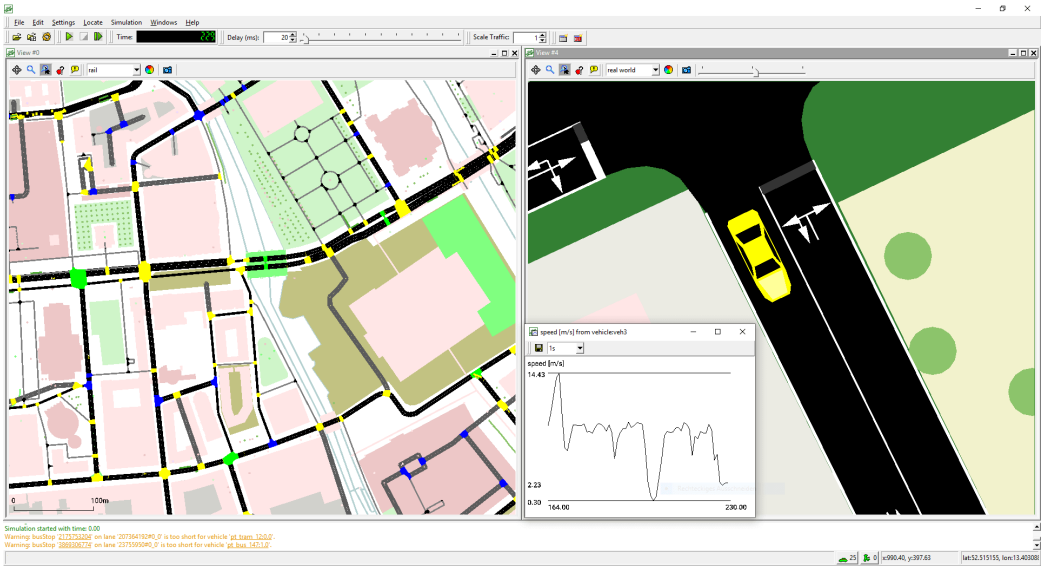


Figure 2.2: A screenshot of the SUMO simulator [31]

Carla is an open-source simulator built on Unreal Engine, and it was designed as a scalable 3D simulation platform for autonomous driving [11]. Carla features a realistic rendering of lights, high-quality sensor rendering and physics computation and provides both Python and C++ API for manipulating vehicles, pedestrians, and virtual world states. Multiple weather conditions and sensor imperfections are simulated, allowing for the development and testing of robust ADSs and Computer Vision algorithms. Carla also supports ground truth generation using semantic sensors and depth sensors, suitable for synthetic datasets for machine learning systems. The drawbacks of Carla in the CP field consist of two aspects. Firstly, the rendering capability of Carla is limited to a single GPU for the server-side, while the computation of the client-side can be distributed to multiple machines due to the one-server-to-multiple-client architecture, having multiple CAVs with cameras and LiDARs simulated at the same virtual world leads to a performance bottleneck. Secondly,

Carla has a built-in TrafficManager for simulating traffic flow, which is not as realistic as SUMO. Therefore, Carla supports using SUMO in Co-simulation mode instead, in which SUMO is responsible for traffic flow simulation, and Carla for rendering the 3D scene and generating sensor data. This solution has been used in existing works such as OpenCDA [32].

For the purpose of this research, we will be using the Carla simulator as our testing environment, which is more than sufficient for our purpose because we only consider a maximum of two CAVs, and do not require precise background traffic flow simulation. Besides, the semantic camera provides a simple way to simulate occlusion in our experiment, which we will be discussing in Chapter 3.

The Carla simulator also comes with an open benchmark called the Carla Autonomous Driving Challenge [33] (the Challenge). The Challenge is developed to evaluate the driving proficiency of ADS under different simulated environment and weather conditions. Participants need to follow the pre-defined routes to drive a vehicle from a starting location to a destination, facing a selection of 10 traffic scenarios defined by NHTSA Pre-crash Scenario Topology [34]. The background traffic flow is generated dynamically using the TrafficManager from Carla, and it is possible to select a group of background vehicles as CAVs to provide useful CP, which allows us to evaluate our ADS in relatively realistic driving environments.

Our preliminary test involved modifying an existing ADS called Pylot [35] to add CP implementation and using the modified Pylot to run the public testing routes in the Challenge and see if there is any safety improvement with CP. We recorded the driving data and developed a data visualizer for inspecting the behaviour of the agent during each route to understand the reason for infractions, see Figure 2.4. However, after the preliminary test, we found that the Challenge is not suitable for

(a)

(b)

Figure 2.3: Carla Autonomous Driving Challenge [33] Screenshots

our purpose. Firstly, the Challenge is not solely designed to evaluate safety. Metrics like route completion are also considered in the overall driving score, but failure to complete a route can happen randomly due to the interference of background traffic and decrease the final score. Besides, actual collisions are involved in the Challenge, making our TTC metric not suitable for measuring safety. Since the environment is dynamic, agents are not expected to encounter the same traffic situations at each episode with the same route. Each episode may be terminated early depending on certain conditions, such as the blocking of an agent by obstacles. These problems make a statistical comparison between agents difficult without a significant amount

Table 2.1: NHTSA Pre-crash Scenarios in the Challenge

Scenario	**Available**	**CP can help**	**Description**
1	Yes	No	Control loss
2	No	No	Lead vehicle deceleration
3	**Yes**	**Yes**	**Unexpected obstacle avoidance**
4	**Yes**	**Yes**	**Expected obstacle avoidance**
5	No	Yes	Lane changing for slow leading vehicle
6	No	Yes	Vehicle passing on opposite lane
7	No	Yes	Crossing intersection with vehicle running red light
8	No	Yes	Unprotected left turn
9	No	Yes	Right turn with crossing traffic
10	No	Yes	Crossing at unsignalized intersection

of driving mileage. By preliminary analysis, some of the scenarios are related to limited perception and can potentially benefit from CP. However, among the 10 pre-defined scenarios claimed by the Challenge, only 3 scenarios are available in the public testing routes, and 2 of them can potentially take advantage of CP. The summary of the scenarios based on our preliminary analysis is shown in Table 2.1.

Figure 2.4: A screenshot of the Visualizer. We developed the visualizer for inspecting the recorded driving data during the Carla Autonomous Driving Challenge. The figure shows a bird's-eye view of the traffic situation centered at the top of the ego vehicle, the two blue line segments for the FOV, and the yellow circle indicates the communication range between the ego vehicle and the surrounding CAVs. Orange dots and lines represent the detected position and predicted future trajectories of the nearby obstacles. Blue dots denote the local waypoints, and gray dots denote the global waypoints. Finally, the connectivity between CAVs is shown in purple.

Chapter 3

Methodology

3.1 Information Fusion Framework

We introduce our CP information fusion framework based on PGO in this section. For simplicity, we only consider the 2D coordinate system. Unlike normal SLAM problems, here, the mapping of the environment is not of our concern. Instead, the unconnected road users (URU, i.e. pedestrians and unconnected vehicles) can be interpreted as "moving landmarks."

We assume that a CAV is equipped with a global localization system such as GPS and IMU to provide its global pose estimation with uncertainty and a camera to detect an object that falls into its Field-of-view (FOV) and detection distance without being occluded by obstacles. We also assume that the odometry measurement of the vehicle is also available between each pair of poses of a CAV. This may be achieved using wheel odometry or visual odometry.

Given the available information, we formulate the CP sensor fusion as pose graph similar to Kim et al. [20], where a vertex represents an unknown variable, including the pose of the CAV and the pose of URU; an edge represents the constraint between two variables. The only difference is that we do not use anchor node [20]; instead, the measurement between each pose and the origin is provided by the GPS and IMU.

In the AV situation, a direct encounter is defined as one CAV detecting another CAV, and an indirect encounter is defined as multiple CAVs detecting the same URU.

We define an unknown variable set X that contains the origin o of the global coordinates and R trajectories of objects of interest (CAVs and URUs):

$$X = \{o\} \bigcup \{x_i^r\}_{i=0,r=0}^{M_r-1,R-1} \tag{3.1}$$

where each object r has a set of trajectory length of size M_r.

The odometry measurement set is defined as:

$$Z = \{u_i^r\}_{i=0,r=0}^{R-2,M_r-1} \tag{3.2}$$

The GPS measurement set is defined as:

$$G = \{g_j^r\}_{j=0,r=0}^{R-1,M_r-1} \tag{3.3}$$

The encounter measurement set is defined as:

$$C = \{c_k\}_{k=0}^{N-1} \tag{3.4}$$

where u_i^r, g_j^r, and c_k represent the odometry, GPS, and encounter measurements respectively, and each measurement is assumed to be independent. The *factor graph* [36] in Figure 3.1 is used to illustrate the relations of the variables and their measurement constraints.

The problem of estimating the CAV trajectories and URU trajectory given the sensor measurements and the pose priors can be defined as a Maximum a Posterior

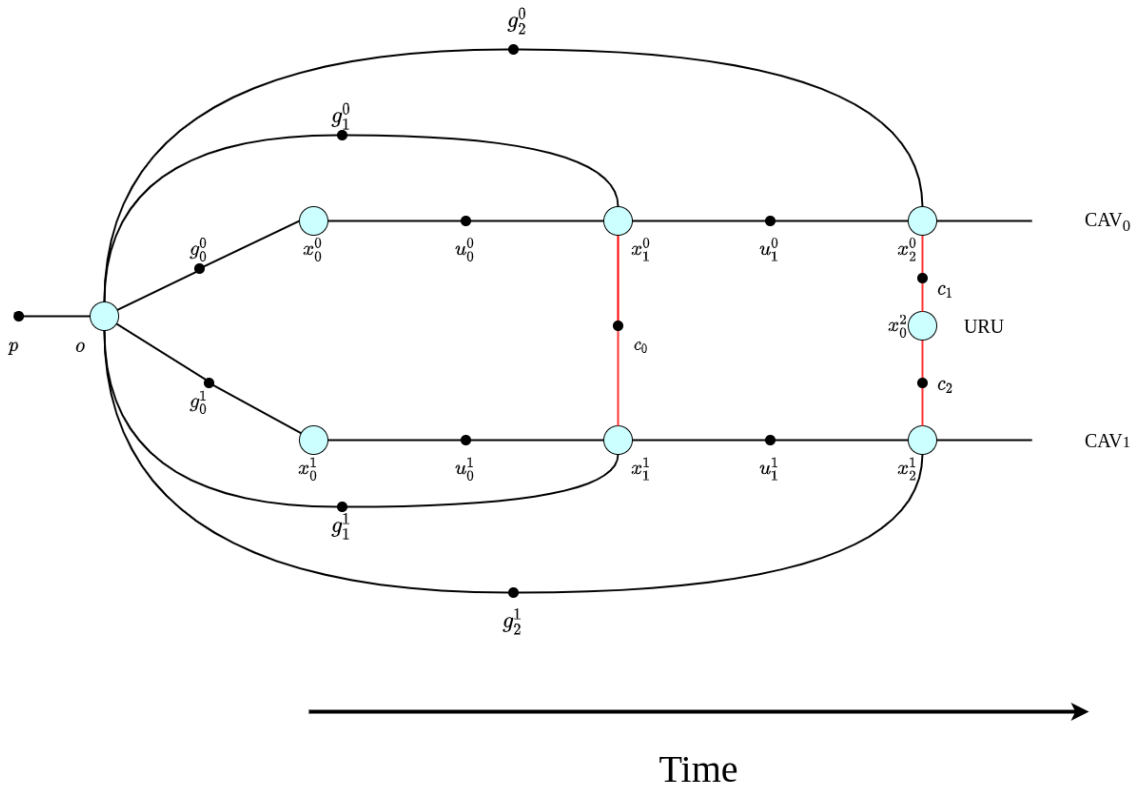


Figure 3.1: Factor Graph illustrating the relation of variables and corresponding measurement constraints in time sequence. The blue circle denotes the unknown variable, and a black dot on an edge denotes the measurement constraint. p is the prior of the origin o, each pose variable is constrained to the origin by GPS and IMU measurement g, two consecutive pose variables are constrained by an odometry measurement u. If an encounter occurs, the two corresponding object poses are constrained by a perception measurement c (red line).

(MAP) problem [18] same as Equation (2.2).

We define our objective function as:

$$
\begin{aligned}
X^* = \arg\min_X & \sum_{u_i^r \in Z,\ x_i^r, x_{i+1}^r \in X} ||x_{i+1}^r \ominus x_i^r - u_i^r||_{\Omega_u}^2 \\
& + \sum_{g_i^r \in G, x_i^r \in X} ||x_i^r \ominus o - g_i^r||_{\Omega_g}^2 \\
& + \sum_{c_k \in C, X_i^r, X_i^s \in X} ||x_i^s \ominus x_i^r - c_k||_{\Omega_c}^2
\end{aligned} \tag{3.5}
$$

where the notation $\ominus$ is for the inverse pose composition operator [37], and Ω_z, Ω_g and Ω_k are the covariance matrices for odometry, GPS, and camera, respectively. We assume the data association (k, i, r, s) is known.

To solve Equation (3.5), we use the CSparse solver from the G2O framework since the CSparse solver is claimed to be faster for small graphs compared to CHOLMOD and PCG [38] [19]. To keep the graph size small, starting from the second vertex, we add a new vertex and a new edge for each CAV and URU every 10 simulation steps and only keep the 50 most recent vertices for each CAV and URU in the graph. For simplicity, we assume the simulation steps are synchronized for all CAVs, or CAVs will miss their encounters with each other or the URU. Also, we assume that there is no communication delay between CAVs.

3.2 Implementation Approach

After getting the fused perception information using the fusion framework discussed in Section 3.1, the perception information and the state of the vehicle are then passed to the ADS to generate safe steering, throttle, and brake commands for the AV to drive to the destination.

For the ADS, we use the generic modular system, which separates the driving task into several sub-problems, and each sub-problem is solved independently by a specific module [2]. Our system consists of a detection module, localization module, data fusion module, object prediction module, planning module, and control module. Figure 3.2 shows the system diagram of the ADS.

The detection and localization measurements are generated using the ground truth states from the simulator and add independent Gaussian noise to simulate the effect

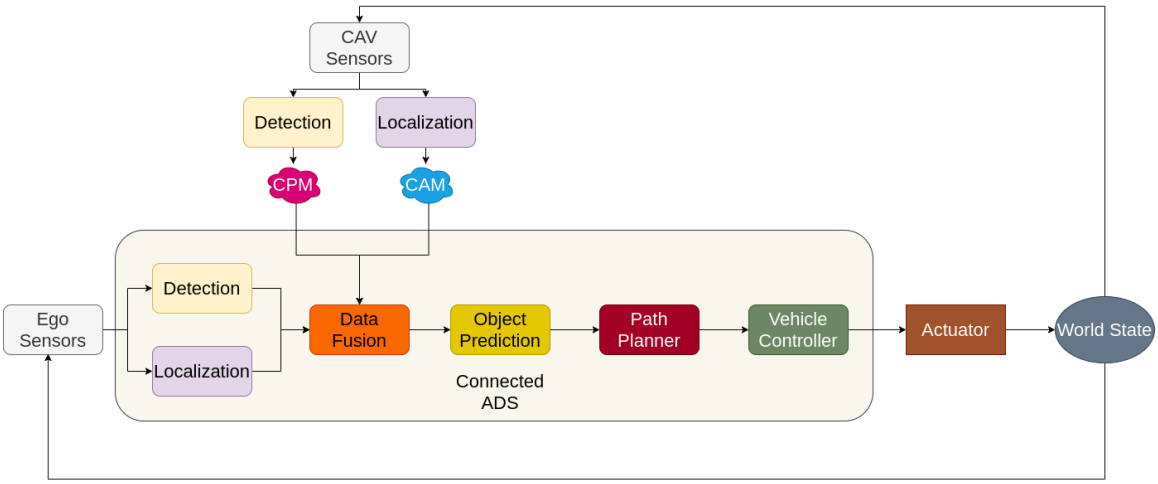


Figure 3.2: System Diagram for our connected ADS. The ADS consists of six sub-modules: detection, localization, data fusion, prediction, planning, and control, and the sub-modules form a computation pipeline, in which the output of one module is passed into the next module, indicated by the arrows. Outside the connected ADS are different components that belong to the simulated world, which provide input sensory data and generated state transition based on the output of the connected ADS. Additional detection and localization information will be provided to the data fusion module in CAM and CPM, in the perspective of CAV.

of sensor uncertainty. The effect of camera occlusion is realized by attaching the semantic camera to the same location and orientation as the imaginary RGB camera of the CAV (the semantic camera is provided by the Carla simulator). When an object is located within the FOV and detection distance, we project the object pose to the camera plane in pixel coordinate and check in the semantic image whether the projected pixel belongs to the same category of the target object. In our experiment, this simple approach is sufficient to simulate occlusion and demonstrate the effectiveness of CP. See Figure 3.3 for a demonstration of this procedure.

Once the states of the CAVs and URUs are fused and optimized, a decision needs to be made for the ego vehicle to avoid any potential hazard outcomes. In the planning module, the decision is based on predicting the future trajectories of the ego vehicle and the URU to a time horizon given their state estimation. If an overlap between the future trajectories is found, then a stop command will be given to the ego vehicle

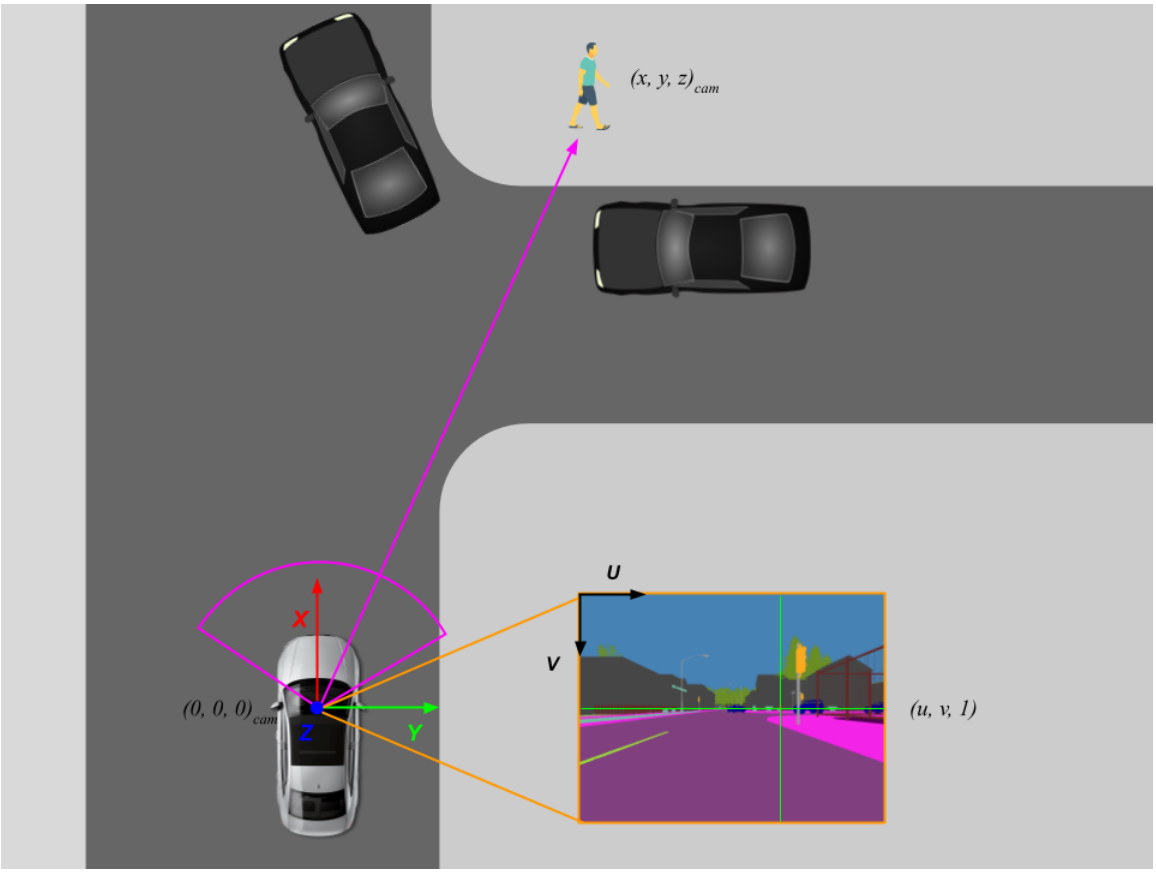


Figure 3.3: Illustration of simulated detection. We do not use real object detection algorithms in our experiment. To simulate the effect of occlusion, we attach a semantic camera to the same location as the imaginary camera for detection. Both the semantic camera and the detection camera have the same intrinsic matrix $\mathbf{K}$. We project the pose of the target on the camera pixel coordinate $x(u, v, 1)^T = \mathbf{K}(y, z, x)^T$, and check whether the pixel in the semantic image belongs to the same category of the target.

to prevent potential traffic accidents; otherwise, the ego vehicle will proceed with its original path.

The Carla simulator has several virtual towns that can be used as testing environments. Each town has a set of pre-defined waypoints of the roads for vehicles to navigate. The path from a starting point to a destination can be easily generated using the high-level GlobalRoutePlanner from Carla API. In the control module, we use PID controllers to control the steering wheel, throttle, and brake of vehicles to follow a set of waypoints. The incoming waypoints can be used directly as the future trajectory of

the ego vehicle. For URU, we use a simple constant velocity predictor in the prediction module, which is sufficient because in our experiment URUs are pre-configured to move in a straight line at a constant speed for simplicity.

3.3 Experiment Design

As mentioned before, the AV local detection sensor is limited in the maximum detection range and occlusion. Since we have not found a public benchmark that suits our purpose, we built three custom scenarios based on a subset of the NHTSA pre-crash topology [34] as our benchmark. The selected scenarios are all vision obscured-related, so they can possibly have been affected by extended perception capability to improve safety. For each custom scenario, we add an additional CAV as a CP provider. We test each custom scenario independently to demonstrate the safety benefits.

Custom Scenario 1 Pedestrian crash *with* prior vehicle maneuver.

> The ego vehicle (indicated by the pink dashed arrow) performs a maneuver (right turn) in a T-junction from the right-hand side, while a pedestrian (path shown by the orange dashed arrow) located in the blinded area is pre-configured to walk across the road when the ego vehicle approaches. The future trajectories between the ego vehicle and the pedestrian are intersected, indicating that the pedestrian will be hit if the ego vehicle does not stop at a sufficient distance. At the same time, a CAV (path shown by the green dashed arrow) is driving from the left-hand side, capable of perceiving the pedestrian's location and informing the ego vehicle if their distance is within the communication range. As shown in Figure 3.4.

Custom Scenario 2 Pedestrian crash *without* prior vehicle maneuver.

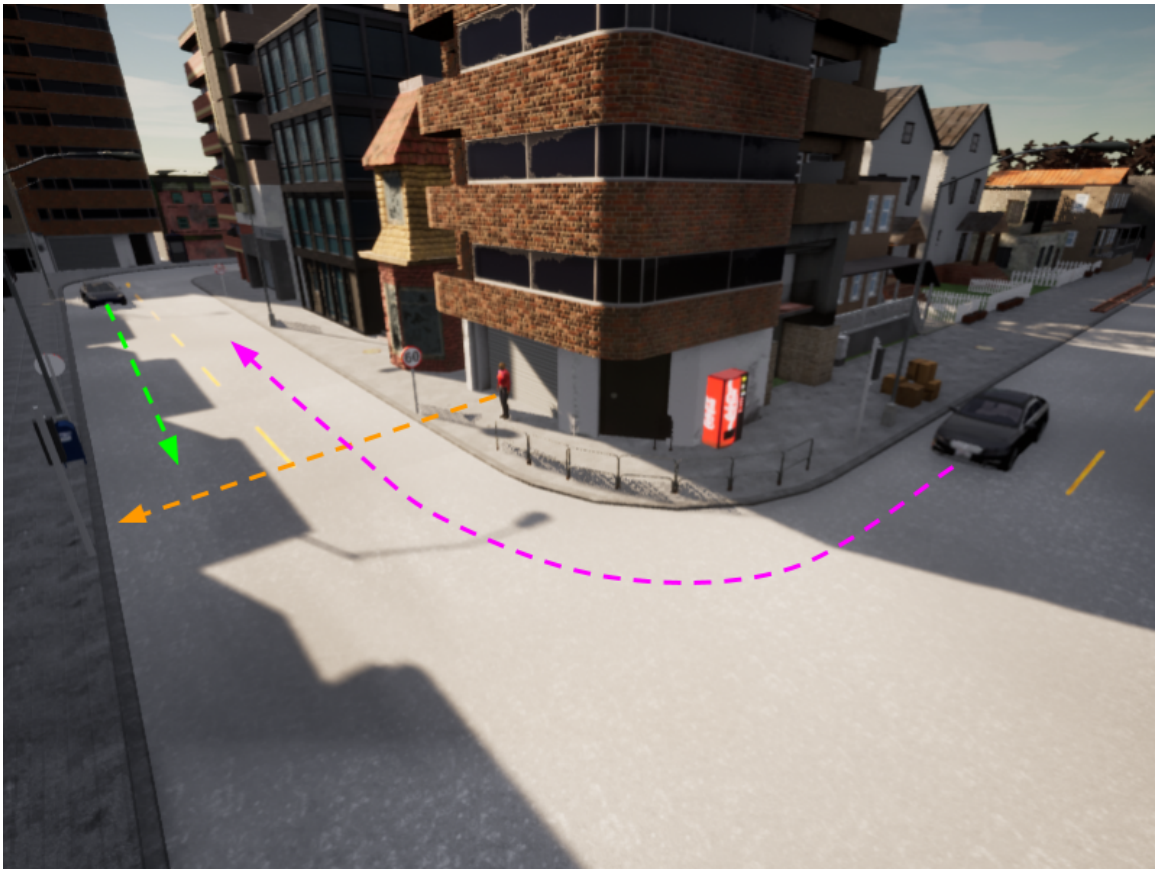
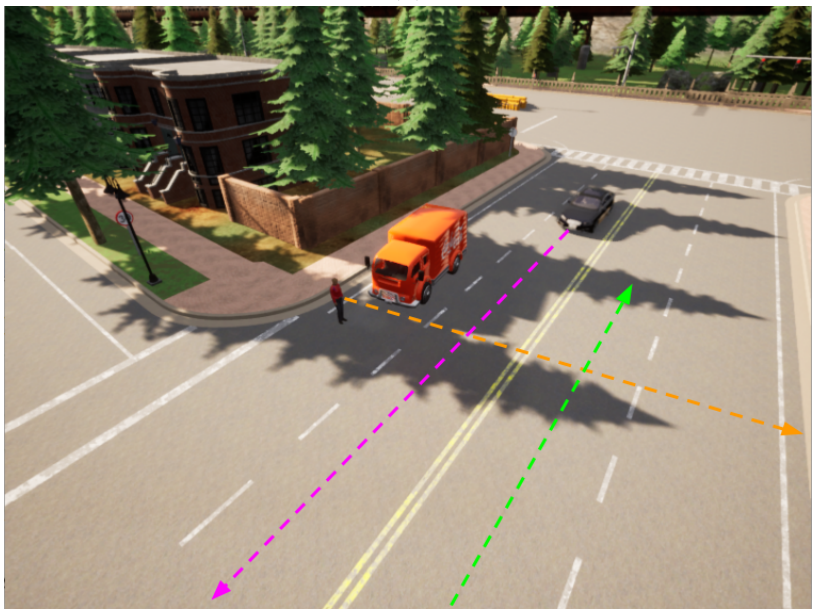


Figure 3.4: Custom Scenario 1. On the right-hand side, the black car is the ego vehicle operated by our connected ADS. The dashed arrows in pink, orange and green represent the moving trajectories of the ego vehicle, pedestrian, and the CAV, respectively. When the ego vehicle is doing a right turn, the pedestrian will start crossing the road, which leads to the potential accident that the ego vehicle needs to avoid. The ego vehicle will come to a full brake upon detecting the pedestrian. The detection of the pedestrian will happen earlier from the CAV's point of view. Therefore, with the CP from the CAV, the ego vehicle will stop at an earlier time, thus increasing the TTC to the pedestrian.

The scenario is similar to Custom Scenario 1, except that the ego vehicle is driving straight (no maneuver), while a pedestrian is occluded by a truck instead of a wall. The pedestrian will start walking across the road when the ego vehicle approaches at a certain distance so that both will collide if the ego vehicle does not brake at a sufficient distance. A CAV is approaching in the opposite direction, detecting the pedestrian's location and informing the ego vehicle if they are within the communication range, as shown in Figure 3.5.

(a)



(b)

Figure 3.5: Custom Scenario 2 (in different views). The dashed arrows in pink, orange, and green represent the moving trajectories of the ego vehicle, pedestrian, and the CAV, respectively. The pedestrian will start to cross the road if the ego vehicle is close enough, leading to a potential traffic accident. The ego vehicle will brake immediately upon the detection of the pedestrian. In Figure 3.5a, the pedestrian is invisible to the ego vehicle due to occlusion but is visible from the CAV's point of view. Therefore, with CP from the CAV, the ego vehicle will stop at an earlier time than using a local sensor only. This will increase the TTC between the ego vehicle and the pedestrian.

Custom Scenario 3 Left turn across path from opposite directions at signalized junctions.

The scenario is intended to demonstrate an object that is far beyond the maximum perception distance of the local sensor. In this scenario, an unconnected vehicle (UV, indicated by the orange dashed arrow) is driving from the far end of the road at a constant speed of 100 km/h. The ego vehicle is approaching the intersection at an initial speed of 50 km/h and attempting to make a left turn while being prepared to yield to the UV because the UV will enter the intersection at roughly the same time. The initial distance between UV and ego vehicle at the beginning is around 120 meters. There is no occlusion between the ego vehicle and the UV, but due to the limited perception range, the UV will not be perceived by the local sensor from the ego vehicle until they are close enough. At the same time, a CAV is driving in the right lane 45 meters ahead of the ego vehicle, and it is capable of perceiving the UV earlier than the ego vehicle, thus can alert the danger of the fast driving UV approaching. As shown in Figure 3.6.

For the experiment, we conduct a comparative study between the baseline and the connected ADS using the method discussed in Section 3.2. The baseline is implemented in the same way as the connected ADS, except that the communication with the CAV is disabled. For each custom scenario, we use the baseline and the connected ADS to drive the ego vehicle, respectively and repeat the simulation 30 times for each method. The TTC value is computed at each simulation step using the circle algorithm discussed in Section 2.3.

For statistical comparison between simulation episodes, we use two aggregate indicators: Time Exposed Time-to-collision (TET) and Time Integrated Time-to-collision (TIT) [29] [39]. TET is defined for object r as follows:

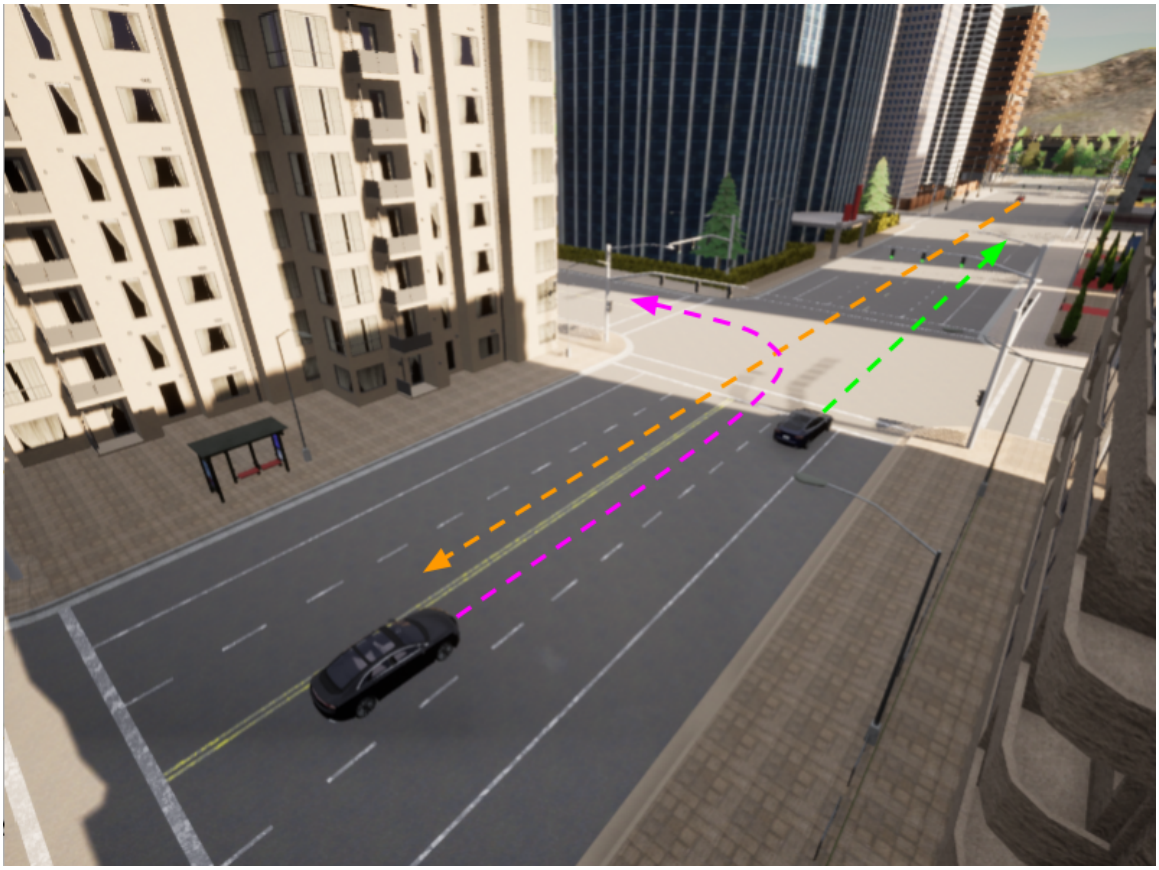


Figure 3.6: Custom Scenario 3. The dashed arrows in pink, orange, and green represent the moving trajectories of the ego vehicle, UV, and the CAV, respectively. The UV is driving at a constant velocity of 100 km/h. The ego vehicle has an initial velocity of 50 km/h and will attempt to make a left turn at the intersection. Both the UV and the ego vehicle will enter the intersection at roughly the same time, and the ego vehicle needs to stop and yield to the UV if the UV is detected. The stopping location of the ego vehicle depends on the timing of detection. In the meantime, a CAV driving on the right lane ahead of the ego vehicle can provide an extended detection range, allowing early detection of the UV and increased TTC.

$$\mathrm{TET}_r = \sum_{t=0}^{T} \delta_r(t) \cdot \Delta k, \tag{3.6}$$

$$\delta_r(t) = \begin{cases} 1 & \forall\ 0 < \mathrm{TTC}_r(t) \leq \mathrm{TTC}^*, \\ 0 & \text{otherwise} \end{cases}$$

where ΔK denotes the time duration for two simulation steps. In our case, it is set to 0.01 since we are using a constant interval of 0.01 seconds for simulation steps. The TTC* denotes the threshold that distinguishes safe and unsafe conditions between

the objects, and we choose the value of 2.0 seconds according to Zhang et al. [29]. Similarly, TIT is defined for object r as follows:

$$\begin{aligned} \mathrm{TIT}_r = & \sum_{t=0}^{T} \left[\frac{1}{\mathrm{TTC}_r(t)} - \frac{1}{\mathrm{TTC}^*} \right] \cdot \Delta k, \\ & \forall\ 0 < \mathrm{TTC}_r(t) \leq \mathrm{TTC}^* \end{aligned} \tag{3.7}$$

The unit of both metrics is seconds. TET aggregates the total time duration that is exposed in unsafe situations determined by the TTC* threshold, whereas TIT aggregates levels of unsafe situations during the simulation as the final indicator. We use both metrics for a more comprehensive comparison, and the mean and the standard deviation for both TET and TIT will be computed for 30 repetitions for our statistical significance test between the baseline and the CP method.

Chapter 4

Result and Interpretation

In this chapter, we present the experimental details, including parameter selection, results, and interpretations.

4.1 Parameter Selection

In the experiment, we set the communication range to 300 meters, as this is the operational range for commonly used DSRC based radios for V2V communication [7]. The maximum perception distance is set to 100 meters [40], and the FOV of the camera is 110 degrees.

We also simulate sensor measurements by adding independent zero-mean Gaussian noise to the GPS, IMU, and odometry sensors. The standard deviation for the GPS sensor is set to $\sigma_{\text{GPS,lon}} = 1.08$ meters and $\sigma_{\text{GPS,lat}} = 0.94$ meters [41]; for the IMU sensor is $\sigma_{\text{IMU},\theta} = 0.0021$ degrees [42]. For visual odometry, the noise standard deviation is set to $\sigma_{\text{Odom},x} = 0.003$ meters, $\sigma_{\text{Odom},y} = 0.007$ meters and $\sigma_{\text{Odom,yaw}} = 0.2$ degrees [43] in the ego vehicle's local coordinate system. To clarify, the coordinate systems for the ego vehicle and the GPS are shown in Figure 4.1.

We do not know the appropriate noise model for the camera detector, so we assume that it is also a zero-mean Gaussian noise, and we study the effect of different standard

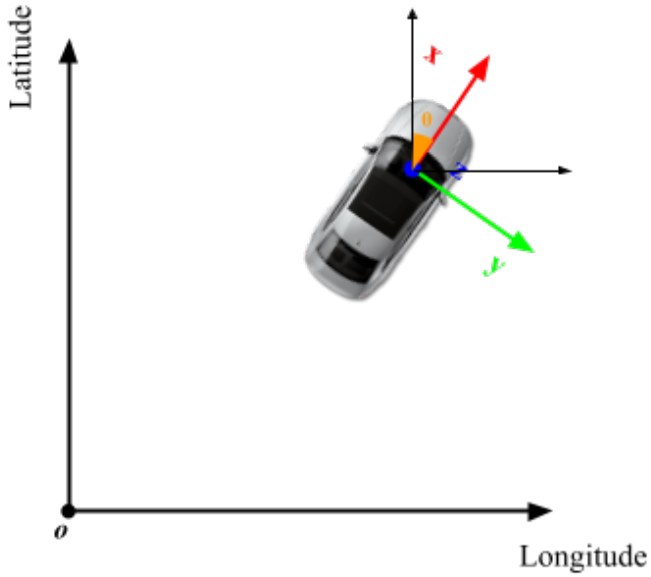


Figure 4.1: Coordinate Systems. The origin o of the GPS coordinate is shown in the figure, and the local coordinate system of the vehicle camera is denoted by the red, green, and blue axes, with a rotation angle θ to the GPS coordinate o.

deviations of perception noise to the state estimation of URU. The metric we used here is Absolute Trajectory Error (ATE_{RMSE}) of the URU in global coordinates [44]. The detection noise standard deviation varies from 0.1 meters to 1.0 meters, and the results are shown in Figure 4.2. Note that here we only consider the URU localization accuracy because our custom scenarios are only designed to avoid collisions between the ego vehicle and URU, and the CAV is only used for providing CP.

Since the baseline uses the raw GPS, IMU, and the camera sensory data directly and does not attempt to reduce uncertainty, transforming the URU from a local sensor coordinate to a global coordinate will lead to accumulated error. The CP approach, on the other hand, fuses the GPS, IMU, odometry, and perception data from both the ego vehicle and CAV and applies equation (2.5) to reduce the uncertainty jointly; thus, it can lead to reduced uncertainty in the URU state estimation. From Figure 4.2, we can see that the CP approach outperforms the baseline by a large margin. However, the

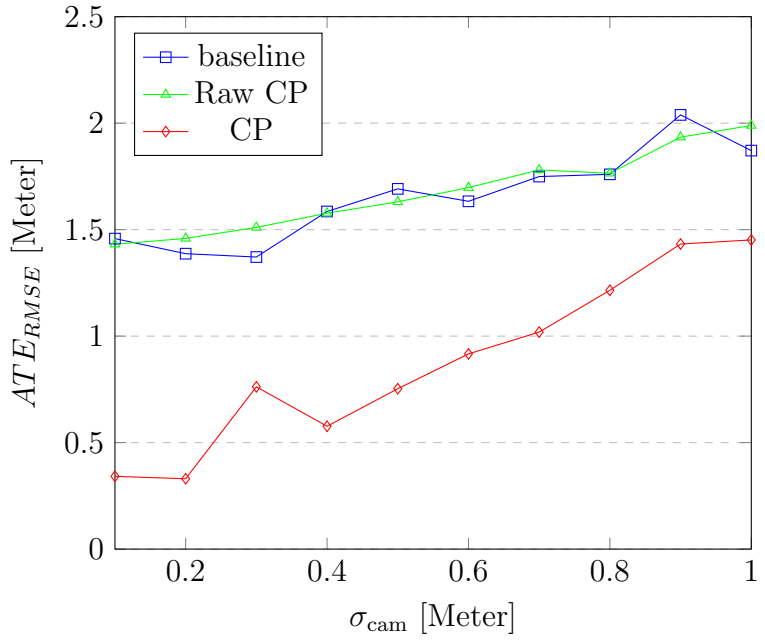


Figure 4.2: URU Localization Accuracy. The figure compares the difference between the baseline, Raw CP (un-fused CP) and CP method in localizing URU with different standard deviations of the camera detector. Raw CP has roughly the same ATE as the baseline, which is expected because both methods do not attempt to reduce localization error. The Raw CP curve shows a smaller variation compared to the baseline because the effect of Raw CP is equivalent to averaging two independent episodes of the baseline. CP outperforms both the baseline and Raw CP by a large margin, but the differences start to diminish as the detection uncertainty increases. This happened because the increased detection uncertainty level reduces the confidence in the corresponding measurement constraint during optimization.

difference between the two approaches gradually diminishes as the σ_{cam} grows. This decrease is because increasing σ_{cam} affects the covariance matrix Ω_k in equation (2.5), making the estimation less constrained by the perception measurement. Eventually, the URU localization accuracy will arrive at roughly the same level as the baseline for the same value of σ_{cam}. Therefore, for the fusion between CAVs to show benefit in increasing localization accuracy, the camera perception uncertainty needs to be reasonably small.

We also conducted an ablation study to see the effect of PGO in reducing the localization error compared to CP using un-fused raw sensory data (Raw CP). The result in Figure 4.2 indicates that CP still outperforms Raw CP, and Raw CP has similar performance as the baseline. This is not surprising because the CAV has the same sensor uncertainty as the ego vehicle, and the effect of Raw CP is equivalent to averaging the localization errors for two independent episodes of the baseline.

4.2 Safety

We choose $\sigma_{\text{cam}} = 0.1$ for evaluating the designed scenarios for fewer false positive alerts of collision that may interfere with normal driving behaviour of the ego vehicle because our approach does not consider false positive filtering. The comparison is made between the baseline, CP, and Raw CP, and in the end, we compute the TET and TIT metrics discussed in Section 3.2.

In all scenarios shown in Figures 4.3, 4.5, and 4.7, the TTC value is not measurable at the beginning of the simulation according to Equation (2.7) because there is no chance that an ego vehicle will collide with the URU based on their states at that time. Starting from a certain time during the simulation, the future paths between the ego vehicle and the URU intersect, so TTC becomes measurable. The decreasing

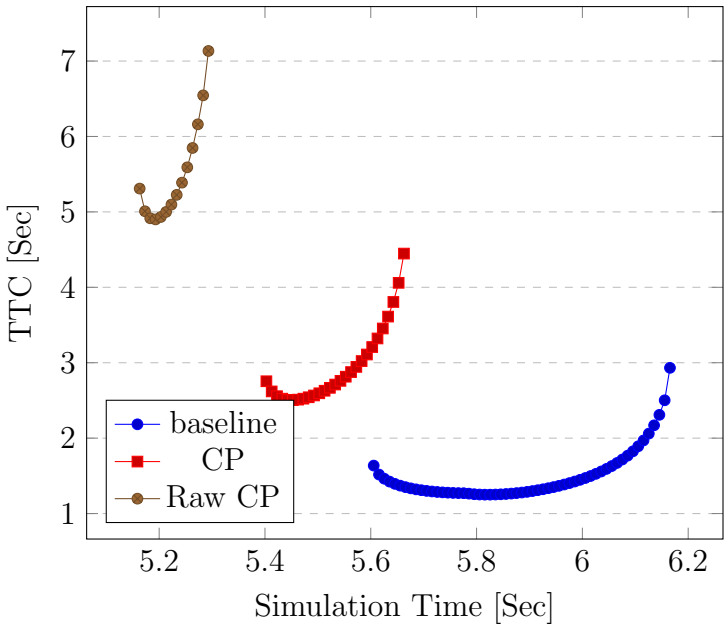


Figure 4.3: Scenario 1 Safety. The Figure shows a comparison of TTC between the ego vehicle and the URU as a result of using different state estimation methods. The baseline here represents the worst situation when the TTC reached its lowest level at around 1.2 seconds. This happened because the URU was not visible due to occlusion at the beginning of the right turn, and there wasn't enough time for the ego vehicle to take action to reduce its speed. CP outperforms the baseline because, for the same situation, the CAV is capable of detecting the URU and informing the ego vehicle to reduce its speed early. The Raw CP has an even better TTC in this case because it did not attempt to optimize the URU pose, and that can lead to higher uncertainty in the URU localization and trigger the emergent brake earlier than CP.

of TTC indicates that the two objects are approaching each other until the ego vehicle realizes the danger and issues the emergent brake maneuver; as a consequence, the TTC starts rising. Eventually, TTC becomes un-measurable as the ego vehicle successfully reduces its speed to zero. Note that in all scenarios, we deliberately pre-configure the settings of the scene so that there is no actual collision involved. Because we use a Leading Measure TTC to measure safety, it makes sense only before the crash happens; see Section 2.3 for an explanation of Leading Measure and Lagging Measures. We do not consider situations after the crash since those will be quantified using Lagging Measures, as discussed in Section 2.3.

In Scenarios 1 and 2, the CP approach always stays above the baseline, indicating that CP is safer for the ego vehicle and the URU than the baseline because, with CP, the ego vehicle can perceive the URU when the local sensor is occluded by obstacles (walls for scenario 1 and the truck for scenario 2), which gives extra reaction time for the ego vehicle to make emergent stop maneuver and thus can reduce the speed of the vehicle to prevent a collision.

In scenario 3, the TTC curve for the baseline is close to linear and drops at roughly a constant rate because the URU is approaching the ego vehicle at the constant speed when the ego vehicle is stopping at the intersection. After the ego vehicle completely stops, the TTC quickly becomes un-measurable at around 2.9 seconds. Note that there is no TTC curve for CP in this scenario because the ego vehicle stopped with its heading parallel to the trajectory of the URU (See Figure 4.8b), and there was no intersection between the predicted trajectories of both vehicles, indicating that there is no chance a collision could occur if both vehicles continue following their current trajectories. Although there is no occlusion between the ego vehicle and the URU in scenario 3, detection of the URU is still not possible since the URU is located beyond the maximum perception range. When the URU is perceived, the driving speed of

the URU is too fast, and there is very little time for the ego vehicle to brake. On the other hand, with the help of the CAV ahead, the URU can be perceived earlier, allowing extra time for the ego vehicle to reduce its speed. Therefore, in scenario 3, CP still outperforms the baseline in terms of TTC.

The ablation study between CP and Raw CP for all three scenarios is conducted. We can see that Raw CP outperforms CP in Figure 4.3, and in Figure 4.5, Raw CP has roughly the same performance. In Figure 4.7, the TTC is not measurable for Raw CP because the vehicle stopped at the intersection with its heading parallel to the trajectory of the URU, which is the same as CP. The fact that Raw CP has a worse localization accuracy than CP but sometimes shows better TTC curves than CP is contradictory, and it may reflect that our experiment design is not comprehensive. Specifically, the emergent stop maneuver could be triggered much earlier than it should due to false positive detection in the Raw CP agent, and we do not consider the consequences of this misbehaviour to safety.

For the purpose of this research, we conduct statistical significance tests between the baseline and CP, using the aggregate metrics TET and TIT for all three scenarios. We do not compare the baseline and Raw CP due to time constraints. The means and standard deviations for the metrics are listed in Table 4.1 for 30 independent runs. Because there are significant differences in standard deviations between the baseline and the CP approach, we use the Welch t-test. Our null hypothesis is $\mu_{\text{Baseline}} \leq \mu_{\text{CP}}$, and the alternative hypothesis is $\mu_{\text{Baseline}} > \mu_{\text{CP}}$, we use the significant level 0.05. The Welch T-test statistics, the P-values, and the decisions are listed in Table 4.2.

Note that in the first two scenarios, the standard deviation for CP is larger than the standard deviation of the baseline for both TET and TIT, but for scenario 3 they show the opposite trend. Our explanation is that the URU was detected by

the CAV before a direct encounter occurred in the first two scenarios, and it was the detection from the CAVs that triggered the emergency stop for the ego vehicle. The relative localization for the URU and the ego vehicle has not been refined by the direct encounter, which leads to a larger amount of uncertainty in the timing of issuing the emergency stop maneuver in CP compared to the baseline. For scenario 3, the direct encounter occurred at the beginning of the episode before detecting the URU; therefore, the localization for the URU was more accurate than the baseline, which makes the standard deviations smaller in the CP than the baseline.

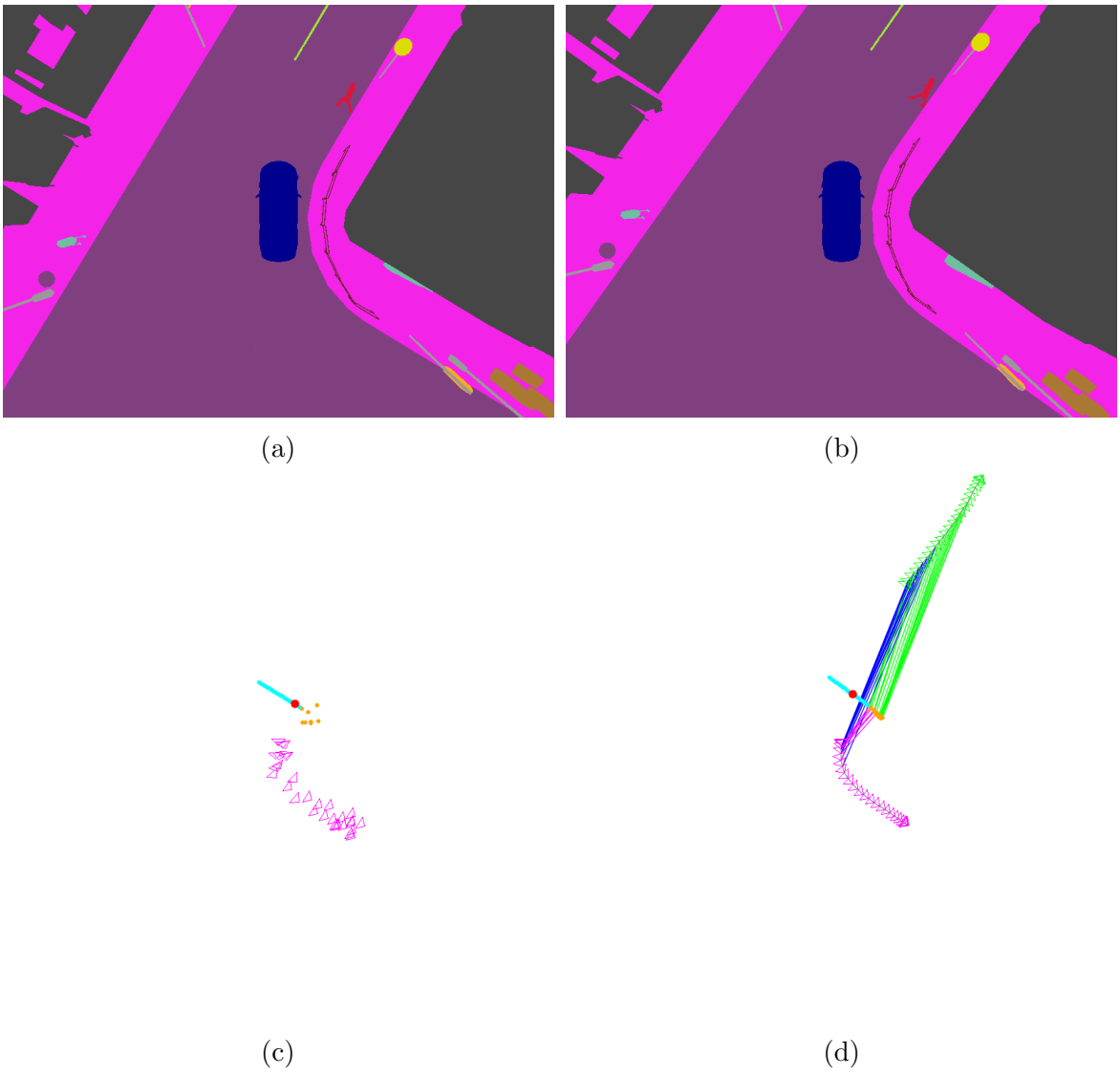
Table 4.1: Safety Aggregate Indicators

Scenarios	TET (Mean/Std) (Sec)		TIT (Mean/Std) (Sec)	
	Baseline	CP	Baseline	CP
1	0.207/0.009	0.071/0.095	0.027/0.002	0.009/0.012
2	0.645/0.027	0.225/0.181	0.191/0.020	0.028/0.033
3	0.802/0.294	0.017/0.090	2.701/3.686	0.002/0.012

Table 4.2: Welch T-test

Scenarios	TET			TIT		
	Test stat	P-value	Decision	Test stat	P-value	Decision
1	7.609	9.761e-9	Reject	7.837	4.132e-9	Reject
2	12.343	1.211e-13	Reject	22.706	1.197e-27	Reject
3	13.758	7.740e-16	Reject	3.942	0.0002	Reject

As shown in Table 4.2, all test results lead to a strong rejection of the null hypothesis with P-values way smaller than our pre-selected significant level. Therefore, we conclude that there is significant evidence that the CP approach is safer than the baseline for the three scenarios we designed, using TET and TIT as evaluation metrics.



(a) (b)

(c) (d)

Figure 4.4: Custom scenario 1. The top figures 4.4a and 4.4b are the bird's-eye views of the ego vehicles using the semantic camera images, where the ego vehicles are the dark blue cars at the center of the images, and the URUs (i.e., the pedestrians) are marked in red. The semantic images show the states where the ego vehicles were completely stopped for the URUs. We can see that in Figure 4.4a, the baseline agent was stopped at a distance slightly closer to the URU than the CP agent in Figure 4.4b, which leads to a higher risk in terms of a smaller value of TTC for the baseline than CP. The bottom two figures, 4.4c and 4.4c, are the corresponding state estimations of the environments for the two methods. Within the bottom two figures, pink triangles and green triangles denote the camera poses of the ego vehicle and the CAV, respectively, and the pink lines and the green lines denote the detection measurement with respect to each camera. Blue lines show the direct encounter between the ego vehicle and the CAV. Gray lines connect the two cameras' poses for the odometry measurements. Orange dots show the URU observations. The Cyan line denotes the predicted trajectory of the URU. Finally, the red circle denotes the predicted collision location.

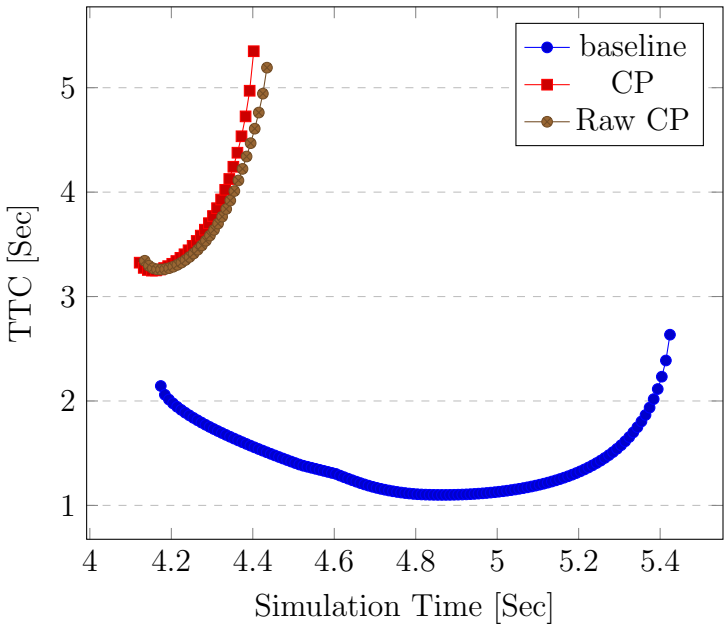
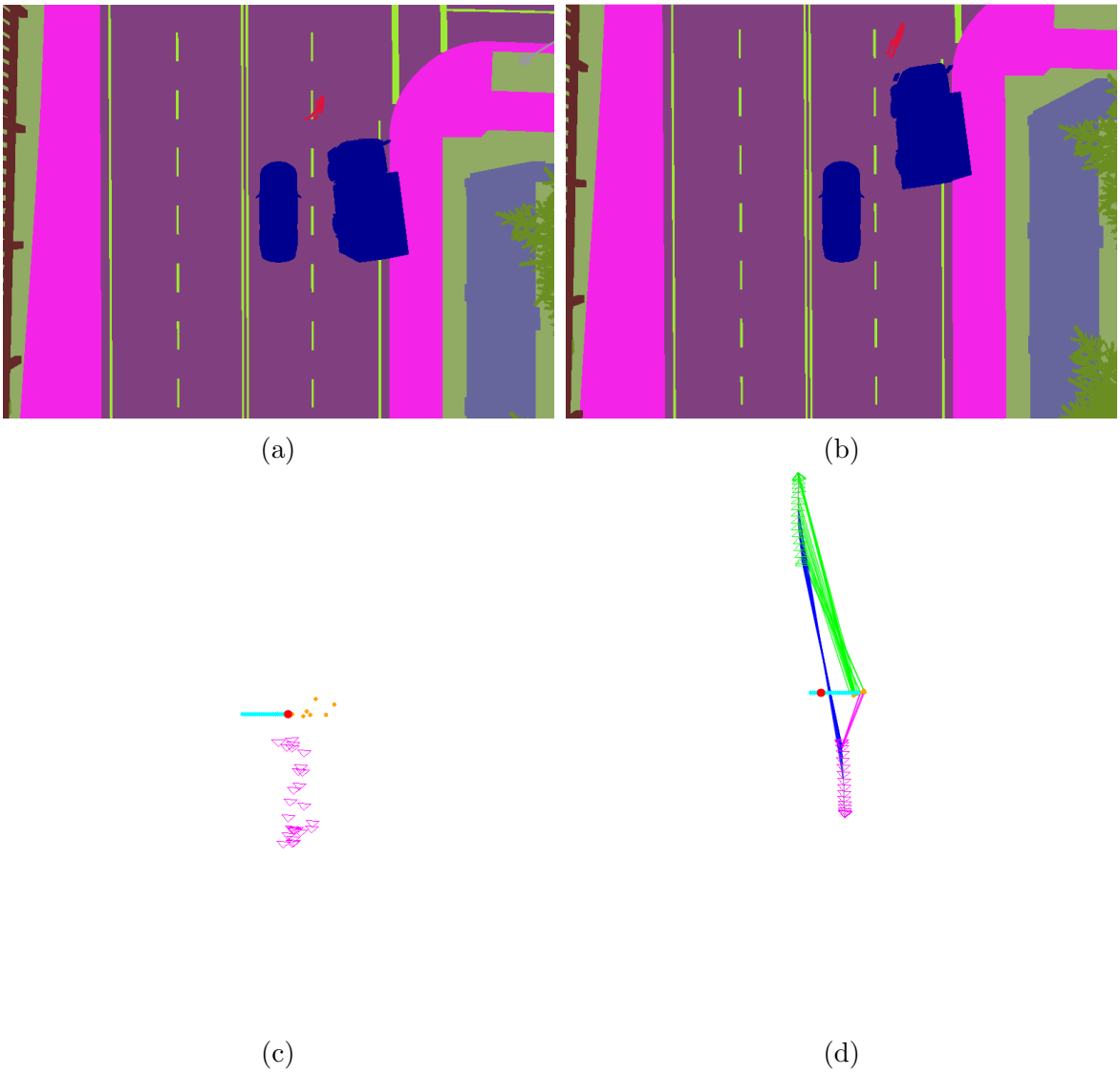


Figure 4.5: Scenario 2 Safety. The Figure shows a comparison of TTC between ego vehicle and URU as a result of using different state estimation methods. The baseline here represents the worst situation when the TTC reached its lowest level at around 1.1 seconds. Because the URU is occluded by the red truck as the ego vehicle is passing through, as shown in Figure 3.5, and when the URU is visible, the ego vehicle does not have sufficient braking time. Both CP and Raw CP have roughly the same values in TTC, and both of them outperform the baseline because, for the same situation, there was a CAV capable of detecting the URU, and informing the ego vehicle to reduce its speed early.



(a) (b)

(c) (d)

Figure 4.6: Custom scenario 2. The top figures 4.6a and 4.6b are the bird's-eye views of the ego vehicles using the semantic camera images, where the ego vehicles are the dark blue cars at the center of the images, and the URUs (i.e., the pedestrians) are marked in red. The dark blue trucks beside the ego vehicles are static and only used to block the detection of the URUs in the ego vehicle's points of view. We can see that in Figure 4.6a, the baseline agent stopped at a distance much closer to the URU than the CP agent in Figure 4.6b. This leads to a higher risk in terms of a smaller value of TTC for the baseline than CP. The bottom two figures, 4.6c and 4.6c, are the corresponding state estimations of the environments of the two methods. Within the bottom two figures, pink triangles and green triangles denote the camera poses of the ego vehicle and the CAV, respectively, and the pink lines and green lines for the detection measurement with respect to each camera. Blue lines show the direct encounter between the ego vehicle and the CAV. Gray lines connect the two cameras' poses for the odometry measurements. Orange dots indicate the URU observations. The cyan line denotes the predicted trajectory of the URU. Finally, the red circle denotes the predicted collision location.

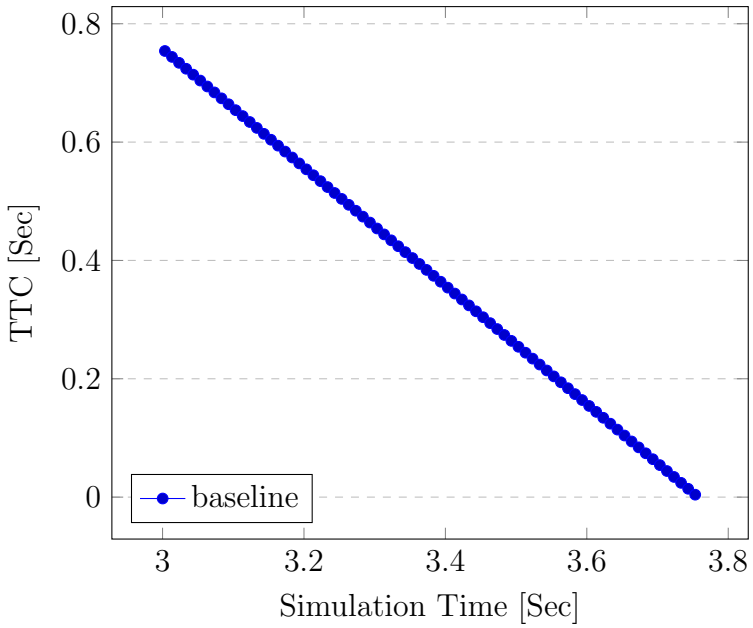


Figure 4.7: Scenario 3 Safety. The Figure shows the TTC curve of the baseline in scenario 3. The curve is roughly linear because the URU is approaching to the ego vehicle at a constant speed when the ego vehicle is stopped at the intersection in Figure 3.6. When the URU passes through, there is no chance for a collision between the ego vehicle and the URU. Both the CP agent and the Raw CP agent were able to stop early enough without entering the intersection so that the heading direction of the ego vehicle is parallel to the trajectory of the URU (see Figure 4.8b), and the TTC is not measurable in this situation.

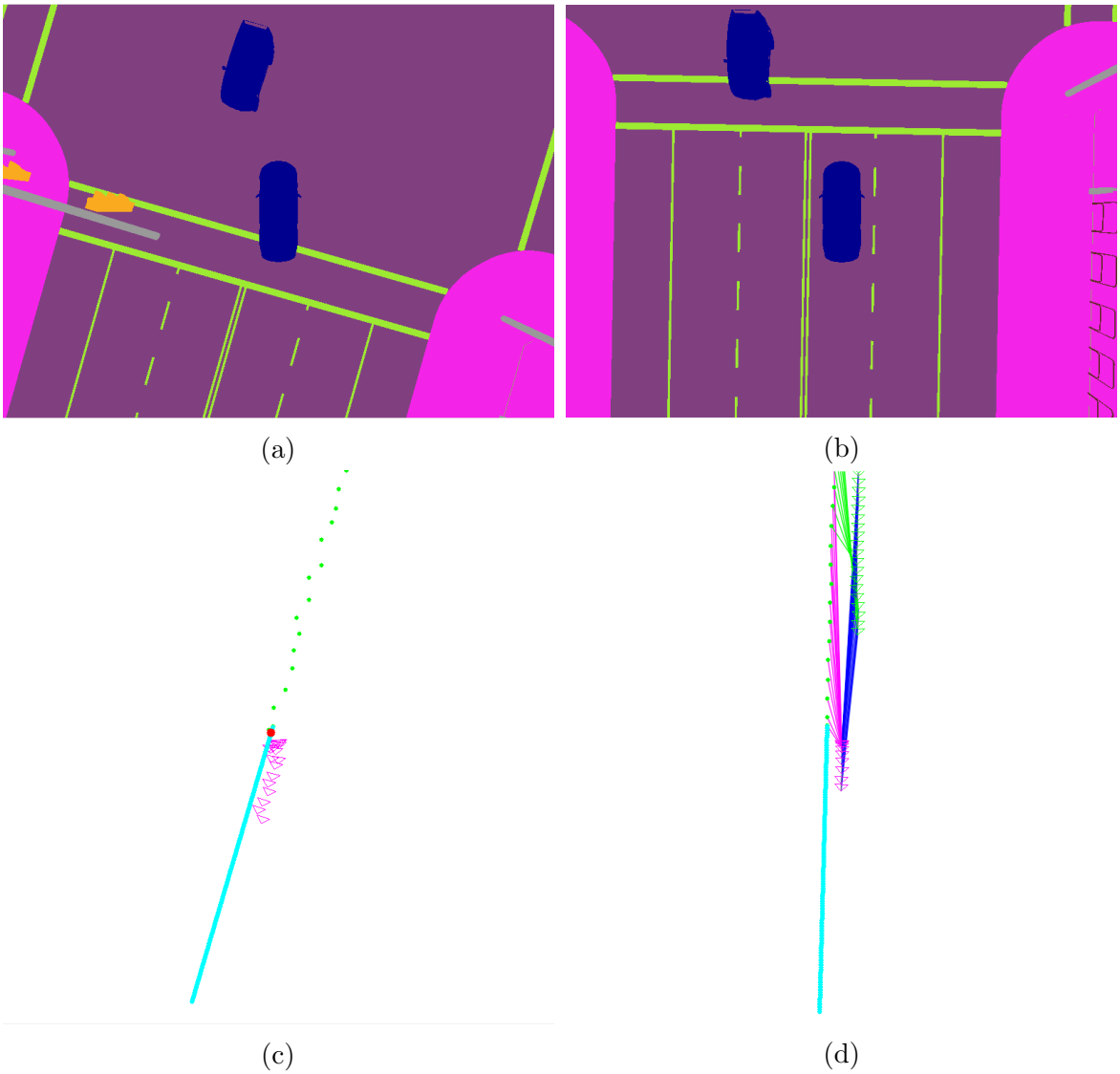


Figure 4.8: Custom scenario 3. The top figures 4.8a and 4.8b are the bird's-eye views of the ego vehicles using the semantic camera images, where the ego vehicles are the dark blue cars at the center of the images, and the NVs are also the dark blue cars in front of the ego vehicles. We can see that in Figure 4.8a, the baseline agent has entered the intersection and stopped at a distance very close to the UV. This is a dangerous behaviour and may lead to a traffic accident. Compared to the baseline, the CP agent did not enter the intersection, as shown in Figure 4.8b. The bottom two figures, 4.8c and 4.8c, are the corresponding state estimations of the environments of the two methods. Within the bottom two figures, pink triangles and green triangles denote the camera poses of the ego vehicle and the CAV, respectively, and the pink lines and green lines denote the detection measurement with respect to each camera. Blue lines show the direct encounter between the ego vehicle and the CAV. Gray lines connect the two camera's poses for the odometry measurements. Orange dots show the URU observations. The cyan line denotes the predicted trajectory of the URU. Finally, the red circle denotes the predicted collision location.

Chapter 5

Conclusion

In this research, we focused on the safety aspect of CP on AV. We formulate the high-level CP sensor fusion as a PGO problem to reduce the sensor uncertainty, and we use the fused information to estimate the state of the ego vehicle and the URU. The potential future collision can be predicted based on the state estimation, and an emergency stop maneuver is generated to avoid a possible hazard between the ego vehicle and the URU. We designed three scenarios where local sensor capability is limited due to occlusion and maximum range to demonstrate the benefits of CP. A baseline agent without CP is compared to the CP agent. Time-to-collision is computed between the ego vehicle and the URU for the baseline and CP, and aggregate safety indicators TET and TIT are also computed to conduct statistical significance tests. Both TTC and the Welch T-test show evidence that CP outperforms the baseline for all three scenarios indicating that CP can improve safety based on our custom scenarios, because CP extended the perception range of the ego vehicle, enables early detection of potential hazards for extra reaction time, and reduces the TTC between the ego vehicle and the URU.

There are still limitations in our research. Firstly, our implementation of the ADS is naive in the sense that it does not consider false alarm filtering, and it treats every occurrence of the predicted collision as an absolute signal of danger, even if the

prediction was not reliable. This could lead to overly cautious driving behaviour. In practice, an AV that stops too often during driving could have a very safe TTC with front-end vehicles and pedestrians, but rear-end collisions become a larger threat, and it may cause traffic congestion. Besides, the AV driving behaviour is affected by only a single CAV in our designed scenarios, which is not realistic in evaluating the safety. In future work, we need to consider a more dynamic environment with realistic traffic flow and use a more comprehensive safety evaluation method, such as RSS [25].

The formulation of PGO in high-level CP sensor fusion provides a framework to reduce sensor uncertainty using GPS, IMU, odometry, and the cameras of multiple CAVs all at the same time, making the coordinate transform of detected objects between different CAVs more robust against localization uncertainty. However, the localization performance of this method needs to be studied and compared with other works, such as [8] [14] [15] and [16] mentioned in Section 2.1, to understand the strengths and weaknesses of our proposed method. It is also possible to extend our work by considering unknown data association, the existence of communication delay between CAVs, and using a real camera for detecting objects. These problems can be studied in future research.

Bibliography

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," Tech. Rep., 2015.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[3] D. J. Yeong, G. Velasco-Hernandez, J. Barry, J. Walsh, *et al.*, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[4] E. Marti, M. A. de Miguel, F. Garcia, and J. Perez, "A review of sensor technologies for perception in automated driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[5] M. Shan *et al.*, "Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations," *Sensors*, vol. 21, no. 1, p. 200, 2021.

[6] H. Huang, W. Fang, and H. Li, "Performance modelling of v2v based collective perceptions in connected and autonomous vehicles," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, IEEE, 2019, pp. 356–363.

[7] *Intelligent transport system (its); vehicular communications. basic set of applications; analysis of the collective perception service (cps)*, https://www.etsi.org/deliver/etsi_ts/103300_103399/10330002/02.01.01_60/ts_10330002v020101p.pdf, [Online; accessed 1-December-2021].

[8] A. Rauch, F. Klanner, and K. Dietmayer, "Analysis of v2x communication parameters for the development of a fusion architecture for cooperative perception systems," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, pp. 685–690.

[9] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2x-based perception in a high-level fusion architecture for cooperative perception systems," in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, 2012, pp. 270–275.

[10] L. Fraade-Blanar, M. S. Blumenthal, J. M. Anderson, and N. Kalra, *Measuring automated vehicle safety: Forging a framework*. 2018.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*, PMLR, 2017, pp. 1–16.

[12] *Ko-PER - Cooperative Perception*, http://www.kofas.de/ko-per.html, [Online; accessed 1-December-2021].

[13] H.-J. Günther, B. Mennenga, O. Trauer, R. Riebl, and L. Wolf, "Realizing collective perception in a vehicle," in *2016 IEEE Vehicular Networking Conference (VNC)*, IEEE, 2016, pp. 1–8.

[14] J. Godoy, V. Jiménez, A. Artuñedo, and J. Villagra, "A grid-based framework for collective perception in autonomous vehicles," *Sensors*, vol. 21, no. 3, p. 744, 2021.

[15] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *arXiv preprint arXiv:2109.11615*, 2021.

[16] D. D. Yoon, G. M. N. Ali, and B. Ayalew, "Data association and fusion framework for decentralized multi-vehicle cooperative perception," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, vol. 59216, 2019, V003T01A019.

[17] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[18] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[19] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3607–3613.

[20] B. Kim *et al.*, "Multiple relative pose graphs for robust cooperative mapping," in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 3185–3192.

[21] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.

[22] M. Kaess, A. Ranganathan, and F. Dellaert, "Isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[23] K. Konolige, "Large-scale map-making," in *AAAI*, 2004, pp. 457–463.

[24] J. Rehder, K. Gupta, S. Nuske, and S. Singh, "Global pose estimation with limited gps and long range visual odometry," in *2012 IEEE international conference on robotics and automation*, IEEE, 2012, pp. 627–633.

[25] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[26] M.-J. Kim, S.-H. Yu, T.-H. Kim, J.-U. Kim, and Y.-M. Kim, "On the development of autonomous vehicle safety distance by an rss model based on a variable focus function camera," *Sensors*, vol. 21, no. 20, p. 6733, 2021.

[27] M. Khayatian *et al.*, "Cooperative driving of connected autonomous vehicles using responsibility-sensitive safety (rss) rules," in *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*, 2021, pp. 11–20.

[28] J. C. Hayward, "Near miss determination through use of a scale of danger," 1972.

[29] J. Zhang, K. Wu, M. Cheng, M. Yang, Y. Cheng, and S. Li, "Safety evaluation for connected and autonomous vehicles' exclusive lanes considering penetrate ratios and impact of trucks using surrogate safety measures," *Journal of advanced transportation*, vol. 2020, 2020.

[30] J. Hou, G. F. List, and X. Guo, "New algorithms for computing the time-to-collision in freeway traffic simulation models," *Computational intelligence and neuroscience*, vol. 2014, 2014.

[31] P. A. Lopez *et al.*, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.

[32] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, "Opencda: An open cooperative driving automation framework integrated with co-simulation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 1155–1162.

[33] *Carla Autonomous Driving Challenge*, https://leaderboard.carla.org/, [Online; accessed 14-December-2021].

[34] W. G. Najm, J. D. Smith, M. Yanagisawa, *et al.*, "Pre-crash scenario typology for crash avoidance research," United States. National Highway Traffic Safety Administration, Tech. Rep., 2007.

[35] I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez, and I. Stoica, "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles," *arXiv preprint arXiv:2104.07830*, 2021.

[36] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[37] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous robots*, vol. 4, no. 4, pp. 333–349, 1997.

[38] T. A. Davis, *Direct methods for sparse linear systems*. SIAM, 2006.

[39] M. M. Minderhoud and P. H. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89–97, 2001.

[40] J. Liu and R. Zhang, "Vehicle detection and ranging using two different focal length cameras," *Journal of Sensors*, vol. 2020, 2020.

[41] S. Bhattacharjya, "Effect of sensor errors on autonomous steering control and application of sensor fusion for robust navigation," 2019.

[42] J.-Y. Rau, L.-C. Chen, C.-C. Hsieh, and T.-M. Huang, "Static error budget analysis for a land-based dual-camera mobile mapping system," *Journal of the Chinese Institute of Engineers*, vol. 34, no. 7, pp. 849–862, 2011.

[43] C. Ye and M. Bruch, "A visual odometry method based on the swissranger sr4000," in *Unmanned Systems Technology XII*, International Society for Optics and Photonics, vol. 7692, 2010, p. 76921I.

[44] D. Prokhorov, D. Zhukov, O. Barinova, K. Anton, and A. Vorontsova, "Measuring robustness of visual slam," in *2019 16th International Conference on Machine Vision Applications (MVA)*, IEEE, 2019, pp. 1–6.