



HAL
open science

Dynamic Occupancy Grid based Collision Avoidance in Robotics

Thomas Genevois

► **To cite this version:**

Thomas Genevois. Dynamic Occupancy Grid based Collision Avoidance in Robotics. Robotics [cs.RO]. Université Grenoble - Alpes, 2024. English. NNT: . tel-04639350

HAL Id: tel-04639350

<https://inria.hal.science/tel-04639350>

Submitted on 9 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique

Spécialité : Informatique

Unité de recherche : Centre de recherche Inria de l'Université Grenoble Alpes



Evitement de collisions en robotique, basé sur grilles d'occupation dynamiques

Dynamic Occupancy Grid based Collision Avoidance in Robotics

Présentée par :

Thomas GENEVOIS

Direction de thèse :

Anne SPALANZANI

PROFESSEURE DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Directrice de thèse

Christian LAUGIER

DIRECTEUR DE RECHERCHE EMERITE, CENTRE INRIA UNIVERSITE GRENOBLE ALPES

Co-directeur de thèse

Rapporteurs :

VERONIQUE CHERFAOUI

PROFESSEURE DES UNIVERSITES, UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE

FAWZI NASHASHIBI

DIRECTEUR DE RECHERCHE, CENTRE INRIA DE PARIS

Thèse soutenue publiquement le **19 juin 2024**, devant le jury composé de :

DOMINIQUE VAUFREYDAZ,

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Président

ANNE SPALANZANI,

PROFESSEURE DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Directrice de thèse

CHRISTIAN LAUGIER,

DIRECTEUR DE RECHERCHE EMERITE, CENTRE INRIA UNIVERSITE GRENOBLE ALPES

Co-directeur de thèse

VERONIQUE CHERFAOUI,

PROFESSEURE DES UNIVERSITES, UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE

Rapporteuse

FAWZI NASHASHIBI,

DIRECTEUR DE RECHERCHE, CENTRE INRIA DE PARIS

Rapporteur

SAMIA AINOUIZ,

PROFESSEURE DES UNIVERSITES, INSA ROUEN

Examinatrice



Abstract

As robots and autonomous vehicles are being deployed in multiple new challenging applications, they have to navigate in various complex and unstructured real-world environments (e.g. urban traffic, human-shared spaces). In order to model the environment of a robot for its local planning, object-based representations are the most commonly used because they are computationally lighter and easier to manipulate in closed-forms. Though such representations often fail to model unstructured or uncertain scenes, especially since the surroundings of a robot generally remain largely uncertain because of sensor limitations and partial scene observability. Under these constraints, probabilistic Dynamic Occupancy Grids seem to be a promising model to represent the environment of a robot throughout its navigation software, regardless the complexity and uncertainty of the scene. Yet, while research in grid-based perception has achieved significant progress in the recent years, the use of Dynamic Occupancy Grids in local planning remains circumstantial. For these reasons, this thesis investigates a method to profit from Dynamic Occupancy Grids in local collision avoidance.

Our first contribution is a grid-based collision avoidance framework along with a new Predictive Collision Detector that we propose as an interface between state-of-art grid-based perception and sampling-based planners. Unlike most other approaches, ours operates only on elementary spatial occupancy such that it captures the richness and versatility of modern occupancy grid perception.

Our second contribution is a human and interaction aware collision avoidance solution. It required an extension of the previous Predictive Collision Detector in order to consider class-specific agent prediction models. A pedestrian-specific prediction model has also been introduced, it was inspired by a state-of-art human-vehicle interaction model.

Our third contribution is a new Augmented Reality test framework that operates at sensor level, on LiDAR data. Unlike most other Augmented Reality frameworks that operate at object level, our new approach enabled the experimental validation in Augmented Reality of the proposed collision avoidance solution, as a continuous pipeline from sensor data to the control of the real-world vehicle.

Across this whole work, a particular care has been taken to operate only on sensor data and elementary spatial occupancy, without using the concept of objects, in order to capture the richness and versatility of modern occupancy grid perception. A massive parallelization of the computations since design of the algorithms allowed a frugal implementation that met the real time constraints even on embedded devices. The proposed methods have been deployed on 3 different robots, including a robotic car, and tested in real-world experiments. With this contribution, we hope to facilitate the use of Dynamic Occupancy Grids for collision avoidance and to enhance the connectivity of the state-of-art of perception and navigation.

Keywords: Dynamic Occupancy Grids, collision avoidance, local planning, Augmented Reality, Robotics, autonomous vehicles, GPGPU

Résumé

Comme robots et véhicules autonomes sont désormais déployés dans de nouvelles applications ambitieuses et exigeantes, ils doivent naviguer dans des environnements réels, complexes et non structurés (e.g. trafic urbain, espaces partagés avec des humains). Pour modéliser l'environnement d'un robot pour sa planification locale, ce sont les représentations par objets qui sont beaucoup plus communément utilisées car elles amènent des calculs moins lourds et sont plus simples à manipuler en expressions mathématiques explicites. Mais ces modèles échouent souvent à représenter les scènes non structurées ou incertaines, d'autant que les alentours d'un robot restent généralement très incertains à cause des limites de capteurs et de l'observabilité partielle de la scène. Dans ces conditions, les grilles d'occupation probabilistes et dynamiques semblent fournir une représentation prometteuse pour modéliser l'environnement d'un robot dans l'ensemble de ses logiciels de navigation, de la perception jusqu'à la planification, indépendamment de la complexité et de l'incertitude de la scène. Mais, alors que la recherche en perception basée sur grilles d'occupation a accompli des progrès significatifs au cours des dernières années, l'usage des grilles d'occupation en planification locale reste marginal. Pour ces raisons, la présente thèse recherche comment valoriser les grilles d'occupation dynamiques au sein d'une méthode d'évitement de collisions local.

Notre première contribution est un système d'évitement de collisions basé sur grilles d'occupation. Il s'accompagne d'un nouveau Détecteur de Collisions Prédicatif que nous proposons comme une interface entre les solutions de perception basées sur grilles et les solutions de planification par échantillonnage. A la différence de la plupart des autres approches, la nôtre opère uniquement sur l'occupation spatiale élémentaire afin d'appréhender la richesse et la polyvalence de la perception des grilles d'occupation modernes.

Notre seconde contribution est une solution d'évitement de collisions sensible aux humains et aux interactions. Ceci a nécessité d'élargir le Détecteur de Collisions Prédicatif avec plusieurs modèles de prédiction, chacun étant dédié à une classe d'agents. Ainsi, un modèle de prédiction spécifique aux piétons a été ajouté. Il a été inspiré d'un modèle d'interaction humain-véhicule de l'état de l'art.

Notre troisième contribution est un environnement logiciel de Réalité Augmentée qui agit au niveau des capteurs, sur les données LiDAR. A la différence de la majorité des autres systèmes de Réalité Augmentée qui opèrent au niveau de représentations par objets, notre approche a permis la validation expérimentale en Réalité Augmentée de la solution d'évitement de collisions proposée, dans son entièreté et sa continuité depuis les données des capteurs jusqu'au contrôle du véhicule en monde réel.

Tout au long de ce travail, un soin particulier a été apporté à la seule prise en compte des données des capteurs et de l'occupation spatiale élémentaire, sans jamais utiliser le concept d'objets, ceci pour appréhender la richesse et la polyvalence des grilles d'occupation modernes. Une parallélisation massive des calculs dès la conception des algorithmes a permis une implémentation frugale qui a satisfait les contraintes de temps réel même sur des appareils embarqués. Les méthodes proposées ont été déployées sur 3 robots différents dont une voiture robotisée et testées dans des expérimentations en monde réel. Avec cette contribution, nous espérons faciliter l'usage des grilles d'occupation dynamiques pour l'évitement de collisions et ainsi améliorer la connectivité de l'état de l'art de la perception et de la navigation.

Mots-Clés : grilles d'occupation dynamiques, évitement de collisions, planification locale, réalité augmentée, robotique, véhicules autonomes, GPGPU

Acknowledgment

It has been 10 years that I am working as a roboticist engineer. I did not expect to write a thesis at this moment of my life. I am still surprised that I am writing it. This thesis results from a procedure of Accreditation of Prior Experiential Learning (APEL), "Validation des Acquis de l'Expérience (VAE)" in French. It gathers research works that I have achieved during the 6 years that I spent as an engineer at Inria Grenoble-Alpes. I am very thankful to Inria who gave me the opportunity to join a brilliant research team, to work with stimulating and kind people and to contribute to exciting research. Thanks to Inria and thanks to the support of IRT NANOEEC, I could work on a continuous research topic such that, over the years, I have built the research material and acquired the experience that I intend to render in the present thesis.

I would like to thank specially Christian Laugier for his guidance and for his continuous support. He has been a kind and careful supervisor for my engineering work. I would not have written this thesis without his encouragements.

I am very thankful to Anne Spalanzani who supervised this thesis. Through her guidance, her advices and with patience, she provided me an important support during the writing and she significantly contributed to improve the quality of this thesis.

Along my career in Robotics, I had the opportunity to meet great people that I would like to thank for the influence they had in my career and in my life. So I would like to thank Gaëtan Garcia who taught me my first ever course in Robotics and later advised me to apply for the master EMARO : what a great idea it was ! I would like to thank Giuseppe Casalino for being the kindest and most inspiring teacher I have ever met. I was extremely sad to hear that he recently passed away. I would like to thank Teresa Zielińska under whose supervision I wrote my first scientific paper. I also would like to thank all my friends that I met during my 2 years in Genova and in Varsaw for the master EMARO, with a special thank you to Shashank Govindaraj who is my best friend in Robotics although we have not seen each other for a while (we should do something about it). I would like to thank Lionel Charpin who taught me how to program a robot under the constraint of an impatient and highly demanding customer looking at each of your moves. I keep good memories of AlpRobotic, my former colleagues and the great team we were.

My recent stay at Inria has also been an occasion to meet fantastic people. It would like to thank the team CHROMA. It has been a pleasure for me to join the team and work with you. I address a special thank you to the subteam of engineers on the IRT NANOEEC projects and those who used to be in this group. I am also extremely thankful to Anouchka Ronceray for her patience and her support in administrative tasks that contribute to the success of projects such as this thesis. Ultimately, I am thankful to Inria for letting me use my "Compte Personnel de Formation (CPF)" (personal training account) to register for this thesis and I would like to thank Magali Berfini who made it possible.

Plus personnellement, je souhaite remercier l'ensemble de mes amis pour leur soutien de

tous les jours, même dans l’adversité, et je les prie de bien vouloir m’excuser pour avoir été aussi peu disponible cette dernière année. Ces remerciements s’adressent à mes compagnons de route depuis le lycée, aux amis que j’ai rencontrés au GUCEM (le meilleur club du monde) et à tous mes amis plus éloignés géographiquement mais qui me restent chers. Je pense par exemple à tous ceux que j’ai rencontrés à Centrale Nantes.

Mes remerciements s’adressent bien sûr à ma famille. En particulier, à mon père, qui le premier m’a donné goût aux sciences techniques et qui fut mon plus grand soutien pendant mes études, à ma mère, qui me comprend si bien et qui m’a soutenu, encouragé et conseillé dans mes projets et mes choix de vie, et à ma grand-mère, Raymonde, qui a su prendre soin de moi aussi bien pendant la rédaction de cette thèse que lorsque j’ai appris à lire.

Naturellement, j’adresse mes remerciements et tout mon amour à Evgeniia. Ton soutien, tes encouragements et tous nos moments de vie à deux me sont très précieux. Nous nous étions tous les deux engagés dans la rédaction d’une thèse, il semblerait que je vais finir le premier, je te souhaite beaucoup de courage pour la continuation de ta thèse. Je serai à tes côtés.

Vous avez tous contribué à mon éducation, à ma carrière et à ma vie. Cette thèse est le résultat de votre influence bienveillante. Merci.

Bien à vous,

Thomas Genevois

Contents

Abstract	3
Acknowledgment	5
Contents	7
Glossary and acronyms	11
1 Introduction	15
1.1 Problem statement	15
1.2 Proposed approach	16
1.3 Contributions	18
1.4 Manuscript outline	18
2 Technical context	21
2.1 Chapter outline	21
2.2 Environment models in perception and collision avoidance	21
2.2.1 Raw sensor data	22
2.2.2 Feature-based models	22
2.2.3 Grid-based models	23
2.2.4 Graph-based models	24
2.3 Dynamic Occupancy Grids for collision avoidance	25
2.3.1 Concept of Dynamic Occupancy Grid	25
2.3.2 Benefits of Dynamic Occupancy Grids for collision avoidance	27
2.4 Massive parallel computations to process Dynamic Occupancy Grids	33
2.5 Chapter conclusion	35
3 Proposed collision avoidance framework	37
3.1 Chapter outline	37
3.2 Need for a new Collision Dectector to interpret Dynamic Occupancy Grids	37
3.3 Review of existing Dynamic Occupancy Grid based collision avoidance solutions and originality of the proposed approach	38
3.3.1 Related works on Dynamic Occupancy Grid based collision avoidance	38
3.3.2 Proposed approach for collision avoidance with probabilistic particle-based representation of obstacles	39
3.4 Methodology of the proposed Predictive Collision Detector (PCD)	41
3.4.1 PCD framework overview	41
3.4.2 Occupancy prediction	42
3.4.3 Collision detection	45

3.4.4	Time To Collision estimation	46
3.4.5	Sampling-based MPC planner	47
3.5	Deployment and experimental results of PCD based collision avoidance	48
3.5.1	Implementation and deployment of PCD based collision avoidance on a car	48
3.5.2	Quantitative evaluation of the safety of PCD based collision avoidance	49
3.5.3	Qualitative results of PCD based collision avoidance	53
3.5.4	Application to other vehicles	58
3.6	Chapter conclusion	61
4	Application of the proposed method to human-aware collision avoidance	63
4.1	Chapter outline	63
4.2	Motivation for an extension of the PCD to consider its application to human-aware navigation	64
4.3	Related works in human-aware navigation and originality of the proposed solution with the extended PCD	66
4.3.1	Related works in human-aware navigation	66
4.3.2	Proposed approach to extend the PCD for human-aware collision avoidance	69
4.4	Methodology of the extension of the Predictive Collision Detector (PCD)	72
4.4.1	Reminder of the Predictive Collision Detector framework	72
4.4.2	Opening the PCD for agents class specific prediction models	73
4.4.3	Definition of a prediction model for pedestrian agents	74
4.5	Deployment and evaluation of the extended PCD in the context of human-aware collision avoidance	77
4.5.1	Deployment of the extended PCD on a real vehicle and transferability to real-world applications	77
4.5.2	Qualitative evaluation of the extended PCD	80
4.5.3	Quantitative evaluation of the extended PCD : efficiency, safety and pedestrian comfort	82
4.6	Chapter conclusion	88
5	Proposed Augmented Reality test framework	91
5.1	Chapter outline	91
5.2	Testing automotive collision avoidance software requires new test frameworks	92
5.3	State of the art of automotive mixed reality testing and originality of the proposed approach	94
5.3.1	State of the art of mixed reality testing for automotive software	94
5.3.2	Proposed approach of Augmented Reality at sensor level	97
5.4	Methodology of the proposed Augmented Reality (AR) solution	98
5.4.1	Structure of the proposed AR framework	98

5.4.2	LiDAR point cloud augmentation for AR at sensor level	100
5.5	Implementation and experimental validation of the proposed AR framework .	103
5.5.1	Procedure for the validation of the proposed AR framework	103
5.5.2	Implementation and real time execution of the proposed approach . .	103
5.5.3	Integrity of the shapes objects in AR augmented scenes	105
5.5.4	Relative position and speed consistency of virtual objects	107
5.6	Applications of the proposed Augmented Reality Framework	111
5.6.1	New opportunities for automotive software testing	111
5.6.2	Application to test collision mitigation	112
5.6.3	Application to test navigation in crowded environments	113
5.7	Chapter conclusion	115
6	Conclusion and future works	117
6.1	Global conclusion	117
6.2	Contributions	118
6.2.1	List of publications	118
6.2.2	Scientific diffusion	119
6.2.3	Transfer projects	119
6.3	Future works	122
	References	123
	Appendices	131
A	Experimental platforms	131
A.1	Experimental automated car	131
A.2	Urban light vehicle prototype	133
A.3	Industrial logistics robot	134

Glossary and acronyms

- ADAS** Advanced Driver Assistance Systems. 15, 92, 93, 115, 131
- ADS** Automated Driving Systems. 15, 16, 18, 19, 22, 24, 28, 29, 31, 32, 37, 50, 55, 61, 63, 64, 66, 92, 93, 96, 115, 117, 131
- agent** In the path planning process, actors of the scene and moving obstacles that coexist with the ego-vehicle and risk to collide with it are referred to as agents. 38, 44, 58, 66, 67, 69, 71–73, 81, 89, 90, 95, 117, 118
- AR** Augmented Reality, within this thesis, it refers to Augmented Reality on sensor data that is used as a test method that challenges the navigation software of a robot by adding virtual elements to its sensing of the environment. 17–19, 32, 33, 57, 79, 89, 91, 93, 95–98, 100, 103–105, 107, 111–113, 115–118, 120–122
- CMCDOT** Conditionnal Monte-Carlo Dense Occupancy Tracker, it is a generic spatial occupancy tracker which infers dynamics of the scene through an hybrid representation of the environment, consisting of static occupancy, dynamic occupancy, empty spaces and unknown areas. It produces a Dynamic Bayesian Occupancy Grid. It has been proposed in [RNL15]. This software together with the Ground Elevation and Occupancy Grid Estimator [RPNL17] forms the perception suite that has been used all along the present thesis. 16, 23, 25–27, 29, 32, 34, 38, 46, 48, 58–60, 64, 69, 71, 72, 78, 89, 106, 107, 120, 121, 132, 134, 135
- Collision Detection** Collision Detection is the process of interpreting grid-based perceived obstacles into collision risks in a configuration space. 16, 24, 38
- configuration** A configuration is the description of a state of a robot, including the position of its joints. For planar mobile robots, a configuration is given by the two-dimensional position and the heading. A time stamp might also be included in the configuration. 24, 39, 41, 45, 46, 108
- CUDA** NVIDIA Compute Unified Device Architecture [L08]. 34, 48, 78, 104
- Dynamic Occupancy Grid** please refer to section 2.3.1 for an extensive definition. 16, 18, 19, 21, 23–27, 35–39, 41, 42, 46, 61, 63, 72, 89, 106, 115, 117, 118, 122
- ETSI** European Telecommunications Standards Institute. 22, 23, 33
- FRP** Freezing Robot Problem, a common issue of robotic navigation in crowded environments that has been identified in [TK10]. It occurs when the environment surpasses a certain level of complexity such that the planner decides that all forward paths are unsafe then the robot is 'frozen'. 63, 65, 67–69, 72, 83, 84, 89

GEOG Ground Elevation and Occupancy Grid Estimator, it is a method for ground labeling in 3D point clouds that filters local ground elevation with a Spatio-Temporal Conditional Random Field. It is combined with a LiDAR inverse sensor model to estimate an instantaneous Bayesian Occupancy Grid. It has been proposed in [RPNL17]. This software together with the Conditionnal Monte-Carlo Dense Occupancy Tracker [RNL15] forms the perception suite that has been used all along the present thesis. 25, 32, 34, 106, 120

GMM Gaussian Mixture Model. 74, 75

GP-GPU Global Purpose computing on Graphics Processing Unit, more details are provided in section 2.4. 17, 19, 21, 34–36, 38, 41, 44, 47, 48, 61, 74, 78, 91, 100–102, 104, 115

HiL Hardware-in-the-Loop, it refers to a simulation environment where a software under test can be executed as it would be on the actual device and on the same computing hardware. This is a step more towards real-world than software-in-the-loop. It forces to respect the memory and computing capability constraints with a real-time execution. 48, 49, 53, 54, 58, 94, 95, 131, 134, 135

instance segmentation Instance segmentation is the process of distinguishing objects of the same class. 23

MPC Model Predictive Control. 17, 24, 38, 46–48, 60, 69, 78, 109

ODD Operational Design Domain. 15, 32, 40, 64, 82

path A path is a sequence of configurations, without time stamps. 49

PCD Predictive Collision Detector, it is a software module which is proposed in the present thesis. Please refer to section 3 for more details. 17–19, 37, 38, 41, 42, 47, 48, 52–55, 58–61, 63–67, 69, 71–74, 77–89, 91, 92, 96, 97, 107–111, 113–118, 120–122

perception Perception is the process of representing a robot’s sensory information in a task-oriented model of the world (from [SK16]). 15, 16, 21, 95

planning Planning is the process of searching a sequence of commands to drive a robot on a collision-free trajectory towards a goal. It can be subdivided in global planning (also known as path planning), the long-term planning with little consideration of dynamic elements, and local planning (also known as motion planning and collision avoidance), the short-term planning that considers dynamic elements on the way to a local goal. The present thesis focuses on the task of local planning. 15, 16, 21

representability The representability of a simulation or of an augmented reality system is defined as the capacity of the system to reproduce the conditions of real-world tests.

If the system is representable, the tests that are conducted in the virtual or augmented environment generate the same outcome than real-world equivalent tests. [92](#), [94](#), [103](#), [105](#), [109](#), [111](#)

RSS Responsibility-Sensitive Safety, a mathematical model for safety assurance in driving applications that has been proposed in [[SSSS17](#)]. [19](#), [50–53](#), [55](#), [56](#), [61](#), [85](#), [86](#)

semantic segmentation Semantic segmentation is the process of identifying the nature of an object (human, vehicle, building, ...). [23](#), [25](#)

SFM Social Force Model, it is an empirical model for motion of humans in crowds which is inspired from Newton's laws of motion. It has been proposed in [[MHG⁺09](#)] from experimental data and commonly used since then. [63](#), [68](#), [72](#), [74–76](#), [87](#), [89](#)

SiL Software-in-the-Loop, it refers to a simulation environment where a software under test can be executed as it would be on the actual device. This implies specifically to respect real-time constraints and same inputs and outputs as in the real-world experiment. [79](#), [80](#), [82](#), [83](#), [89](#), [94](#), [95](#), [118](#), [120](#), [121](#), [131](#), [134](#)

SPACISS Simulation of Pedestrians and an Autonomous Car in Shared Spaces, it is a simulator that is inspired from observation of actual human behaviors and tailored to model interactions between a car and numerous pedestrians. It has been proposed in [[P21](#)]. It considers avoidance and social interactions among pedestrians as well as reactions of pedestrians to the car resulting either from the decision of a single person either as a group behavior. Moreover randomized initialization and decisions generate a large manifold of behaviors to challenge the collision avoidance. Simulated behaviors vary from cooperative (pedestrians yield or avoid the car) to selfish (they follow their path and do not mind the vehicle). [49](#), [51](#), [79](#), [82](#), [85](#), [87](#), [105](#)

trajectory A trajectory is a sequence of time-stamped configurations. [24](#), [41](#), [45](#), [46](#)

TTC Time To Collision, it is defined as the mathematical expectancy of the date of first collision on a given trajectory. [19](#), [41](#), [46–48](#), [108–111](#)

V2X Vehicle-to-everything communication i.e. communication between a vehicle and any entity that may affect it or may be affected by it. [23](#), [96](#)

ViL Vehicle-in-the-Loop, it refers to a simulation environment that is used for automotive testing. In this approach, the software under test is executed as it would be on the actual device and on the same computing hardware and on-board of the actual vehicle. While the test happens entirely in the virtual environment, the actual vehicle may be standing on a test bench or driving on an empty road. [95](#), [97](#), [98](#), [100](#), [111](#), [112](#), [115](#)

Chapter 1. Introduction

1.1 Problem statement

In the previous decade, [Automated Driving Systems \(ADS\)](#) went through a significant development. Fully automated driving is becoming a reality with autonomous shuttles and robot taxis [L23]. Regulatory instances have authorized some experimentations of [ADS](#) [W22]. [Advanced Driver Assistance Systems \(ADAS\)](#) are also increasing safety and making drive easier in cars of the general public [NHT23]. With the astonishing results that have been achieved by both industrial and academic actors, one might think that [ADS](#) are ready for an everyday use for people's transportation. Yet, in spite of the millions of miles that were driven by autonomous vehicles, autonomous driving remains an open research field with numerous unsolved issues.

[Perception](#) is one of the most important open research topics [YLCT20]. Commercially released [Automated Driving Systems \(ADS\)](#) have shown that they suffer from perception issues, either faulty detections causing undesired braking [H22, R22, B22] either missing detections resulting in fatalities[A21]. Another critical open research topic is the drive in complex environments. [ADS](#) have only been released on structured roads and highways, the organization and well defined traffic rules of these [Operational Design Domain \(ODD\)](#) make them simple environments where to navigate. Several crashes of automated vehicles proved that it is more difficult to navigate when this structure gets confused, for example in traffic jams resulting from a pre-existing car crash and when vehicles have to yield before a first responder vehicle [PR22]. Urban environments and especially shared spaces, are way more challenging [W23, C20, L23, M23] because they present less structured environments where vehicles might interact with many actors of a wide diversity (pedestrians, cyclists, animals, other vehicles, ...). This is again a problem of perception and also a problem of navigation in interaction with other actors.

Recent major progresses in perception, in sensor fusion, in semantic and instance segmentation, in perception under uncertainty, prepare the way for new developments of [Automated Driving Systems \(ADS\)](#) and solve the above-mentioned issues. However these new perception techniques require to be used with appropriate motion [planning](#) systems that can benefit from such rich inputs. The formats of the data that perception systems use to represent the surroundings of the vehicle have evolved. Therefore planning systems must evolve in concordance. Despite significant progresses in planning for vehicles, research works very rarely consider realistic inputs from state-of-art perception systems. Therefore it appears that the study of jointly designed state-of-art perception and planning systems is needed to benefit from advances in perception and finally have [ADS](#) safer and deployed in new [Operational Design Domain \(ODD\)](#).

While joint design of perception and planning is an ideal for navigation of a vehicle in complex environments and management of sensor faults, this itself is a challenge. First, the

perception system and the planning system must share a common format to represent the environment of the vehicle. On the one hand, state-of-art perception systems tend to complex formats, rich, dense, stochastic and semantic, such as bird-eye-view images of the vehicle's surroundings. On the other hand, planning systems tend to use much simpler formats such as finite lists of objects. This is mandatory to apply several planning techniques, such as closed-forms approaches. Second, making use of richer information in the planning system is likely to increase its computational complexity but the navigation system must still satisfy the hard real-time constraints of **ADS** and the frugality requirements of embedded hardware. Finally, when using a tight coupling of perception and planning through a complex data format, it is highly difficult to validate these systems independently. Indeed there is often no available metrics on this format to evaluate the perception outputs such that the perception performance can not be evaluated independently. Similarly there is generally no simulation model to generate the planning inputs that would be used to evaluate the planning performance in a simulated environment. Therefore perception and planning have to be validated as a whole.

Within this context, the present thesis considers one particular state-of-art perception format, the Bayesian **Dynamic Occupancy Grids** which is the output of the **Conditionnal Monte-Carlo Dense Occupancy Tracker [RNL15] (CMCDOT)** perception software [RNL15]. This perception system has shown superior performances in modeling uncertainties and complex scenes. It appears then as a great candidate to solve the current issues of **ADS** that have been mentioned above. Despite many significant research works on this perception format and other similar formats, very few attention has been focused on its use for planning. Therefore, in order to make **ADS** benefit from such a promising perception system, the present thesis intends to propose a collision avoidance system that has been tailored for Bayesian Dynamic Occupancy Grids. Also a new validation framework has been designed to evaluate the performance of the proposed avoidance framework as a coupled perception and planning system. With this contribution, this thesis aims to facilitate the use of the **CMCDOT** and similar perception systems in **ADS**. Ultimately, the author hopes that this thesis will enhance the connectivity of the states of the art of perception and planning.

1.2 Proposed approach

It has been noticed in section 1.1 that current **Automated Driving Systems (ADS)** suffer from **perception** issues and difficulties to operate in complex crowded environments. While promising research works in perception have been conducted, they are rarely applied to navigation because they make use of new formats to represent the environment that are not recognized in **planning** approaches.

The present thesis proposes to enhance the usability of one particular perception format, the **Dynamic Occupancy Grid**, in local planning applications such that its inherent properties could contribute to solve the problems of current **ADS**. In order to do so, a new Dynamic Occupancy Grid based collision avoidance framework is proposed. It relies on a **Collision Detection** that would interpret this perception format into configuration space. This new Col-

lision Detector combines a prediction module that interprets motion particles with the general collision detection on the probabilistic occupancy grid. The whole algorithm is designed for efficient parallelization in [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#). This contribution is then named [Predictive Collision Detector \(PCD\)](#). In the present thesis, it enables to apply sampling-based [Model Predictive Control \(MPC\)](#) with a Bayesian Dynamic Occupancy Grid based perception. This [PCD](#) is proposed as a general interface between sampling-based planners and Dynamic Occupancy Grid based perception solutions. This collision avoidance framework is deployed and evaluated on several vehicles including a prototype of autonomous car. Various examples illustrate that this proposed system benefits from the probabilistic and dense of the environment that Dynamic Occupancy Grids provide.

This first collision avoidance framework does not consider neither the semantic classification of agents neither interactions with them. An extension of this framework considers a semantic grid as an additional input. It enables to continuously adapt the prediction using a classification probability and class-specific prediction models. This new framework is demonstrated with a class-specific prediction model for pedestrians. This prediction model, inspired by the literature of human motion prediction, is behavioral and interaction-aware. This more accurate prediction significantly improves the previous collision avoidance system and enables navigation in crowded environments. This improved collision avoidance system is capable of cooperative navigation with humans. Its results qualitatively compares with the state of the art of human-aware navigation but it appears to be more versatile and scalable. It has also been deployed on a prototype of autonomous car under realistic perception conditions. It also illustrates the resilience of the overall system when facing highly dense crowds.

In automotive applications, testing and validation commonly rely on the ability to augment a vehicle's perception at run time with virtual objects. This method of [Augmented Reality \(AR\)](#) ensures repeatability of the tests, enables to safely test collision scenarios, allows dynamic reconfiguration of scenarios and reduce the overall engineering effort for testing. However these techniques cannot be used as such with Dynamic Occupancy Grid based collision avoidance because the augmentation of a Dynamic Occupancy Grid with virtual objects is undefined. Instead of augmenting the perception itself with virtual elements, the present work proposes to augment the sensor data. So the concept of [AR](#) at sensor level is introduced. It has the advantage of being agnostic of the internal perception format and more realistic because it challenges jointly the processes of perception and planning. This sensor-level [AR](#) is proposed on LiDAR data. The LiDAR data augmentation can be achieved in real-time thanks to the proposed augmentation algorithm that exploit massive parallelization in [GP-GPU](#). The augmentation is realistic in the sense that it preserves the structure of the actual LiDAR data and models reciprocal occlusions between real and virtual elements. The method is deployed on a prototype of autonomous car. It is then used to test and validate the propose Dynamic Occupancy Grid based collision avoidance.

1.3 Contributions

The thesis is structured with 3 main contributions that are listed in the next paragraphs. They have been reused in different research projects and they included software that has been deployed on several experimental platforms within transfer projects. These applications are described in section 6.2.3.

The first main contribution of the present thesis is a [Dynamic Occupancy Grid](#) based collision avoidance framework together with its core software module which is referred to as [Predictive Collision Detector \(PCD\)](#). This framework achieves a versatile collision avoidance from a probabilistic perception input. The PCD could be generalized as a standard interface between Dynamic Occupancy Grids and sampling-based planners. This development has been realized within the project [IRT NANOEEC – SVA](#) and deployed on 3 vehicles, an autonomous car prototype and 2 lighter robots, within the projects [IRT NANOEEC – SVA](#), [IRT Nanoelec – Light Vehicle Demonstrator](#) and [R&D Booster MoovIT](#). This work has also been used to produce experimental data for the project [PRISSMA](#). The description of this contribution has been published in [\[GRSL23\]](#).

The second main contribution of this thesis is the extension of the [PCD](#) to human-aware navigation thanks to an additional semantic classification input and behavioral interaction-aware prediction models. Enriched with pedestrian prediction model, this new version of the PCD has significantly increased the performance of navigation in crowded environments without reducing safety. This development has been realized within the project [IRT NANOEEC – SVA](#) and deployed on 2 vehicles, an autonomous car prototype and a lighter robot, within the projects [IRT NANOEEC – SVA](#) and [R&D Booster MoovIT](#). This extension of the PCD has been proposed and published in [\[GSL23b\]](#).

The third main contribution of this thesis is an [Augmented Reality \(AR\)](#) testing and validation framework for advanced automotive software together with the real-time LiDAR data augmentation module. This achieves real-time and realistic augmentation of LiDAR sensor data with virtual elements in order to challenge advanced automotive software. This development has been realized and deployed on an autonomous car prototype within the project [IRT NANOEEC – SVA](#). This work has also been used to produce experimental data for the project [PRISSMA](#) that investigates new validation approaches for [Automated Driving Systems \(ADS\)](#). This new approach for testing and validation has been presented and published in [\[GHRL22\]](#).

1.4 Manuscript outline

The chapter 2 [Technical context](#) introduces notions that are not contributions of the present thesis but that are necessary for its understanding. This chapter proposes first an analysis of the interconnections of common perception and planning methods through the chosen environment representation format to illustrate that [Dynamic Occupancy Grid](#) are rarely used for local planning. The [Dynamic Occupancy Grid](#) has been chosen as input of the collision

avoidance framework that proposed in this thesis. This format is described along with its potential benefits and constraints for local planning. The main disadvantage of this perception format is to be particularly heavy in memory and computationally expensive to process. So the concept of massive parallel computing and its application on GPU, [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#), is presented with constraints that it yields on the design of the algorithmic contributions of this thesis.

The chapter [3 Proposed collision avoidance framework](#) introduces the proposed collision avoidance framework. It is first reminds that the motivation of this work comes from the current issues of [Automated Driving Systems \(ADS\)](#) and from the lack of [Dynamic Occupancy Grid](#) based planning solutions. Then it explains the intention to propose an interface between Dynamic Occupancy Grid based perception and sampling-based planners. A state of the art of occupancy grid based collision avoidance is presented and the proposed approach is compared with it. The proposed collision avoidance solution is detailed with an overview of its framework and its major contributed components that constitute the proposed [Predictive Collision Detector \(PCD\)](#), a particle-based occupancy prediction, the stochastic collision detection and the computation of an expected [Time To Collision \(TTC\)](#). The deployment of this collision avoidance is presented on different applications and experimental results on 3 different vehicles are displayed. Multiple qualitative results and analysis of the avoidance behavior are presented. The [Responsibility-Sensitive Safety \[SSSS17\] \(RSS\)](#) definition of safety distances is introduced and it is used in a quantitative evaluation of the safety of the proposed system in simulation.

The chapter [4 Application of the proposed method to human-aware collision avoidance](#) describes the extension of the proposed collision avoidance framework to human-aware navigation. After reminding the current limitations of autonomous vehicles in shared spaces, this chapters presents the proposed additions to the collision avoidance framework. As this extension of the [PCD](#) targets navigation in human populated environments, a state of the art of human-aware navigation is presented. The proposed approach is introduced along with its originality within the state of the art. Then is presented the methodology of this addition to the [PCD](#), how it handles a continuous classification probability and how a state-of-art human-motion model has been integrated in the probabilistic prediction. After an explanation of the implementation details, qualitative and quantitative results are presented. These results include a quantitative comparison with the previous collision avoidance framework and experiments on an autonomous car.

The chapter [5 Proposed Augmented Reality test framework](#) presents the [Augmented Reality \(AR\)](#) validation framework to evaluate the performance of the collision avoidance systems of chapters [3](#) and [4](#). It is first explained why this new tool is needed to validate automotive software such as the proposed collision avoidance. The related state of the art is presented and the originality of the proposed approach, that provides real-time Augmented Reality at sensor level instead of object-based representations, is pointed out. The structure of the framework is described with its modules and the LiDAR data augmentation function which is at the core of this system. Then, implementation details, qualitative experimental results and an

experimental quantitative validation are provided.

Chapter 6 [Conclusion and future works](#) concludes the present thesis. It first synthesizes the overall work that is presented all along the document. Then the contributions of the present thesis are listed in terms of published papers, in terms of reuse in transfer and research projects and in terms of broader scientific diffusion. Finally several possible future works are proposed to continue with the contributions of the present thesis.

Chapter 2. Technical context

2.1 Chapter outline

This chapter introduces concepts that are not contributions of the present thesis but that are necessary for its understanding. The section [2.2 Environment models in perception and collision avoidance](#) proposes an analysis of the interconnections of common perception and planning methods through the chosen environment representation format and it is observed that [Dynamic Occupancy Grid](#) are rarely used for local planning. Then section [2.3 Dynamic Occupancy Grids for collision avoidance](#) defines the notion of [Dynamic Occupancy Grid](#) and presents its potential benefits and constraints for local planning. This will be widely reused further in this thesis since Dynamic Occupancy Grid is the input of the collision avoidance framework that is proposed in this thesis. Afterwards section [2.4 Massive parallel computations to process Dynamic Occupancy Grids](#) describes the concept of massive parallel computing and its application on GPU, [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#). It is presented as a convenient tool to process Dynamic Occupancy Grids but constrains the operations. This technique is considered for the software contributions of this thesis and its constraints are incorporated in the design of the algorithms.

2.2 Environment models in perception and collision avoidance

On the one hand, [\[SK16\]](#) defines [perception](#) as a task-oriented representation of the environment of a robot. On the other hand, the performance of a [planning](#) system is bound to its available perception. Then, it appears that there is an intrinsic dependence between perception and planning that lies in the format which is used to describe the perceived environment. Optimally, perception and planning share the same format. If they do not, a translation is needed from the format in perception to the format in planning. This translation should be lossless such that all perceived information can be valued in planning. However recent evolutions of the perception techniques [\[PCD⁺23\]](#) have led to new ways to represent the environment for which there is no lossless interpretation in planning. This causes suboptimal performance of the whole navigation system that is a possible cause of issues that were presented in section [1.1](#). In order to identify in which cases this occurs and what can be done to improve the situation, the next paragraphs go through the usual formats that roboticists use to model the environment of a robot and their common use in perception and planning. This has been established with the help of [\[SK16\]](#), [\[YLCT20\]](#) and [\[GPMN16\]](#). [\[SK16\]](#) categorizes common environment representations in robotics in 4 base categories : raw sensor data, features, grids and graphs. In a graphical display, [fig.1](#) presents these world modeling formats with the perception and planning techniques they are commonly used with. In this figure, grid-based perception appears actively researched in perception but rarely used in local planning. In the meanwhile, configuration space graphs are commonly used in effective planning

approaches while they are not directly observable. Considering this, the present thesis investigates the interpretation of recent grid-based representations into configuration space graphs. It would enable to combine the benefits of grid-based perception and graph-based planners. The presented list of world modeling formats, perception and planning techniques and its graphical illustration intend to provide an overview but they are not meant to be exhaustive. Also each of the 4 main formats can be subdivided in several modeling approaches that differ in their nature and in their purpose.

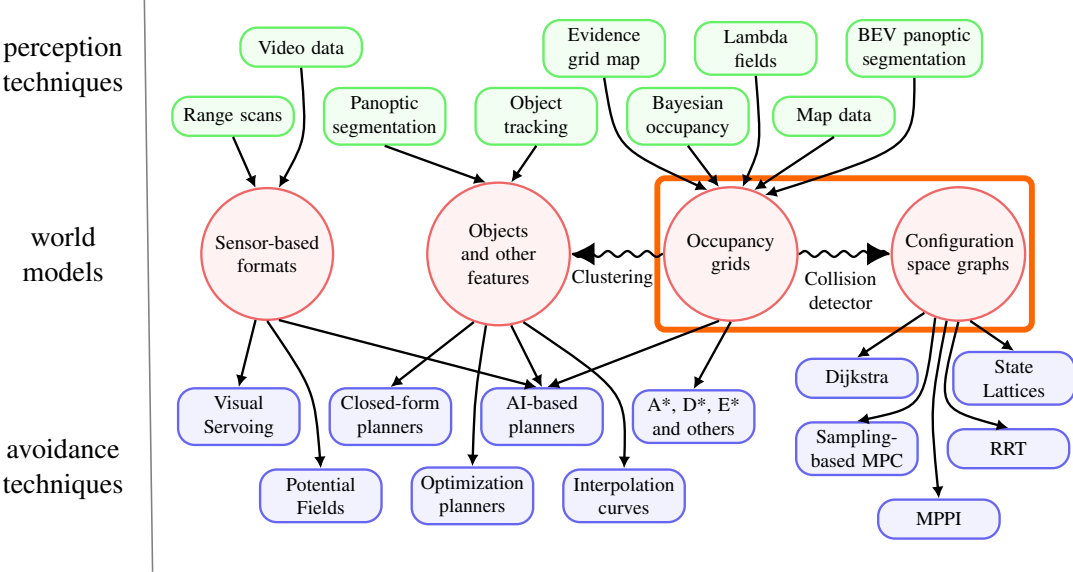


Figure 1 – Non-exhaustive graphical representation of common associations between perception, world modeling and collision avoidance techniques.

2.2.1 Raw sensor data

Some planning methods have been developed to directly process sensor data without using additional perception software. This is less common than other approaches but has been often used with cameras (visual servoing) or range scans. Potential Fields [K86] applied with LiDAR scans is a popular example. However the direct use of sensor data is only possible for simple navigation tasks and lacks of explainability.

2.2.2 Feature-based models

In robotics, The most common way to describe the world around a robot is to describe it with a finite list of features that it contains. These features might be actual entities (objects, actors, obstacles) or abstract features (landmarks, waypoints, curves, particles, probability distributions, ...). Considering Automated Driving Systems (ADS), features are generally the other actors and obstacles, with their shape information as a bounding box, speed and semantic information. Curves are also used to describe the road structure. Regarding the application to ADS, in [ETS19], the European Telecommunications Standards Institute (ETSI) defines

object-based standards for shared perception in [Vehicle-to-everything communication \(V2X\)](#). This standard defines a perceived object by its relative position to the ego-vehicle, its speed, its acceleration, its bounding box and its classification. One global confidence is associated to the whole observation then confidences are provided for the position estimation. With this standard, [ETSI](#) enables a very limited expression of uncertainty. These representations can be obtained by [semantic segmentation](#) and [instance segmentation](#) either in learning-based approaches or with clustering and tracking on sensor data or occupancy grids. In terms of perception, these approach have several drawbacks. First, they do not provide a dense representation of the world i.e. there is no information about places where there are no objects. In the [ETSI](#) standards, this can be partly compensated by providing a simplified free space expression with uniform confidence on simple geometric shapes but this does not suffice to express realistic sensing uncertainties. Second, object-based models fail to represent uncertainties in the existence or segmentation of objects. Finally, the complexity of this representation depends on the scene. In terms of planning, feature-based models are very convenient to manipulate with a low computational cost. Such finite list of object are easy to integrate in closed-forms or optimization problems. They are also often used in learning-based planning approaches [[A20](#)].

2.2.3 Grid-based models

Grid-based representations of a robot's environment are also gaining attention of the research community. They consist of a two-dimensional discretization of the scene, generally in bird-eye-view. They might be allo-centric (in a fixed frame) or ego-centric (in the robot's frame). While maps are the most natural source of grid-based representations, such representations can also be generated with several learning-based approaches and various occupancy filtering methods. Unlike feature-based approaches, grid-based representations provide dense information and a convenient support for uncertainty. Yet they fail to accurately model objects and distances below their resolution and beyond their boundaries while resolution and size must remain bounded to preserve reasonable computational complexity. These grids are also difficult to manipulate with most planning approaches. Currently grid-based representations are mostly for global planning, for example with the A* planner [[HNR68](#)] and associated approaches. Some learning based approaches also process grid-based representations, they are convenient for this because they can be processed as images.

Several research works in perception propose innovative grid-based formats. The Hybrid-State Bayesian Occupancy Filter [[NRL14](#)] and the [Conditionnal Monte-Carlo Dense Occupancy Tracker](#) [[RNL15](#)] ([CMCDOT](#)) [[RNL15](#)] propose the concept of Bayesian [Dynamic Occupancy Grid](#) that models occupancy and motion under uncertainty according to the Bayesian probability theory with a grid representation for static occupancy and list of particles to model moving elements. The notion of Evidence Grid [[MCT⁺17](#)] is a similar concept under the Dempster-Shafer Evidence theory. The Lambda-Fields approach [[LRK⁺21](#)] proposes another solution which does not depend on the tessellation size. [[EWL19](#)] incorporates se-

semantic information into [Dynamic Occupancy Grid](#). While the above-mentioned approaches rely on explicit inverse sensor models and probabilistic occupancy filtering, there are also model-free approaches relying on deep learning to compute occupancy grids. [SBGD22] presents such a method to generate a Dynamic Occupancy Grid together with the semantic grid. [DZSGE⁺23] proposes a semantic grid-based perception. Despite the significant results of these works, there are little research on [Dynamic Occupancy Grids](#) in a planning perspective [F09].

2.2.4 Graph-based models

Unlike the above-mentioned approaches, graph-based formats enable the representation of the structure of world. For [ADS](#), they have been used in decision making problems to represent interactions between agents. Graphs are also used to represent the topological structure of the scene. They can be extracted from maps or grids and interpreted in global planning and routing problems. All graph-based approaches offer high-level abstraction of the problem they have been designed for but they are generally not directly available to any sensor.

A specific but widely used type of graph is the graph in the [configuration](#) space. Configurations denotes the possible states of a robot, in space and possibly in time. While graphs in the State space model the topology of the world, configuration space graphs model the topology of the world according to the robot's motion model. For this reason, they are extremely convenient for planning. If the time dimension is also considered, configuration space graphs enable to model feasible [trajectories](#) of a robot, this is a very popular approach in local planning which is shared by all sampling-based planners. For example, Sampling-based [Model Predictive Control \(MPC\)](#) [DCJC08] generates a configuration graph by sampling commands. The Model Predictive Path Integral [WDG⁺16] investigates randomly the configuration space around a trajectory in order to optimize it. The RRT [LK99] approach relies on the construction of rapidly exploring random trees in the configuration space. Also the concept of State Lattices [PK05] uses tailored motion primitives in the exploration of a configuration space graph to ensure its connectivity. Despite the success of these approaches and their everyday use in the Robotics planning community, configuration space graphs are not observable and there is no standardized manner to generate them from perception.

From the overview that has been given in this section, it appears that there is active and promising research on grid-based perception while there is little consideration of grid-based local planning in research. In the meanwhile, there are widely used and actively researched local planners that rely on configuration space graphs while such graphs are not directly observable. Therefore interpreting novel grid-based perception formats into configuration space graphs would enable to get the best out of both approaches and would be beneficial for both the perception and the planning research communities. This is the motivation of the present thesis. This process of translation from a grid representation to a configuration space graph is referred to as [Collision Detection](#).

2.3 Dynamic Occupancy Grids for collision avoidance

2.3.1 Concept of Dynamic Occupancy Grid

The present thesis uses a perception system that relies on the [Ground Elevation and Occupancy Grid Estimator \[RPNL17\] \(GEOG\) \[RPNL17\]](#) and the [CMCDOT \[RNL15\]](#). Given LiDAR point clouds and the odometry of the vehicle, this software suite computes a [Dynamic Occupancy Grid](#) to represent the environment of the vehicle. This Dynamic Occupancy Grid is then the input perception format for the collision avoidance systems that are studied in this thesis.

In the [CMCDOT](#) framework [RNL15], a [Dynamic Occupancy Grid](#) is a ego-centric 2D bird-eye-view representation of the likelihood of space around the vehicle to be occupied by obstacles, and their estimated speed. It uses a hybrid model, a grid to represent the distribution of occupancy probability and particles to represent the distribution of motion probability. Each cell in the grid provides, $P(D|O)$, the probability of it to be occupied by a dynamic element given that this cell is occupied, $P(O|Z)$, the probability of it to be occupied by an obstacle given that this cell has been observed and, $P(Z)$, the probability of having observed this cell. Each particle comes with p , the probability of occupancy that it conveys and (V_x, V_y) its speed along the planar axes. The dimensions of the grid and the number of particles are fixed parameters that are assumed to be large enough for the desired navigation task.

Fig.2 presents an example of Dynamic Occupancy Grid. This example has been computed on board of an [experimental automated car](#), from LiDAR data that have been processed by the [CMCDOT](#) perception suite [RPNL17, RNL15]. An on-board camera in the front of the vehicle has recorded the scene. The Dynamic Occupancy Grid itself consists of the perceived occupancy and velocity that are displayed in 2 grids for a better understanding. Several elements have been circled on the camera view and on the occupancy grid to help the reader to understand the correspondence. In this urban scene, perceived static objects and road users can clearly be identified in the grids. It also reveals the blind spots, in red. Variations of uncertainty also appear with unclear perception of further elements as well as with inference in currently masked areas. This type of grid is the main perception format for the collision avoidance framework that is proposed in the present thesis. This same display will be reused all along the document.

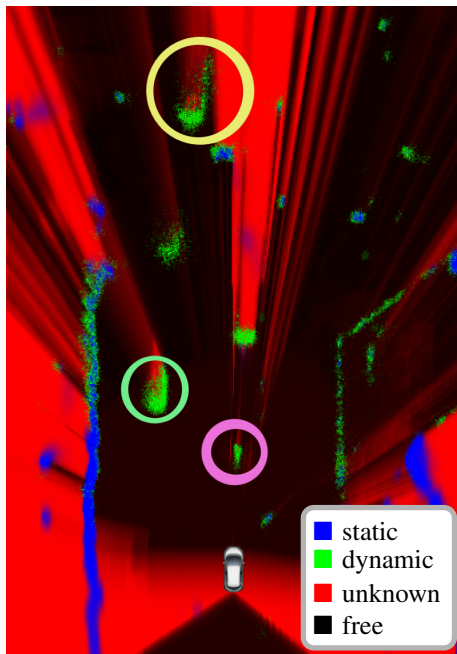
There exist other types of Dynamic Occupancy Grids [N17, SBGD22, LRK⁺21] that rely on different theories of probabilities and different approaches but generate grids that contain similar information. Even if some of these approaches provide a velocity grid instead of list of motion particles, this velocity grid can be regarded as a sparse array whose non-zero elements constitute a list of motion particles. Hence we assume that the collision avoidance framework that is proposed in the present thesis and that relies on the [CMCDOT](#) for perception could theoretically be applied with these perception solutions that also rely on [Dynamic Occupancy Grids](#).

Relying on a [semantic segmentation](#) such as [EWL19] or [DZSGE⁺23], semantic clas-

Vehicle's front camera



Occupancy grid



Velocity Grid



Figure 2 – Example of a *Dynamic Occupancy Grid* as provided by the *CMCDOT* perception suite [RPNL17, RNL15]. This image has been provided by [RDGML22].

sification filtering has been introduced in the [CMCDOT](#) by [\[RDGML22\]](#). This enables the construction of a filtered semantic grid which serves as input of for the work of the present thesis. This semantic grid has the same dimensions that the Dynamic Occupancy Grid and provides for each cell, $P(C_k), k \in \llbracket 0, n \rrbracket$, the probability of particles lying in this cell to belong to an object of class k among n classes. C_0 corresponds to the class of unidentified object. [Fig.3](#) presents an example of semantic grid. In the example, the semantic grid is an additional output of the [CMCDOT](#) perception suite [\[RPNL17, RNL15, RDGML22\]](#). Other approaches such as [\[SBGD22\]](#) integrate the task of semantic classification in the process of estimation of the [Dynamic Occupancy Grid](#). In the present thesis, the semantic grid is used along with the Dynamic Occupancy Grid within the proposed collision avoidance framework.

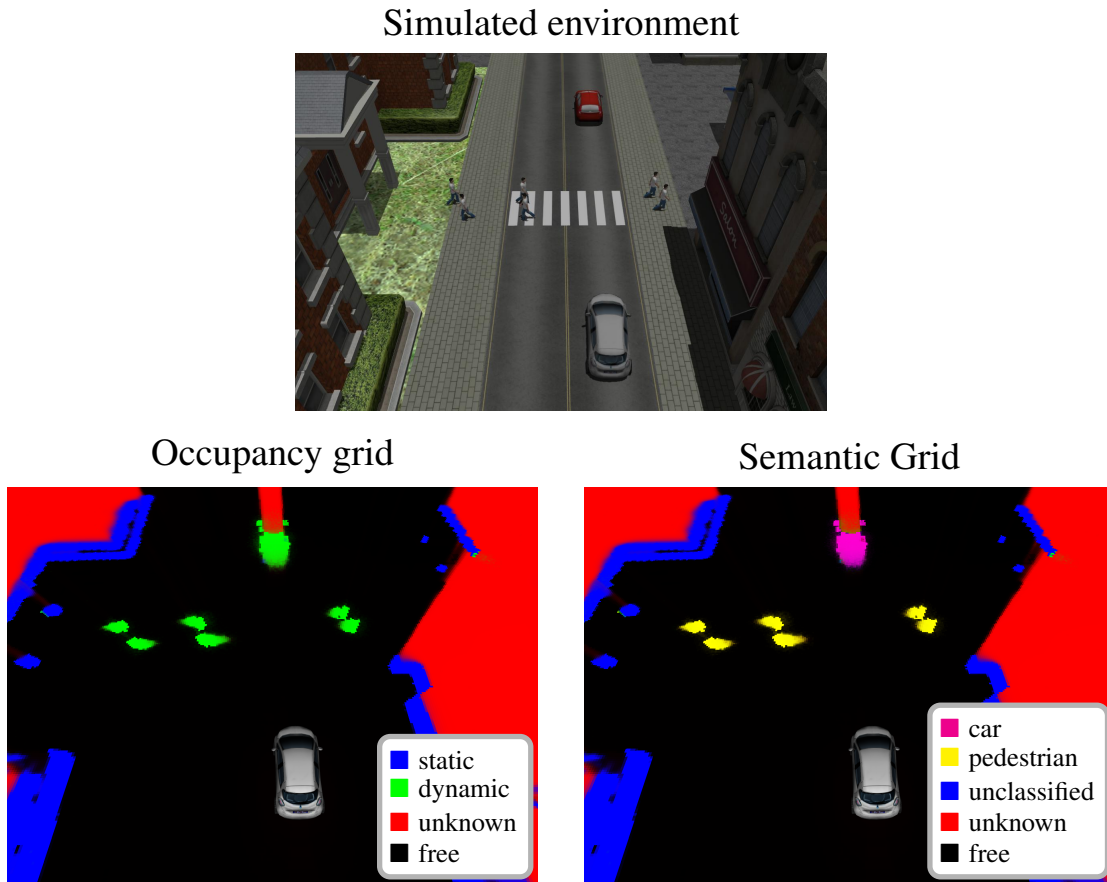


Figure 3 – Exemple of a Dynamic Occupancy Grid and semantic grid as provided by [CMCDOT](#) perception suite [\[RPNL17, RNL15, RDGML22\]](#). The motion estimation is not displayed in this illustration.

2.3.2 Benefits of Dynamic Occupancy Grids for collision avoidance

[Dynamic Occupancy Grids](#) are difficult to manipulate for local planning purposes. First the absence of instantiated objects makes it impossible to use as such with closed forms and several optimization techniques. Then the computational cost to process the large data of this object is an insuperable obstacle for many approaches. Finally, the hybrid representation with

grid and particles is uncommon and prevents the use of classical planning approaches. But the Dynamic Occupancy Grid is a very promising input for new planning approaches. The paragraphs below list the advantages of this environment model for local planning.

Fully probabilistic Unlike object-based representation, Dynamic Occupancy Grids are able to model any probability distribution of the state of the environment. So this format is a versatile tool that can represent accurately any uncertainty in the perception. It can model the uncertainty of the existence of an obstacle, the uncertainty of its shape, of its position, of its classification, of its motion and speed. So Dynamic Occupancy Grids can represent not only a highly confident detailed perception of a scene but also a very vague detection in poor sensing conditions. Such poor sensing conditions are common in Robotics and ADS, either for distant objects, in bad weather, when encountering partial occlusions or with cheap sensors. While object-based approaches are at some point bound to the binary existence or absence of an object, Dynamic Occupancy Grids offer the possibility to model a diffuse likelihood of existence of something unclassified.

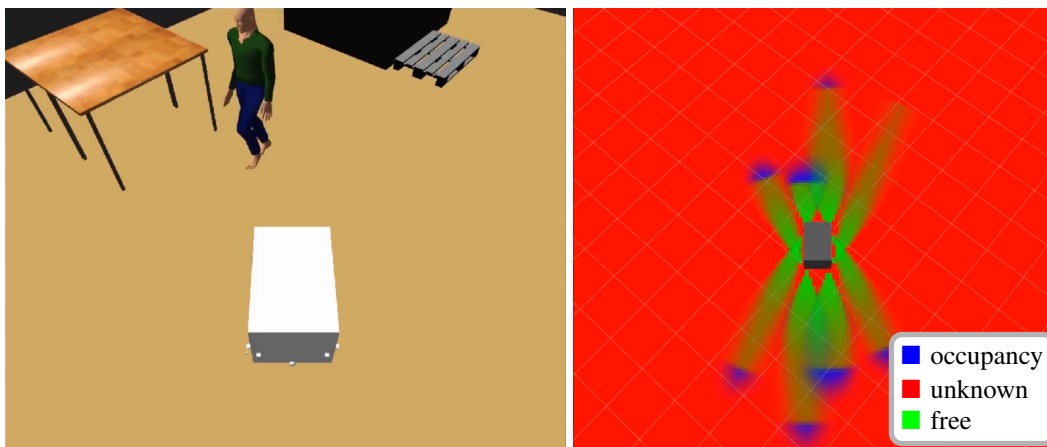


Figure 4 – Example of an instantaneous occupancy grid from ultrasonic range sensors on board of an *industrial logistics robot*. The distribution of occupancy probabilities accurately reflects the uncertainties that are inherent to the sensing modality.

If this could be used for planning, it would help navigation systems of autonomous vehicles to make appropriate collision risk estimations when sensing is limited. It could prevent some fatalities such as those that have been mentioned in section 1.1. Fig.4 and fig.5 illustrate situations where this probabilistic representation of obstacles enhances a robot's perception. In fig.4, an *industrial logistics robot* evolves in an indoor environment. The displayed instantaneous occupancy grid results from the combined perception of 8 ultrasonic range sensors. As these sensors provide only one range measurement over a large angular sector, their detections are modeled by diffuse occupancy on the grid. This grid also reveals blind spots and overlapping observations. The distribution of occupancy probabilities in the grid successfully captures the uncertainties that are inherent to this sensing modality, it would be impossible with object-based models. Fig.5 depicts a similar scene with the same robot and sensors.

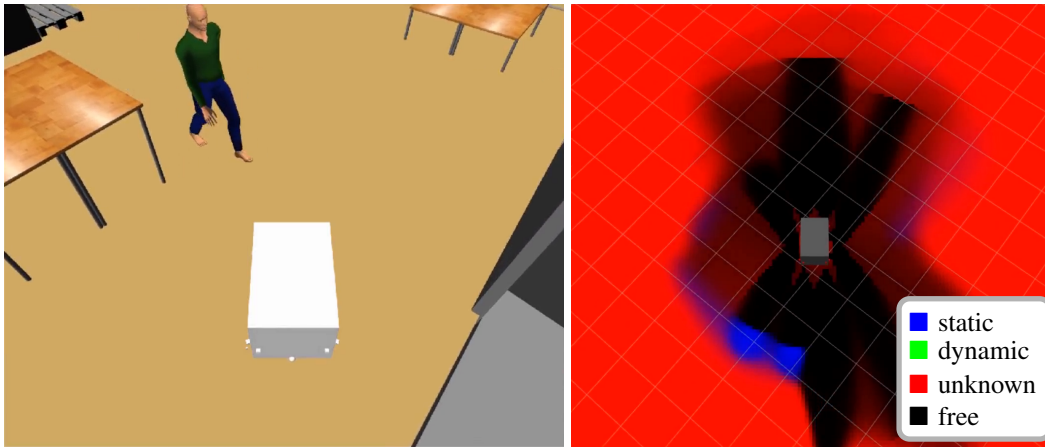


Figure 5 – Example of a filtered occupancy grid from ultrasonic range sensors on board of an industrial logistics robot. Thanks to its distribution of occupancy probabilities, the grid conveys the uncertainties in the inference that result from the spatio-temporal filter.

This time the filtered occupancy grid is displayed. It shows that the spatio-temporal filtering (provided by CMCDOT) associates high confidences to current observations and diffuse lower confidence to persistent past observations. This could enable a planning system to take advantage of memories of previous observations while knowing that it is an uncertain information that might not be valid anymore. Again, such a representation of inferred information could not be achieved with different environment models.

Dense information Unlike other perception representations, occupancy grids offer a unique dense representation of space. While object-based approaches only provide knowledge on the areas where objects lie, grid-based approaches provides the likelihood of occupancy and observability everywhere within the boundaries of the grid. This enables to distinguish the space that has been observed to be free from blind spots and areas with limited observations. Using this for local planning could enhance the robustness of ADS. Fig.6 and fig.7 present examples of the usage of dense occupancy and observability information on occupancy grids. Fig.6 displays a filtered occupancy grid that has been perceived by a car in a simulated urban environment. The grid clearly distinguishes the blind spots (behind building on the right and before leading vehicle) from the space that is observed to be free. This spatially dense classification of space is then available for planning. Fig.7 presents a real world example of a filtered occupancy grid that has been enriched with prior knowledge from a map. The fusion assumes that the information from the map is not as reliable as current observations. This is visible in the occupancy grid through light red colors, it represents some uncertainty in the free space that comes from the map but has not been observed. Fused in this single occupancy grid, both local perception and prior map knowledge have served a path planner in the computation of the green path that drives the car across this environment. The information density serves here to differentiate the reliability of knowledge from different sources.

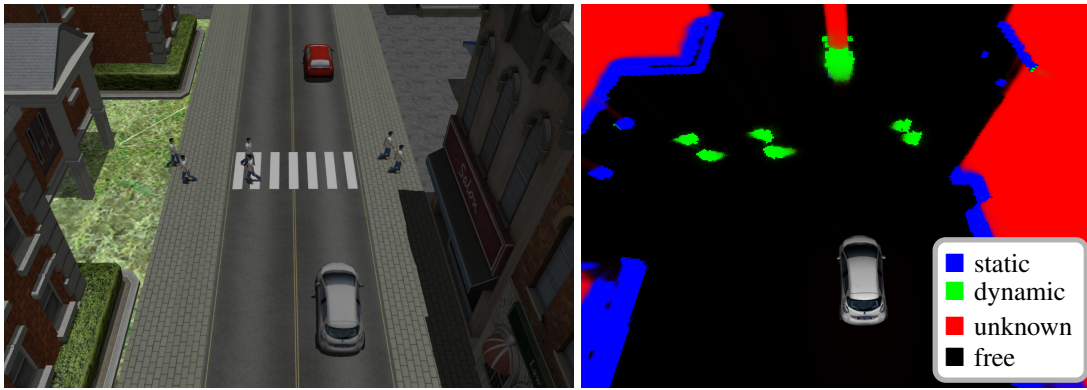


Figure 6 – Example of a filtered occupancy grid from a car in a simulated urban environment. It explicitly displays blind spots and free space.

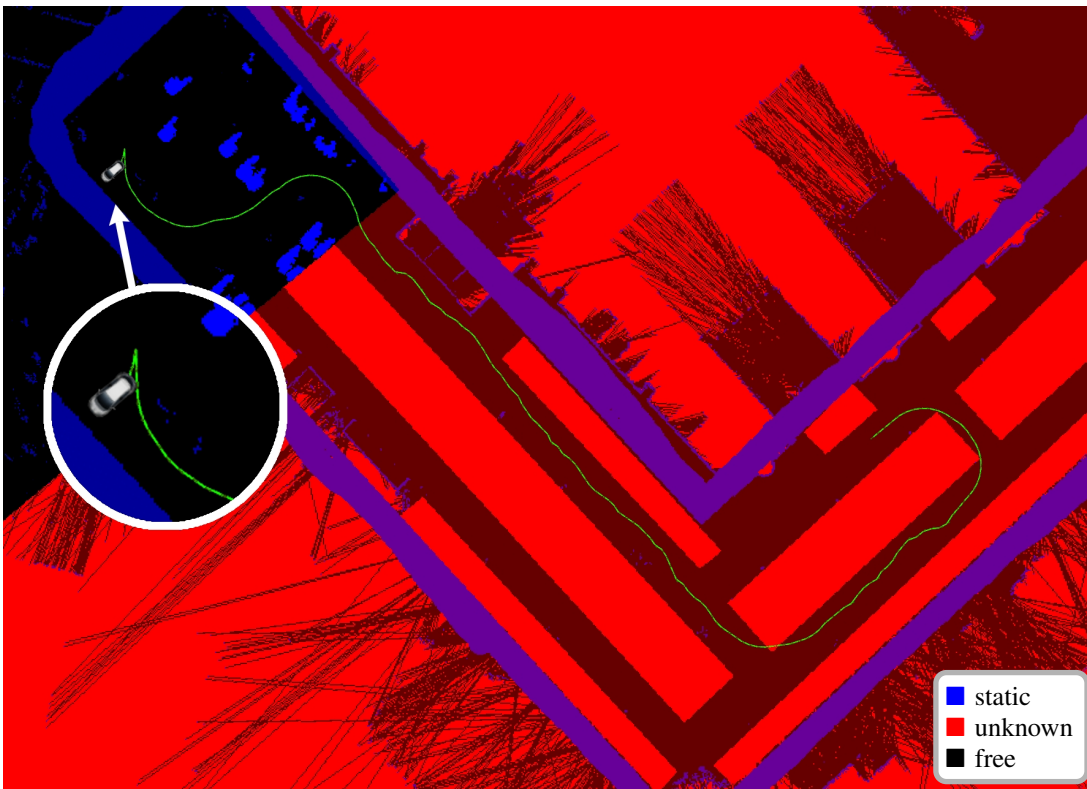


Figure 7 – Real world example a filtered occupancy grid resulting from the fusion of observations from a car and previous knowledge from the map. From this occupancy grid that associate different confidences to each of these 2 sources of knowledge, a path planner has computed the green path to drive the car to another parking spot.

Sensor fusion framework Dynamic Occupancy Grids are agnostic from the sensor. The perception software that produce these grids relies on either on end-to-end approaches either on inverse sensor models that capture the specificity of the sensing device but the resulting grid is abstracted from it. While this is beneficial for the modularity and interoperability of the navigation stack, this also enables sensor fusion at grid level. The grid-based fusion is particularly rich because it takes advantage of dense information in free space that has been

mentioned in the previous paragraph i.e. the observation of a sensor will complete the grid in areas which have remained unobserved by other sensors.

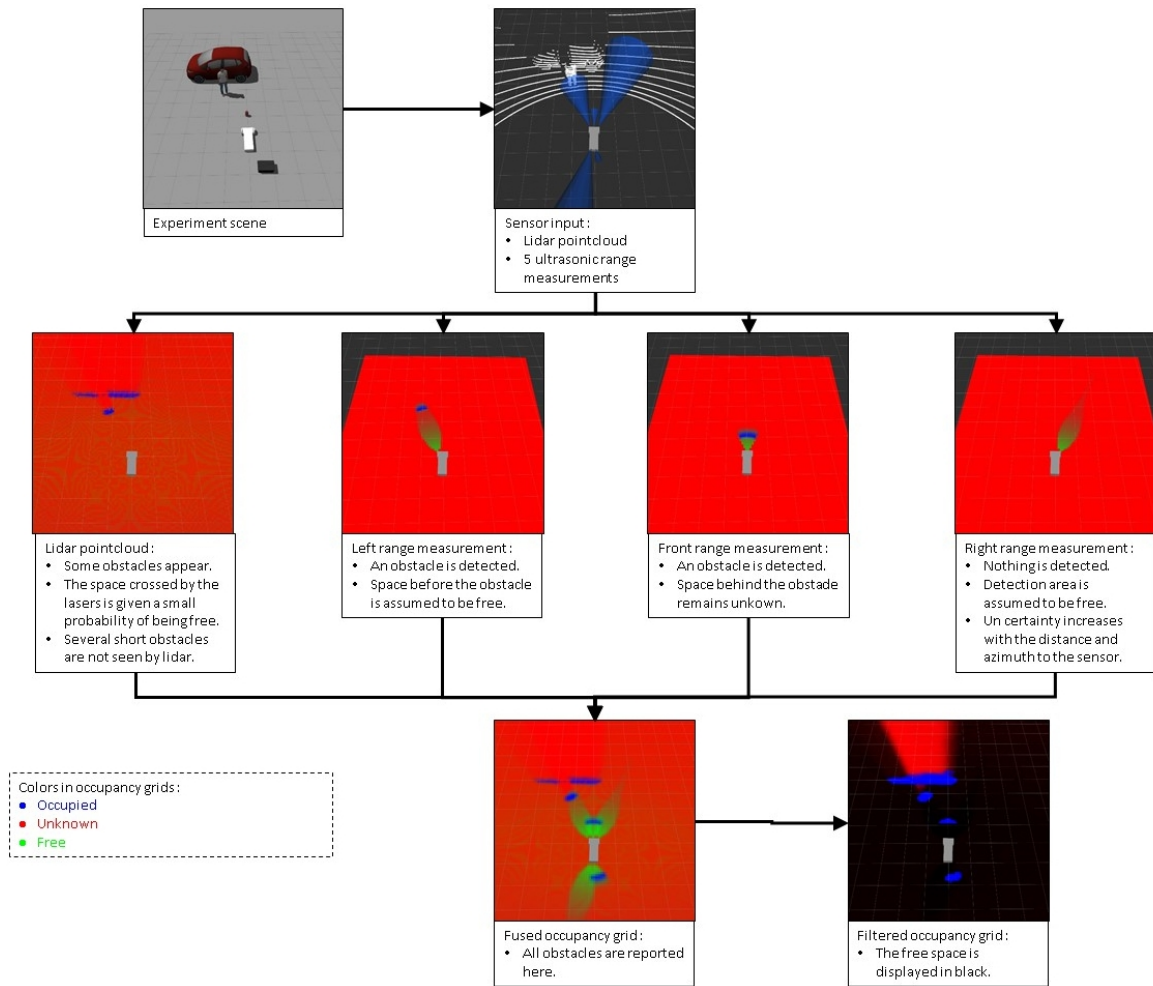


Figure 8 – The perception process that has been deployed on the project *IRT Naoelec – Light Vehicle Demonstrator*. It takes advantage of probabilistic occupancy grids to accurately fuse and filter observations from multiple sensors of different modalities.

The fused perception of several sensors with different points of view is highly profitable for navigation since it reduces blind spots and ensures robustness. Even better, the fused perception of sensors of different modalities makes the system more robust and, at the same time, enables more detections under a wider range of operating conditions. Therefore the use of such perception frameworks for ADS could solve issues of sensing failures and perception robustness that have been mentioned in section 1.1. Fig.8 and fig.9 exemplify grid fusion with different sensors of 2 different modalities. Fig.8 illustrates the the perception process that has been deployed on the project *IRT Naoelec – Light Vehicle Demonstrator*. The vehicle is equipped with 1 LiDAR and 5 ultrasonic range sensors. In this exemple in simulation, none of the sensor perceives the whole scene on its own. However the fusion of their detections enables the generation of a Dynamic Occupancy Grid that represents accurately the entire scene. As it is visible on the fused instantaneous grid, the confidence in free space



Figure 9 – Real world example of fused perception with the *urban light vehicle prototype*. The Dynamic Occupancy Grid gathers observations from different sensors and sensing modalities to accurately represent the scene.

observations depends on the sensing modalities and eventual overlapping observations. This is an accurate modeling of the specificities of each sensor and their combined perception, this information can then be filtered and used for planning. Fig. 9 provides another example of fused perception with the same vehicle but this time in a real world scene. The presented Dynamic Occupancy Grid gathers observations from the different sensors and sensing modalities to accurately represent the scene for further use in planning.

Scalability As Dynamic Occupancy Grids rely on a fixed-sized grid and a fixed amount of particles, this format is agnostic of the complexity of the scene. If one obstacle is a distribution of probabilities of occupancy over cells of a grids and particles, same are a hundred obstacles. Also as the distribution on cells and particles is not related to actual objects, this format does not require neither instance neither semantic segmentation of objects that can be extremely difficult in crowded environments.

Fig. 10 presents the example of a dense crowd with unstructured motion that surrounds a car. This scene has been generated in *Augmented Reality (AR)* on sensor data and recorded on board of the actual *experimental automated car*. Thanks to the LiDAR sensors, using the perception suite of *GEOG* and *CMCDOT*, the Dynamic Occupancy Grid provides a satisfactory overview of the scene in spite of its high complexity. The density of this crowd would make instance segmentation extremely challenging that is why the grid's indirect representation of objects is very profitable in this example. In this way, Dynamic Occupancy Grids offer a unique scalability and robustness to complexity that could enable navigation of autonomous vehicles in crowded environments such as city centers and shared spaces. As it has been mentioned in section 1.1, these *Operational Design Domains (ODDs)* are still out of reach of current *ADS*.

So the Dynamic Occupancy Grid appears to be a promising perception format. The numerous examples that have been presented in this section reveal its relevancy in various envi-

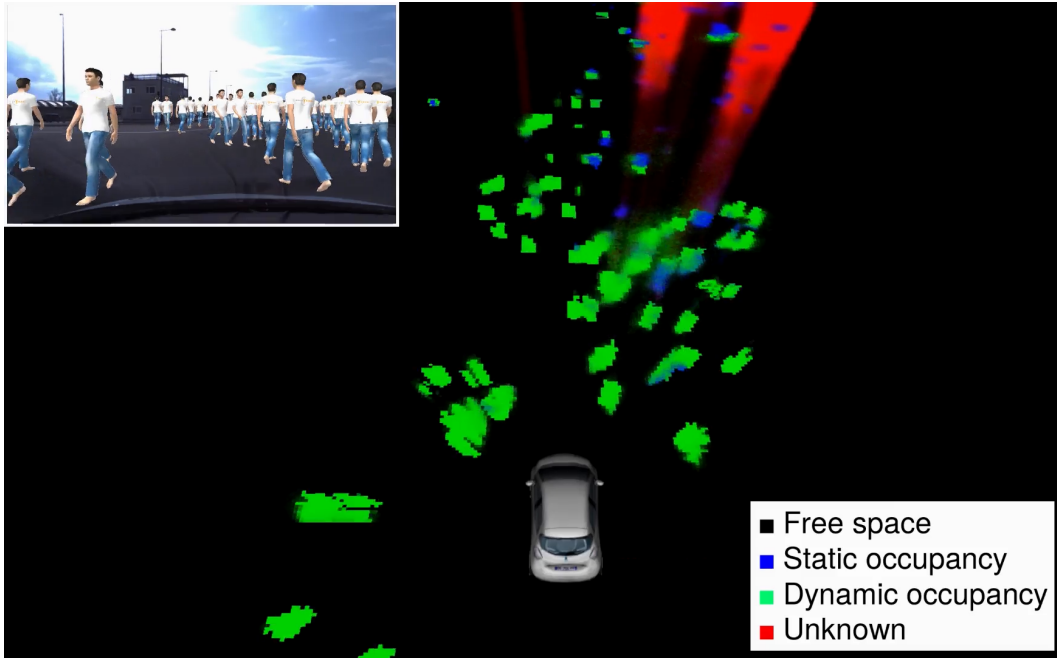


Figure 10 – AR generated example of a dense crowd with unstructured motion surrounds the *experimental automated car*. In spite of the high complexity of this environment, the Dynamic Occupancy Grid provides a very satisfactory overview of scene.

ronments, different applications and on several vehicles. This demonstrates the versatility of this tool. It has also been shown that, if they were used for collision avoidance, Dynamic Occupancy Grids have the potential to leverage the current limitations of nowadays autonomous vehicles. Yet these grids are rarely used for local planning. It seems that the main barrier to this are the technical difficulties of processing this heavy data format and interpreting its indirect representation of obstacles.

2.4 Massive parallel computations to process Dynamic Occupancy Grids

On the one hand, feature-based perception models have the advantage to produce very concise representations of the scene. In terms of data, a perceived object, for example a vehicle in a urban scene within the ETSI standards [ETS19], can be represented with about 30 numbers, the observation confidence, its identification number, its time stamp, its relative position and confidence, its bounding box, its speed, its acceleration and its semantic class. In any case, it is only a few integers and floating-point numbers to represent an object. This is less than 100 bytes for an object. Even in the case of dense traffic in urban intersection, with in a worst case 100 objects, this could be 10kB a little amount of data to process.

On the other hand, modeling the same urban intersection with a Dynamic Occupancy Grid would require a grid that covers at least about 40m ahead of the ego vehicle, about 15m behind it and possibly 20m on each side. An accurate estimation of the positions and sizes of the other vehicle would require to use a 10cm resolution. Then such a grid would contain 220000 cells, each providing occupancy, observability and classification information.

Moreover it requires an equivalent data size for particles to represent motion in this scene. So this Dynamic Occupancy Grid would represent about 3MB, this is orders of magnitude larger than the size of the feature-based representation.

Processing this amount of data with classical computing approaches would be very computationally demanding. Such computations would not be feasible under real-time and embedded hardware constraints such as in automotive applications. However occupancy grids have the advantage of a fixed-size, spatially dense and regular structure. All cells contains the same data and they are independent from each other apart of the vicinity relationship. The list of particles which is used to represent motion has the same properties. With such properties, computations on Dynamic Occupancy Grids can be very efficiently parallelized. So massive parallel computing can be used on Dynamic Occupancy Grids and it has the potential to leverage the computational cost of their applications to navigation.

Massive parallel computing is a current trend in algorithmic design that considers the distribution of computations in numerous concurrent operations while traditional approaches mostly consider sequential operations. When executed on dedicated hardware, massive parallel computing can provide dramatic acceleration of computations and it enables computations that could not be reasonable otherwise. Initially designed for video processing, Graphics Processing Units are very common hardware that can execute massive parallel computing for other purposes than graphics processing.

Fig.11 depicts the distributions of resources (transistors) on a CPU chip versus a GPU chip. It reveals that in the GPU architecture more resources are allocated for data processing cores. But, as caching and flow control are minimal in the GPU architecture, synchronization and sharing memory between processes is more demanding than on CPU. This is the reason why [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#) achieves highly efficient computations, as long as they can be parallelized and do not require significant memory transfers.

GP-GPU has been popularized by [NVIDIA Compute Unified Device Architecture \[L08\] \(CUDA\)](#). It is commonly used for deep learning applications but conceals a high potential for other applications. As mentioned in the previous paragraph, Dynamic Occupancy Grids are suitable for massive parallel computations. Software such as the [GEOG\[RPNL17\]](#) and the [CMCDOT\[RNL15\]](#) have demonstrated that processing Dynamic Occupancy Grids in GP-GPU is actually extremely efficient and can even be frugal. While these 2 software could not be executed in real-time with classical programming approaches, their GP-GPU implementation achieved more than satisfactory real-time performance even on light embeddable hardware such as the NVIDIA Jetson TX2. Some operations on occupancy grids have been accelerated by about 1000 times when porting them to GP-GPU. Therefore GP-GPU is a major tool when dealing with Dynamic Occupancy Grids and it can actually enable their broader use. That is why all the occupancy grid based applications that are proposed in the present thesis (in chapters 3 and 4) rely on extensive use of GP-GPU. It is also a powerful tool with other heavy data formats. Another contribution of this thesis relies on GP-GPU to process LiDAR point clouds, it is described in chapter 5.

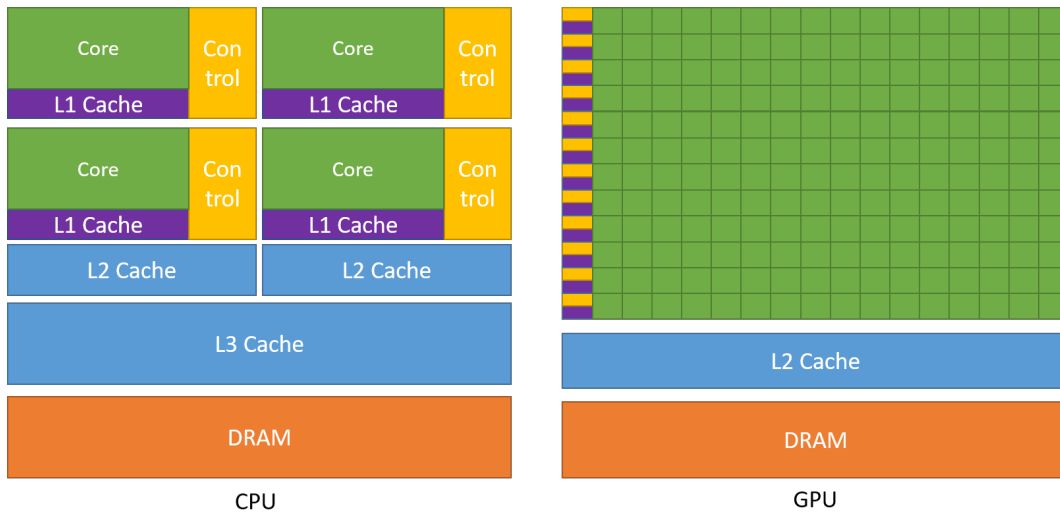


Figure 11 – Originally from [NVI23], this illustration depicts the distributions of resources (transistors) on a CPU chip versus a GPU chip. With more resources that are allocated for data processing cores, the GPU architecture is meant for highly efficient computations but it is limited regarding synchronization and memory sharing.

While GP-GPU opens new opportunities for occupancy grid based planning, it also imposes specific constraints. First, it requires to proceed with large groups of simple identical operations that can be executed concurrently with as little synchronization as possible. Also it requires to rely on fixed-size and fixed-structure indexed memory. Finally, memory transfers between CPU and GPU are slow and should be avoided. These constraints are not only affecting implementation but also the algorithmic design. So, in this thesis, all the proposed algorithms have been tailored for optimal massive parallelization. All processes rely on large number of parallelizable elementary operations. All memory structures are grids or lists of simple structures of data. In the proposed collision avoidance framework, sensors' drivers, perception and local planning are all executed on GPU. Only the control output is sent to the CPU such that memory transfers are minimal. This particular care during the algorithmic design enabled the deployment of all the software contributions of this thesis on experimental platforms, under the demanding real-time constraints on the autonomous car prototype in project [IRT NANOelec – SVA](#) and on frugal embedded hardware on the robots of projects [IRT NANOelec – Light Vehicle Demonstrator](#) and [R&D Booster MoovIT](#).

2.5 Chapter conclusion

In this chapter, it has been observed in this chapter that grid-based representations and particularly [Dynamic Occupancy Grids](#) are actively researched for their capability to model the perception of the environment of a robot. However there is little consideration of grid-based local planning in research. In the meanwhile, there are widely used and actively researched local planners that rely on configuration space graphs while such graphs are not directly observable. It has been concluded that interpreting novel grid-based perception for-

mats into configuration space graphs would enable to get the best out of both approaches. The potential benefits of [Dynamic Occupancy Grids](#) for collision avoidance have been listed. These expected benefits have motivated the development of the Dynamic Occupancy Grid based collision avoidance that is proposed in this thesis. Finally [GP-GPU](#) has been identified has a tool to leverage the computational cost of processing occupancy grids and enable their broader use in planning.

Chapter 3. Proposed collision avoidance framework

3.1 Chapter outline

This chapter introduces a new collision avoidance system that is proposed within this thesis. Section 3.2 [Need for a new Collision Dectector to interpret Dynamic Occupancy Grids](#) explains that the proposed avoidance solution intends to combine the promising approaches of perception with [Dynamic Occupancy Grids](#) and sampling-based planning in order to overcome the current limitations of [Automated Driving Systems \(ADS\)](#). Section 3.3 [Review of existing Dynamic Occupancy Grid based collision avoidance solutions and originality of the proposed approach](#) details the proposed approach and clarifies its originality with respect to the related works, it consists of a stochastic particle-based representation of obstacles, which differs from more common object-based representations. The [Predictive Collision Detector \(PCD\)](#), the core module of the avoidance framework that combines particle-based occupancy prediction and stochastic grid-based collision detection is described in the section 3.4 [Methodology of the proposed Predictive Collision Detector \(PCD\)](#). Then the deployment of the proposed collision avoidance system on 3 different platforms is presented along with experimental results in section 3.5 [Deployment and experimental results of PCD based collision avoidance](#).

The contributions that are presented in this chapter are :

- a new [Predictive Collision Detector \(PCD\)](#) and its GP-GPU implementation
- a collision avoidance solution using this detector
- simulated and experimental results of the method

These contributions have been published in :

Thomas Genevois, Lukas Rummelhard, Anne Spalanzani, and Christian Laugier. From probabilistic occupancy grids to versatile collision avoidance using predictive collision detection. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023

3.2 Need for a new Collision Dectector to interpret Dynamic Occupancy Grids

It has been noticed in section 1.1 that current limitations of [Automated Driving Systems \(ADS\)](#) could be leveraged by state of the art perception techniques. As it has been observed in section 2.3, such an approach, [Dynamic Occupancy Grid](#) based perception conceals high potential for automated drive. However, as seen in section 2.2, grid-based models are rarely used in planning, especially in local planning. In the meanwhile, sampling-based planning is a flourishing approach in local planning that relies on investigating the connectivity of the configuration space. The limit of such planners is that the configuration space is not directly observable.

Therefore, in order to get the best out of both approaches, this thesis proposes a software

module, a [Collision Detection](#) that interprets [Dynamic Occupancy Grids](#) into a configuration space representation. As this module processes not only probabilities of static occupancy but also of predicted motion, it is referred to as [Predictive Collision Detector \(PCD\)](#). The originality of this collision detector is that it is tailored for Dynamic Occupancy Grids. Also the [PCD](#) is designed for optimized computations on [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#) such that it satisfies real-time constraints even on frugal hardware.

The [PCD](#) is designed and deployed with the [Conditionnal Monte-Carlo Dense Occupancy Tracker \[RNL15\] \(CMCDOT\)](#) perception suite [[RPNL17](#), [RNL15](#)] and a sampling-based [Model Predictive Control \(MPC\)](#) planner [[DCJC08](#)]. The resulting collision avoidance framework captures the rich description of obstacles that modern occupancy grid perception provides. As a result, it achieves versatile collision avoidance, under uncertainty and even in complex scenarios.

3.3 Review of existing Dynamic Occupancy Grid based collision avoidance solutions and originality of the proposed approach

3.3.1 Related works on Dynamic Occupancy Grid based collision avoidance

Several recent research works already use [Dynamic Occupancy Grids](#) to achieve collision avoidance. [[LCB20](#)] used Dynamic Occupancy Grids with artificial potential fields and velocity synthesis to navigate a robot in an environment with static obstacles and humans. [[MCT⁺17](#)] makes use of Evidential Grids and clothoid sampling of trajectories to achieve collision avoidance in urban road scenarios. [[LK19](#)] and [[SCLC20](#)] are both using Predictive Occupancy Maps for collision avoidance in structured road environments. The former samples trajectories in acceleration, the latter uses RRT* as local planner. Unfortunately these methods do not consider uncertainties in the dynamic part of the environment. They do not consider the uncertainty in the existence of [agents](#). Only [[SGSM⁺22](#)] considers probability of existence of agents. This work uses Bayesian Dynamic Occupancy Grid with a MPPI planner to mitigate the collision damage in road scenarios. However, as the previously mentioned works, it uses a simple prediction of the motion of the obstacles and does not consider uncertainty of the prediction. It does not consider the probabilistic distribution of motion that particles of the Dynamic Occupancy Grid express. [[FSL09](#)] is one of the rare works that relies on advanced probabilistic prediction of the motion of obstacles. It combines a probabilistic occupancy grid with a list of moving obstacles whose motion prediction is done with Hidden Markov Models and local planning has been done with RRT. Yet it is limited to simple shapes and motion models of the ego-vehicle and the obstacles. Also, as the above-mentioned approaches, it relies on identified and instantiated agents and this does not cover the uncertainty which is modeled by the motion particles that do not correspond to actual objects.

The methods that have been listed in this review of the state of the art propose a broad integration of perception and collision avoidance that is applied with an occupancy grid based perception under realistic observability constraints. However all these methods extract ob-

jects from the occupancy grid in order to model the dynamic part of the environment. Then they operate only on objects and do not capture the dense probabilistic distribution of occupancy and motion that [Dynamic Occupancy Grids](#) convey. This comes to be a limitation when the object segmentation is uncertain, for example in complex unstructured scenes or when limited sensing capabilities or inference provide uncertain information. Also, while occupancy grids are a versatile manner to represent the environment of a robot, most of these methods rely on assumptions on the structure of the environment. So that they become tailored for a specific environment and do not preserve the versatility which is a major advantage of occupancy grids.

3.3.2 Proposed approach for collision avoidance with probabilistic particle-based representation of obstacles

In order to preserve the benefits of the rich [Dynamic Occupancy Grid](#) environment representation at the elementary occupancy level, this thesis proposes to push it forward with a particle-based approach until it is provided to the local planner. The proposed approach, which is illustrated in [fig.12](#), relies on motion particles for occupancy prediction. The occupancy in every cell with observed motion is split in particles that correspond to the set of actions that an agent with the corresponding position and speed could do. Then the sub-particles are propagated in time with an agent motion model that considers the corresponding action. This approach preserves the probabilistic and elementary representations of occupancy and motion.

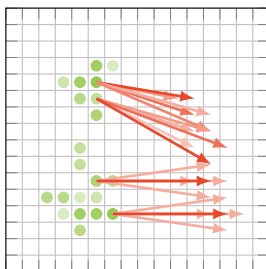


Figure 12 – Illustration of the approach for the particle-based occupancy prediction. It considers a probabilistic distribution of all the actions that agents could do. This preserves a probabilistic and elementary representation of occupancy and motion.

After considering this probabilistic distribution of actions that agents could do, the result of the occupancy prediction is reported on a grid representation. Finally, when the planner explores the configuration space, it produces sampled ego [configurations](#). These sampled [configurations](#) are projected on the predicted occupancy grid to compute the collision risk and return this information to the planner. All of this is achieved at elementary occupancy level, without using any object representation. No other research work in the studied state of the art operates at such a microscopic level. This elementary representation of obstacles provides a direct expression of the uncertainties that are modeled in the occupancy grid and it also leads to a different paradigm of collision detection and avoidance.

Comparable works in the state of the art rely on assumptions on the structure of the environment to simplify the definition of the collision risk. This is illustrated by the images in fig.13 that depict the disjunction of the collision risk definition that depends on relative positions of vehicles in a multiple lane scenario in [SCLC20] and [LK19]. This disjunction restricts and divides the **Operational Design Domain (ODD)** of the corresponding methods, it reduces the possible use cases and introduces fuzzy ODD transitions. In the meanwhile, the proposed collision detection operates without making any assumption, neither on the environment nor on the moving elements in it. It bears a unified generic risk definition, without any disjunction. This allows the proposed collision avoidance to handle a wide diversity of scenarios.

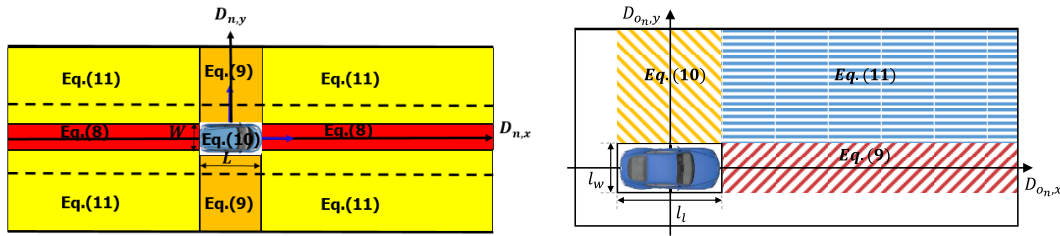


Figure 13 – Taken from [SCLC20] and from [LK19], these images respectively illustrate how these works propose a case disjunction in the definition of collision risk depending on the relative positions of vehicles in a multiple lane scenario. The risk definition that is proposed in the present thesis has no such disjunction, it is more generic.

Also related works use explicit definitions of the collision risk. Fig.14 presents 2 images from [MCT+17] and from [LK19]. The first scheme illustrates that [MCT+17] relies on an explicit definition of safety distances. The other graph plots the explicit risk function that is used in [LK19]. In the meanwhile, the present thesis proposes a different paradigm for collision detection. It defines the collision risk at particle level. So a broad and stochastic comprehension of the concept of collision risk implicitly emerges from particle-based predictive collision detection.

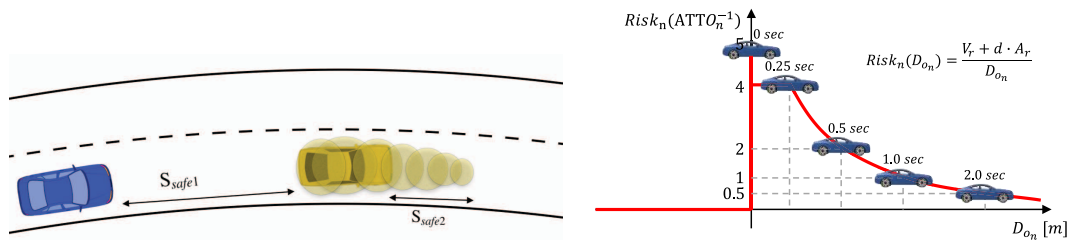


Figure 14 – Taken from [MCT+17] and from [LK19], these images illustrate explicit definitions of safety distances and risk function in these works. In the meanwhile, the present thesis proposes a particle-based definition of the collision risk that implicitly induces similar results, in a broad and stochastic approach.

Moreover the proposed approach can handle advanced probabilistic occupancy prediction models and probabilistic collision detection with complex ego-motion and collision models. Thanks to this broad comprehension of the collision risk, the proposed approach is versatile

in the sense that it could apply to a wide range of applications. This is also a matter of robustness since it guarantees that the software can operate under most circumstances. It is shown, in section 3.5.3, that the particle-based approach does not prevent this solution from generating advanced macroscopic driving behaviors and that it compares with the state-of-art of section 3.3.1 while being more versatile and handling complex scenarios.

To sum up, the proposed approach mainly differs from the related state of the art by its elementary representation of obstacles. This enables a faithful representation of the uncertainties that are expressed in the [Dynamic Occupancy Grid](#) throughout occupancy prediction, collision detection and collision avoidance. This also leads to a broad comprehension of the notion of collision risk that makes the proposed collision avoidance versatile.

3.4 Methodology of the proposed Predictive Collision Detector (PCD)

3.4.1 PCD framework overview

The proposed collision framework is built around the [PCD](#) that makes the interface between a perception system that generates a [Dynamic Occupancy Grid](#) (as defined in section 2.3.1) and a planner with a sampling-based approach, such as sampling based MPC [DCJC08], state lattices [PK05], MPPI [WDG⁺16] or RRT[LK99]. These planners, associated with a motion model of the ego-vehicle, provide a list of configurations {2D position, heading, time} for which they request the probability of collision for the ego vehicle standing in the configuration. Then the inputs of the [PCD](#) are a [Dynamic Occupancy Grid](#) and a list of [configurations](#) and its output is the collision risk associated to each [configuration](#). Additionally, if provided with [trajectories](#), the [PCD](#) can output the expected [Time To Collision \(TTC\)](#) for each [trajectory](#). Fig.15 provides an overview of the proposed collision avoidance framework, from the perceived [Dynamic Occupancy Grid](#) until the planner selects a trajectory. The [PCD](#) is at the core of this process. Fig.16 details the flow of data through the different processes of the framework. The proposed [PCD](#) consists of the 3 following modules :

- The occupancy prediction that projects the current occupancy grid along the motion particles and estimates the evolution of the occupancy grid over the next seconds. It computes the predicted occupancy grid.
- The collision detection that investigates the predicted occupancy grid at the [configurations](#) that the planner requests and returns the probability of collision for each [configuration](#).
- The optional [TTC](#) estimation uses the cumulated probabilities of collision over sequences of [configurations](#) in order to compute the expected [TTC](#) on each [trajectory](#). This is used for planners, such as [DCJC08], that require a risk estimation over a whole trajectory, but may not be needed with other planners.

These modules manipulate large arrays of data but perform simple parallelizable operations. Therefore, as mentioned in section 2.4, [GP-GPU](#) is used to accelerate these computations thanks to a massive parallelization. This allows to meet the real-time constraints that

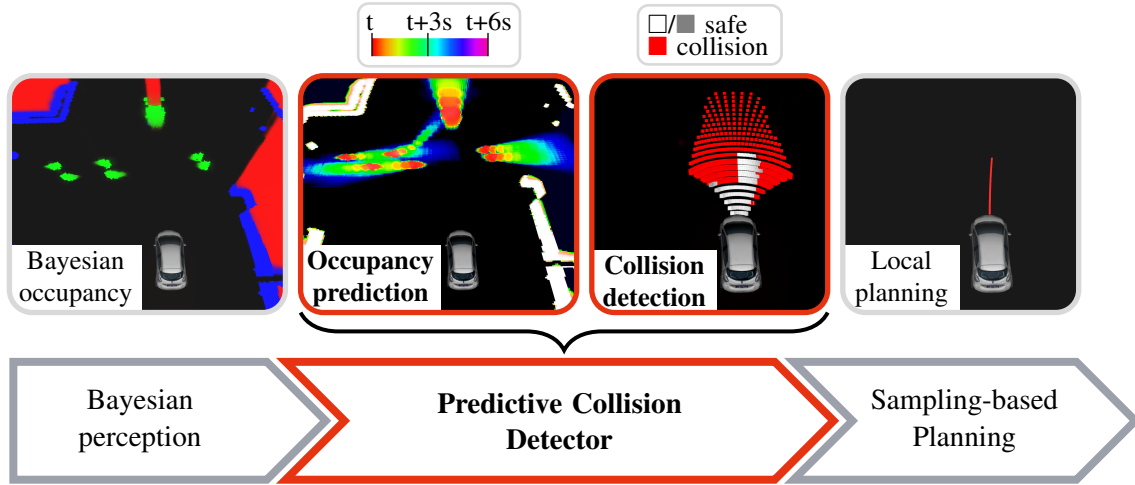


Figure 15 – This scheme illustrates the major steps of the proposed collision avoidance framework. The *PCD* is at the core of this process.

apply to collision avoidance. Throughout the whole process, the parallelization goes along different data to optimize each computation.

3.4.2 Occupancy prediction

The occupancy prediction is the module that projects the perceived motion over time in order to generate an occupancy grid for each sampled time interval. As explained in section 2.3.1, the *Dynamic Occupancy Grid* represents the perception of motion by a list of motion particles i.e. bits of occupancy that have followed a consistent motion over the previous frames. The particles do not refer to an object segmentation or instantiation. In case the perception does not directly provide particles but a velocity grid, it can be regarded as a sparse array and its non zero elements as particles in order to apply the same approach. As all particles are independent, the following algorithm is designed to be parallelized over the list of particles.

Each particle has a speed vector and carries a probability of occupancy that represents the probability of existence of a moving obstacle in the cell in which is the particle. Let t_0 be the current time, Δt the time step and M the number of time intervals. Let \mathcal{P} be a particle with a speed $V_{\mathcal{P}}(t_0)$ and an occupancy probability p . Its position is $X_{\mathcal{P}}(t_0)$. Then the occupancy prediction investigates the evolution of \mathcal{P} considering the different actions that an agent could execute if it were in place of \mathcal{P} . In the proposed approach, a prediction model is defined by \mathcal{U} , a set of actions that the agent could do, $P(u)$, the likelihood of each action $u \in \mathcal{U}$ and $X_{\mathcal{P}}(t) | u$ the future position of the particle if action u is realized. Then at t_0 , \mathcal{P} is split into a set of independent sub-particles $\{\mathcal{P}_u, u \in \mathcal{U}\}$. Each \mathcal{P}_u propagates its occupancy probability, p_u , it is given by (1). This definition guarantees that the occupancy probabilities of all sub-particles are equal and that the combined occupancy probability is equal to the one of the initial particle given that $\sum_{u \in \mathcal{U}} P(u) = 1$. p_u is reported on the prediction grid for the

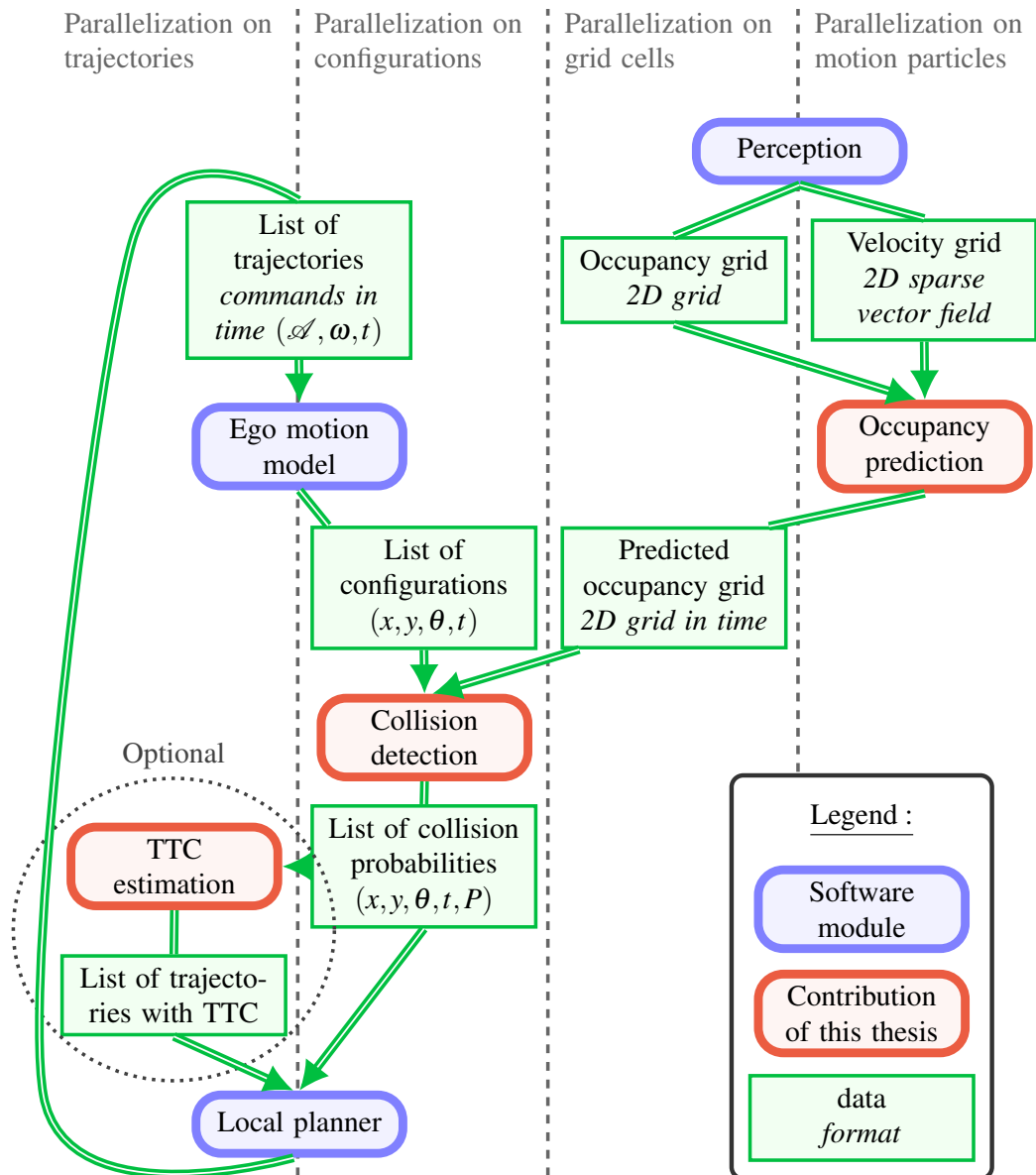


Figure 16 – Design of the software framework of the proposed collision avoidance system. The different processes are presented along with the data that they exchange. The columns refer to the possibilities of parallelization in the implementation of the processes.

next time steps at the corresponding $X_{\mathcal{D}}(t) | u$.

$$\forall u \in \mathcal{U}, p_u = 1 - (1 - p)^{P(u)} \quad (1)$$

Let \mathcal{C} be a cell of the predicted occupancy grid. Let $O_{\mathcal{C}}(t)$ be its probability to be occupied by an obstacle during $[t, t + \Delta t]$. The purpose of the occupancy prediction is then to compute $O_{\mathcal{C}}(m\Delta t)$ for each time interval $[m\Delta t, (m + 1)\Delta t]$ for $m \in \llbracket 1, M \rrbracket$. Before reporting the occupancy that the particles convey, for each $t = m\Delta t$, $O_{\mathcal{C}}(t)$ is first initialized with the static occupancy that the occupancy grid provides. And after $O_{\mathcal{C}}(t)$ is increased with each sub-particle that falls in \mathcal{C} during the corresponding time interval. This is expressed by (2) that formalizes that the cell is occupied if there is already an obstacle in it or if a sub-particle brings an obstacle along. The parallel implementation of (2) must be done carefully because the threads of different sub-particles that fall in the same cell are likely to attempt to modify the value of this cell simultaneously, the reporting on the grid should be done in a manner that avoids concurrency.

$$O_{\mathcal{C}}(t) \leftarrow 1 - (1 - O_{\mathcal{C}}(t))(1 - p_u) \quad (2)$$

For a first implementation, a kinodynamic unicycle prediction model is considered to represent the motion of the **agents**. So the action set \mathcal{U} consists of tuples of acceleration/deceleration and steering speed with a uniform distribution between the worst cases of acceleration and steering. For a fine sampling of the actions, $|\mathcal{U}| = 100$. This proposed prediction model considers that all actions u to be equally probable then $\forall u \in \mathcal{U}, P(u) = \frac{1}{|\mathcal{U}|}$. Finally $X_{\mathcal{D}}(t) | u$ is given by application of the kinodynamic unicycle model. The purpose is not to obtain an accurate prediction but a short term conservative prediction that will serve the collision detection. This is why the whole range of actions is considered, including worst cases of acceleration, deceleration and steering. With this prediction model, as sub-particles are considered to be equally probable, (1) can be simplified into (3). Fig. 17 presents an example of a predicted occupancy grid that has been obtained with this model.

$$\forall u \in \mathcal{U}, P_u = 1 - \sqrt[|\mathcal{U}|]{1 - p} \quad (3)$$

This approach of dividing initial particles into sub-particles that spread acts similarly to a Monte-Carlo filter. It tends to filter out lonely initial particles that spread little occupancy probability in a large area while it reveals consensual motion of groups of particles that generates a wide distribution of predicted occupancy. The division in many sub-particles leads to a high number of computations. The occupancy grid might provide a large list of particles, about 100 thousands. There are then several millions of sub-particles. Yet, as the operations on the sub-particles are individually simple and independent from each other, this approach leads to an efficient implementation in **GP-GPU**.

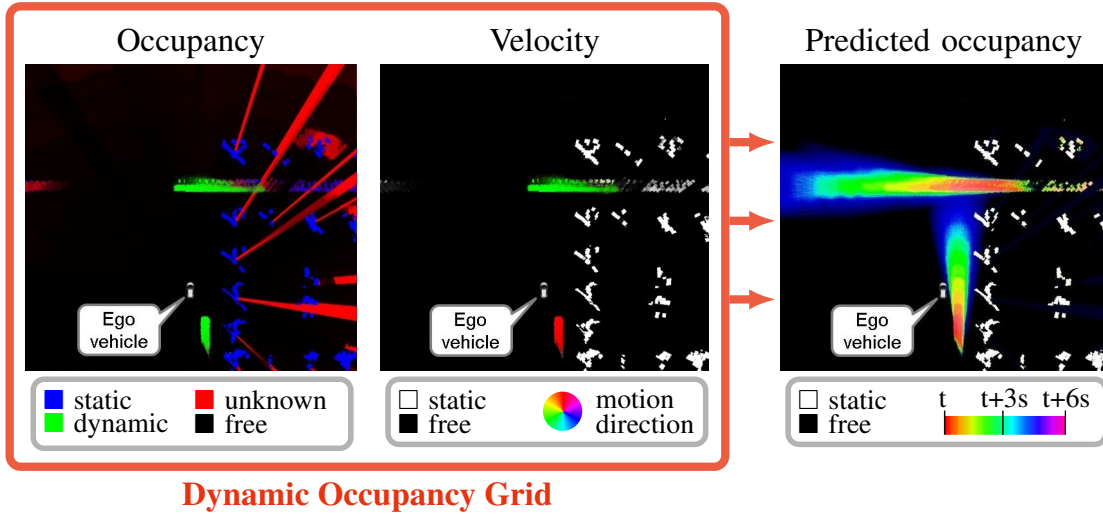


Figure 17 – This figure presents a Dynamic Occupancy Grid and the resulting predicted occupancy grid after applying the proposed occupancy prediction process. NB : The color scheme which used to represent the occupancy over time only represents the dominant moment of occupancy in a position. This is not an exact representation of the predicted occupancy which is actually a 3D array (2 Cartesian coordinates plus time).

3.4.3 Collision detection

The local planner, a sampling-based planner, generates samples of ego-trajectories. Then it needs an estimation of the collision risk to select a safe trajectory. In this purpose, the collision detection is the module that computes this collision risk. As the sampled **trajectories** are interpreted as sequences of **configurations** thanks to an ego motion model, the collision detection investigates the predicted occupancy grid at these configurations. Then, for each **configuration**, the Collision Detector returns the probability of a collision if the ego vehicle happens to be in that configuration at the corresponding time. As all configurations can be investigated independently even across different trajectories, the algorithm of the Collision Detector is designed to be parallelized over the list of configurations.

Let q be a **configuration** of the ego vehicle, it is defined by the 2D coordinates, a heading and a time. The shape of the ego vehicle is discretized, at the same resolution than the occupancy grid or finer. Let \mathcal{S} be the set of vectors that define the discretized shape of the ego vehicle with respect to its reference frame. Let T_q be the transformation matrix from the ego vehicle reference frame to the grid frame for a given q . Let $O(X, t)$ be the occupancy probability of the cell of the predicted occupancy grid in which falls the given 2D position X and for the time interval corresponding to t . Then we use T_q to perform the forward projection of \mathcal{S} on the predicted occupancy grid. The probability of a collision if the ego is in q , $P_{\text{coll}}(q)$, is defined by (4) as the probability that there is no obstacle in none of the cells that are beneath the ego shape.

$$P_{\text{coll}}(q) = 1 - \prod_{\vec{v} \in \mathcal{S}} (1 - O(T_q \vec{v}, t)) \quad (4)$$

(4) assumes that the event of collision is defined as the presence of an obstacle and a

discretized portion of the ego vehicle in the same cell during the same time interval. Then the definition of O requires to interpret the uncertainty of the [Dynamic Occupancy Grid](#) in terms of probability of existence of an obstacle with respect to a surface and a time interval. In the case of Bayesian occupancy grid as provided by the [CMCDOT](#), the grid defines the probability of having observed a cell and the probability of existence of an obstacle inside it given that it has been observed. O is the combination of these probabilities, unobserved state resulting in a prior distribution. Although the prior has to be specified carefully. Applying a prior of 0.5 probability of occupancy per cell is a common practice but it would lead to $P_{\text{coll}}(q) = 1 - 0.5^{|\mathcal{S}|} \approx 1$ if q is in an unknown area, unknown would then be perceived almost as occupied areas. Therefore the prior value should be normalized for a given area and time interval so that the definition of collision does not depend on the tessellation size in space and time.

This method allows to use even complex shapes of the ego vehicle. The massive parallelization over the [configurations](#) leads to a very efficient computation. As the collision detection is done on the predicted occupancy grid, it takes into account the estimated motions in the environment. The output of this module, the collision probability associated to each q , is the required input for most sampling-based planners.

3.4.4 Time To Collision estimation

Some planning techniques, like the sampling-based [MPC](#) [[DCJC08](#)] require an estimation of collision risks over [trajectories](#) instead of a collision risk per [configuration](#). For this, the proposed method provides the expected [TTC](#) as a risk measurement for each trajectory. The algorithm of [TTC](#) estimation is designed to be parallelized over the list of [trajectories](#).

Let $q_{\llbracket 0, n \rrbracket}$ be a trajectory i.e. a sequence of [configurations](#). The probability that a collision occurs in the i first configurations of $q_{\llbracket 0, n \rrbracket}$ is given by (5).

$$P_{\text{coll}}(q_{\llbracket 0, i \rrbracket}) = 1 - \prod_{j=0}^i (1 - P_{\text{coll}}(q_j)) \quad (5)$$

(5) must be considered carefully. It is valid only if $P_{\text{coll}}(q_j)$ is defined as the probability of a collision to occur within the corresponding time interval and normalized as such otherwise the definition of $P_{\text{coll}}(q_{\llbracket 0, i \rrbracket})$ would depend on the time step. Let $P_{1^{\text{st}} \text{ coll}}(i, q_{\llbracket 0, n \rrbracket})$ be the probability that the first collision over the trajectory $q_{\llbracket 0, n \rrbracket}$ occurs in configuration q_i . So it can be computed from $P_{\text{coll}}(q_{\llbracket 0, i \rrbracket})$ and $P_{\text{coll}}(q_{\llbracket 0, i-1 \rrbracket})$ according to (6).

$$\begin{cases} P_{1^{\text{st}} \text{ coll}}(0, q_{\llbracket 0, n \rrbracket}) = P_{\text{coll}}(q_0) \\ \forall i \in \llbracket 1, n \rrbracket, P_{1^{\text{st}} \text{ coll}}(i, q_{\llbracket 0, n \rrbracket}) = P_{\text{coll}}(q_i) (1 - P_{\text{coll}}(q_{\llbracket 0, i-1 \rrbracket})) \end{cases} \quad (6)$$

As the investigated trajectory covers only a limited time horizon, t_f . The probability of collision is not defined over this horizon and the [TTC](#) estimation is restricted to this horizon. That is why a final virtual configuration to each trajectory with a probability of collision of

1 at the prediction horizon is added to each trajectory. This sets t_f as the upper bound of the estimated TTC. Finally, the **TTC**, $\tau(q_{\llbracket 0, n \rrbracket})$, is defined as the expectancy of the time of the first collision in (7). This is a stochastic estimation of TTC that integrates the expression of uncertainty that the perception and the prediction convey.

$$\begin{aligned} \tau(q_{\llbracket 0, n \rrbracket}) &= \left(\sum_{i=0}^{n+1} t_i P_{1^{\text{st}} \text{ coll}}(i, q_{\llbracket 0, n \rrbracket}) \right) \\ &= t_0 P_{\text{coll}}(q_0) + \left(\sum_{i=1}^n t_i P_{\text{coll}}(q_i) (1 - P_{\text{coll}}(q_{\llbracket 0, i-1 \rrbracket})) \right) + t_f (1 - P_{\text{coll}}(q_{\llbracket 0, n \rrbracket})) \end{aligned} \quad (7)$$

Even though there is a smaller amount of trajectories to be processed than of motion particles or configurations, it is still relevant and beneficial to parallelize the TTC estimation process over the list of trajectories. As **GP-GPU** is used for parallelization, having this module computed on the GPU also saves a costly memory transfer from GPU to CPU. In the implementation, the evaluation of the cost function of the sampling-based **MPC** is also executed on GPU and parallelized over the trajectories. In the end, only the selected trajectory is sent to the CPU for it to be followed by the low level motion control. This enables to parallelize the computations as much as possible and reduces the memory transfers to very little data.

3.4.5 Sampling-based MPC planner

The proposed contribution of this work lies in the 3 modules of the **PCD** that have been described so far. The **PCD** framework is theoretically compatible with a wide range of sampling-based planning planners. Within this thesis, the PCD has been applied with a sampling-based Model Predictive Control planner. This planner is not regarded as an original contribution of this thesis. The concept of sampling-based **MPC** has been introduced in [DCJC08]. It is a convenient planning approach to integrate non-linear kinodynamics and can be used for local and global planning. Various implementations of it have been proposed for a wide range of robotic applications. The implementation that is used in the present thesis has been optimized for acceleration on GPU. Hence sequential operations have been replaced by parallel computations. As a result, the proposed planner consists of the following operations that are sequentially executed at every time step. Each of them is parallelized over the list of trajectory candidates.

1. **Generation of the command set** – At every time step, the planner generates a list of trajectory candidates with a uniform sampling of the admissible command space over the next control time horizon. In the proposed implementation, each trajectory is a sequence of acceleration and steering commands.
2. **Application of the ego motion model** – For all the trajectory candidates, the commands are integrated within a ego motion model in order to compute the geometrical trajectory. In the proposed implementation, the motion model considers kinematic

constraints, maximal acceleration and steering values and actuation delays.

3. **Collision risk evaluation** – For all the trajectory candidates, the geometrical trajectory is projected on the predicted occupancy grid in order to evaluate the collision risk. This corresponds to the collision detection and the [Time To Collision \(TTC\)](#) evaluation of the proposed [PCD](#) that have been described in sections [3.4.3](#) and [3.4.4](#).
4. **Cost function** – For all the trajectory candidates, a cost function is evaluated. Its cost renders how the trajectory matches with the navigation reference and formulates a compromise between the objective and the collision risk. In the proposed implementation, the cost function discards all trajectories whose estimated [TTC](#) is too low. Then it evaluates how the overall trajectory corresponds to a reference path and how advanced the final position of the trajectory is on this path.
5. **Executing the best trajectory candidate** – The trajectory candidate with the minimal cost is sent as a reference for the motion control. In the proposed implementation, the sequence of acceleration and steering commands is used as a reference value for PID controllers that regulate the steering and acceleration of the vehicle.

As the motion model can handle complex kinodynamics and as there is no restriction on the cost function, this approach is highly versatile. The main drawback of this approach is that it requires to make a compromise between the computational complexity and the richness of the produced trajectories. Both of them strongly depend on the resolution of the initial sampling of the command space. Though, the [GP-GPU](#) acceleration mitigates this drawback as it reduce the computation burden.

3.5 Deployment and experimental results of PCD based collision avoidance

3.5.1 Implementation and deployment of PCD based collision avoidance on a car

Experimental platforms and implementation details The proposed collision avoidance framework has been applied with a LiDAR-based application of [CMCDOT](#) perception suite and a sampling-based [MPC](#) planner that combines path following and obstacle avoidance. It has been programmed in [NVIDIA Compute Unified Device Architecture \[L08\] \(CUDA\)](#) for execution on GPU. Then this collision avoidance system has been deployed on an [experimental automated car](#). A realistic [Hardware-in-the-Loop \(HiL\)](#) simulation of this vehicle in a virtual urban environment has been designed to evaluate the performance of the proposed collision avoidance. Real world tests have also been executed.

On this robotic car, the PCD has to satisfy a challenging real-time constraint. However the efficient implementation on GPU allows to satisfy this constraint. In the configuration on this vehicle, the occupancy grid contains 350000 cells and the planner investigates a total of 25960 configurations. Despite this large amount of data, the occupancy prediction is executed in a maximum of 15ms, the collision detection in 14ms and the TTC estimation in 1ms while

the whole collision avoidance is executed at 15Hz. The program could run faster but this frequency has been chosen to match the frequency of the sensors.

Navigation target The planner that is used for collision avoidance combines avoidance and path following in its objective function. A target [path](#) is given to this planner. In some tests, the path is manually defined from a user input or a recorded path. In the other tests, a global planner based on E* [\[PS05\]](#) is used. In all cases, the constraint on the path following is much weaker than the collision avoidance. Thanks to this, the proposed navigation leads the ego-vehicle approximately on the path and performs avoidance, either leaving the path either staying on the path and acting on the longitudinal control. The path following being a minor concern in the proposed navigation, it is only evaluated indirectly within the navigation performance. The focus is made on the evaluation of the collision avoidance.

Testing environment The evaluation of the collision avoidance system involves simulation results. These are [HiL](#) simulations, it means that the software uses only information given by the emulated sensors and provides control outputs to the simulated vehicle as it would do with the actual one. It means also that the software ran on the actual computing hardware of the robot. Therefore these tests could be executed as such in real world experiments, given that the test scenario can be safely reproduced in real world. Also the simulations intend to reproduce as faithfully as possible the sensing processes and the dynamics of the vehicles. Most simulations scenarios involve pedestrians. Some pedestrians have predefined trajectories but, in more complex scenarios, interactive and realistic behaviors of groups and crowds of pedestrians have been simulated with [Simulation of Pedestrians and an Autonomous Car in Shared Spaces \[P21\] \(SPACISS\)](#). Inspired from observation of actual human behaviors, this simulator is tailored to model interactions between a car and numerous pedestrians. It considers avoidance and social interactions among pedestrians as well as reactions of pedestrians to the car resulting either from the decision of a single person either from a group behavior. Moreover randomized initialization and decisions generate a large manifold of behaviors to challenge the collision avoidance. Simulated behaviors vary from cooperative (pedestrians yield or avoid the car) to selfish (they follow their path and do not mind the vehicle).

3.5.2 Quantitative evaluation of the safety of PCD based collision avoidance

Evaluation scenario The quantitative evaluation of the proposed collision avoidance is conducted in [HiL](#) simulation with the [experimental automated car](#) considering navigation at reduced speed (maximum 30 km/h) in a shared space with pedestrians. This scenario has been chosen because it presents the most challenging dynamic avoidances. The car has to perform several avoidances that often include several pedestrians at a time. This is also a rather complex perception task since many partial occlusions occur and motion estimation and object segmentation is tricky in such a dense crowd. Also the randomly initialized complex behavior of the simulated pedestrians generates a large diversity of situations.

Safety metric This evaluation considers 2 criteria. The most important evaluation criterion is the safety one. In the low speed conditions that are considered, it is not enough to observe that no collision occur to guarantee the safety of the system. Since the speed is rather low, the ego-vehicle can quickly decelerate to full stop. So it may drive aggressively and reach near-collision situations and still evade the collision. Moreover, as the simulated pedestrians interact with the car, they might as well avoid collisions. Therefore counting collisions is not sufficient to identify all dangerous situations that occur during the execution of a test. Measuring simply the distance to the obstacles is not sufficient neither since it may be safe to drive close to an obstacle at low speed while it may be dangerous to drive fast at the same distance. For these reasons, the dynamic safety distances that are recommended by the method [Responsibility-Sensitive Safety \[SSSS17\] \(RSS\)](#) are used as a reference for safety evaluation. This approach defines the safety distances that are necessary for to avoid a collision considering the road topology, the relative speeds, the reaction delay and braking capabilities of the actors. It is a well-established standard of safety that is commonly used in the [ADS](#) research community. This approach provides a satisfactory evaluation of the driving safety even though it is generally considered as pessimistic since it considers the worst case which is very often not realistic. As a result, it is more conservative than most human drivers. According to [\[SSSS17\]](#), in the case of frontal collision risk between a car and a pedestrian, the required safety distance, d_{RSS} , depends on the ego speed, V_{ego} , is defined by (8). It considers the worst case i.e. the ego and the agent are not aware of the risk and continue to accelerate during a reaction delay, then they both decelerate. In this application, the reaction delay considers a automated drive whose reaction delay is shorter than the one of a human driver.

$$\left\{ \begin{array}{l}
 \text{Ego acceleration : } A_{ego} = 2 \text{ m/s}^2 \\
 \text{Ego deceleration : } B_{ego} = -6.1 \text{ m/s}^2 \\
 \text{Ego reaction delay : } r_{ego} = 0.3 \text{ s} \\
 \text{Pedestrian speed : } V_{ped} = 1 \text{ m/s} \\
 \text{Pedestrian acceleration : } A_{ped} = 1 \text{ m/s}^2 \\
 \text{Pedestrian deceleration : } B_{ped} = 2 \text{ m/s}^2 \\
 \text{Pedestrian reaction : } r_{ped} = 0.5 \text{ s} \\
 d_{RSS} = r_{ego} \left(V_{ego} + \frac{r_{ego} A_{ego}}{2} \right) + \frac{(V_{ego} + r_{ego} A_{ego})^2}{2B_{ego}} + \frac{(V_{ped} + r_{ped} A_{ped})^2}{2B_{ped}} + \\
 r_{ped} \left(V_{ped} + \frac{r_{ped} A_{ped}}{2} \right) \\
 d_{RSS} \approx 0.08 \text{ s}^2/\text{m} \times V_{ego}^2 + 0.40 \text{ s} \times V_{ego} + 1.31 \text{ m}
 \end{array} \right. \quad (8)$$

As visible in (8), the [RSS](#) safety distance mostly depends on the car since its speed and dynamic capabilities are higher than those of pedestrians. [Fig.18](#) illustrates this with a plot of [RSS](#) safety distance as a function of the speed of the car. Therefore, used as a reference for evaluation, it outlines mainly dangerous situations that result from the action of the car

and minimizes the effect of pedestrians in the avoidance. Still, as the purpose is to evaluate the safety which is induced by the avoidance system of the car, the situations where the car is stopped are ignored. Then a metric, Q_{RSS} , the RSS satisfaction ratio is defined for the whole execution of a test by (9), $d_{\text{car-pedestrian}}$ being the distance from the car to the closest pedestrian in a large sector in front of the car. $Q_{RSS} > 1$ signifies that the RSS recommended safety distance is observed. This metric can easily be computed for each simulated experiment that contains only pedestrians. For sake of simplification, the computation of d_{RSS} only considers the case of frontal collision risk. This is not exact ; this is the worst case. So the metric is pessimistic but guarantees the identification of all risks.

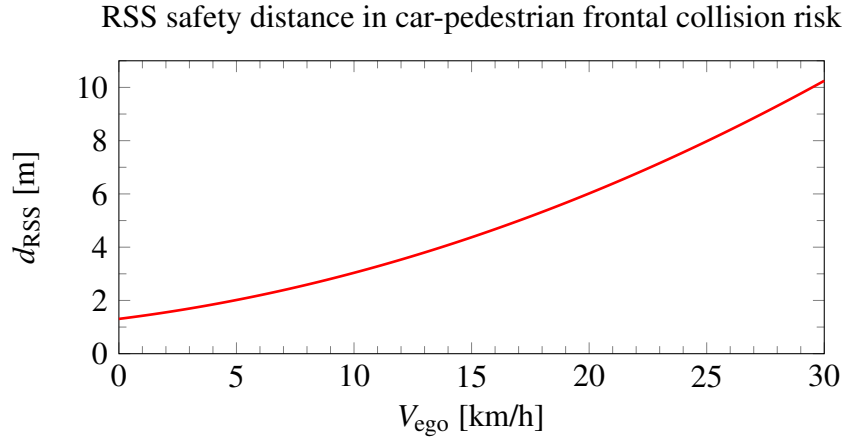


Figure 18 – Considering the risk of collision between a car and a pedestrian going in opposite directions, this graph presents the recommended **RSS** safety distance as a function of the speed of the car. This assumes the same parameters as in (8).

$$Q_{RSS} = \min_{t, V_{\text{ego}} \neq 0} \frac{d_{\text{car-pedestrian}}}{d_{RSS}} \quad (9)$$

Efficiency metric The second criterion of evaluation is the navigation performance. To assess this, the navigation duration, Δt_{nav} , is defined as the amount of time that the vehicle needs to complete the given path. Its variation when the density of obstacles increases illustrates how efficient is the avoidance.

Quantitative evaluation Using this 2 metrics, a quantitative evaluation of the navigation performance and safety has been realized in a 50 m long and 7 m wide simulated shared space, with crowds of different densities from 2 to 14 people per square meter. Pedestrians have been simulated with **SPACISS**. The proposed collision avoidance system has been used, it was free to adapt its path and speed within the road boundaries. Q_{RSS} and Δt_{nav} have been measured.

Fig. 19 presents a simulated test of navigation in a shared space with pedestrians. In this scenario, the car and pedestrians are going in opposite directions and share this 7 m wide road over 50 m. The pedestrians are randomly initialized and simulated with **SPACISS**. The

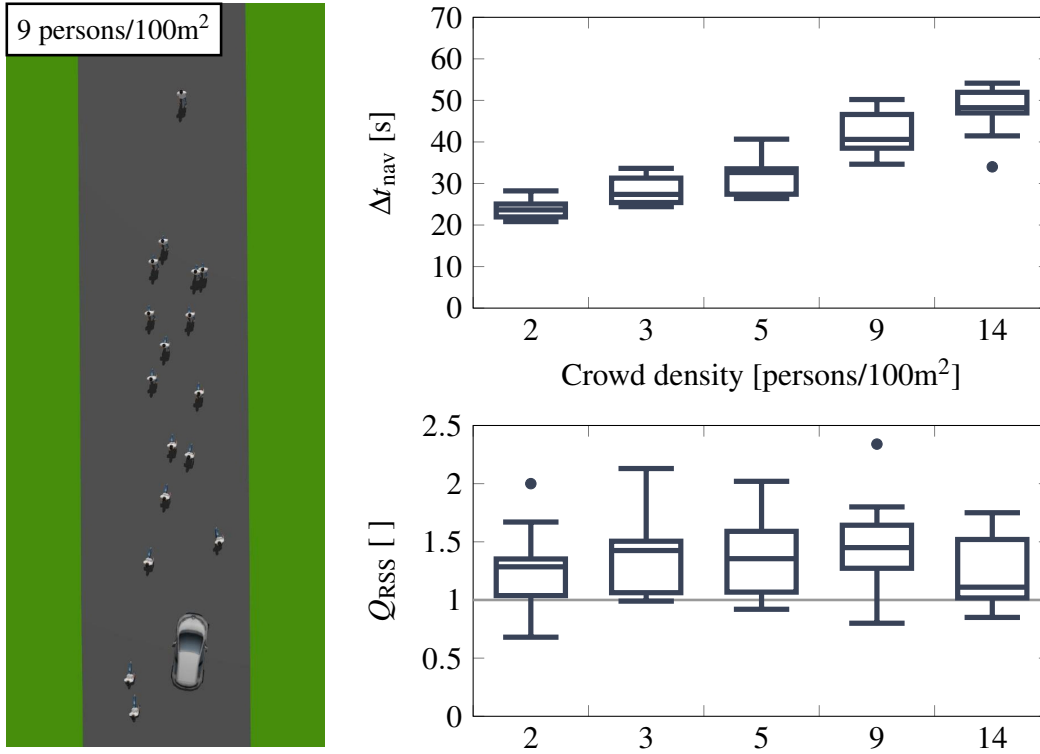


Figure 19 – Illustrations and results of a simulated test of navigation in a shared space with pedestrians. The graphs display the navigation duration, that increases with the crowd density, and the RSS satisfaction ratio, that remains stable and mostly superior to 1. Each boxplot displays the median, quartiles, extrema and outliers over 10 simulations. These results globally demonstrate that the proposed method respects the RSS safety recommendations and maintains a constant safety level despite the increasing complexity of the navigation task.

local planner is given the task to cross this crowd with the only constraint is to remain on the road. The planner is free to adjust speed and path. During 50 iterations, with different crowd densities, no collision occurred. The vehicle speed reached 25 km/h in low densities and went below 5 km/h in dense crowds. Apart of the actual crossing time, Δt_{nav} includes acceleration and deceleration phases so it does not directly reflect the vehicle speed. A graph in this figure displays the distribution of the results of navigation duration as a function of the crowd density. Δt_{nav} increases with the crowd density, proving that the planner globally adapts its speed to the complexity of navigation task. Yet, despite of this increasing complexity, a second graph displays Q_{RSS} that remains mostly stable and higher than 1 no matter the crowd density. This shows that the proposed method maintains a constant safety level despite the increasing complexity of the navigation task. Q_{RSS} being superior to 1 shows that the avoidance respects the RSS safety standards. Moreover Q_{RSS} remaining stable while the density of agents increases shows that the avoidance is able to maintain similar level of safety regardless the complexity of the scenario. This result can be explained by the fact that **RSS** and the proposed **PCD** both consider the worst cases of acceleration, braking and steering in the actions of the agents. The **PCD** encapsulates this key feature of **RSS** in a particle-based stochastic approach.

Qualitative comparison with related works Other methods [LCB20, SGSM⁺22, FSL09] of the studied state of the art considered occupancy grid based navigation in human populated environments. The chosen test scenario compares with test scenarios that have been used in [LCB20] and [FSL09]. However the testing methods differ by the sensing and perception system, the ego motion constraints and the pedestrian motion simulation. Therefore a quantitative comparison is impossible. Yet, by the density and the realist behavior of the crowd, the speed and kinodynamic constraints of the ego-vehicle, the test method which is proposed in the present thesis is more challenging. So the proposed method appears to outperform the above-mentioned works by the complexity of its operating constraints and environment. Besides, even if RSS is gaining significant interest from the research community, no related work has been found using a RSS based safety metric. This quantitative validation has demonstrated that the proposed system navigates safely, even in more complex scenarios than those considered in the related works.

3.5.3 Qualitative results of PCD based collision avoidance

Anticipating a pedestrian crossing A first simple HiL simulated example illustrates a behavior that is induced by the PCD. As a pedestrian is about to cross the road ahead of the ego vehicle, the occupancy prediction estimates the crossing time. So the probabilistic risk assessment identifies which of the ego-trajectories risk to intercept the trajectory of the pedestrian. Provided with these clues, the planner is able to decide early whether to safely cross before or to yield smoothly. Fig.20 shows the predicted occupancy grid and the estimated collision risk on the sampled trajectories. In this scenario, as the car arrives early near the intersection, the planner decides to cross before the pedestrian. This results in a safe crossing, without modifying its speed, the pedestrian reaches the zebras second, a significant safety distance is observed.

In fig.21, a first graph plots the ego-vehicle speed during this scenario. t_0 represents the date of the first observation of the pedestrian in the perception system. During 2 seconds after t_0 , the ego-vehicle slightly decelerates, probably because it considers an uncertain risk of collision with the pedestrian in long term prediction. Then it accelerates again and crosses safely. A slight variation of this scenario considers the car arriving 1s later at the intersection, it leads to a different outcome. The planner considers that, in these conditions, it is unsafe to cross first, so it decides to yield. As this decision process occurs early, a smooth deceleration can be applied.

A second graph in fig.21 displays the speed of the vehicle in this scenario. The beginning of the previous curve is superposed with this one to show that they diverge at t_0+2s . This demonstrates that the decision to continue or to yield is taken at this moment. As this is rather early, in case of yielding, a smooth deceleration can be applied. This example illustrates the benefit of the anticipation of the collision risk. The Predictive Collision Detection enables an early decision, so only smooth actions are applied. This behavior of the proposed particle-based avoidance system qualitatively compares with object-based approaches that

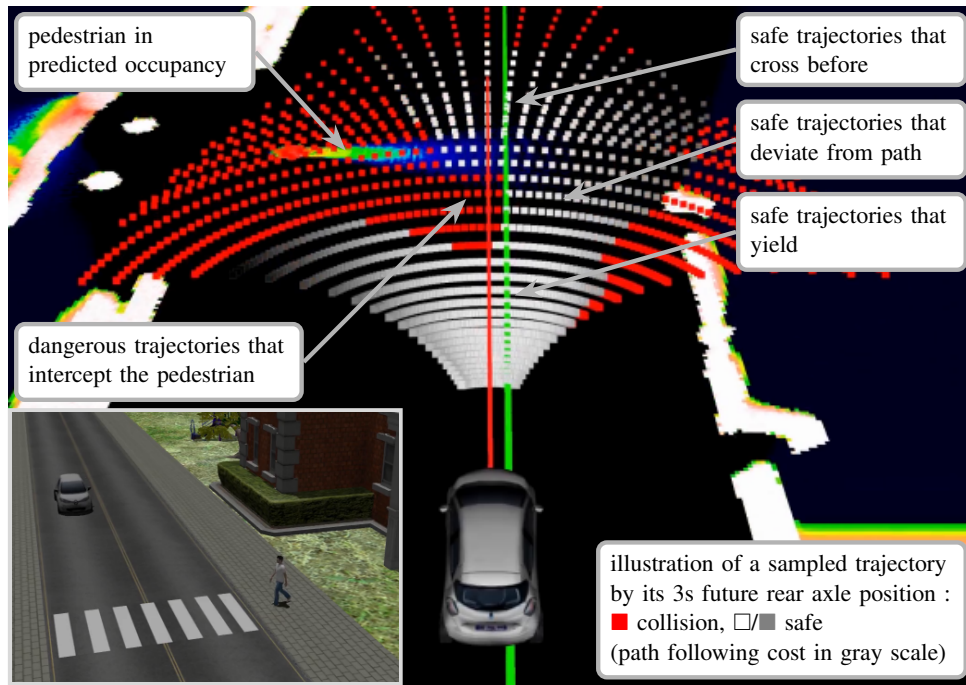


Figure 20 – In a *HiL* simulation, the ego vehicle approaches a footpath that a pedestrian is about to cross. In this example, the planner chose to safely cross before the pedestrian, without disturbing him. This scene corresponds to graph (A) in fig.21 at the date $t_0 + 2s$. The predicted occupancy and the set of evaluated trajectories with the estimated collision risk are displayed. The *PCD* is able to clearly identify the sets of trajectories that lead to a collision and the actions that lead to safe trajectories. In this, the *PCD* provides support for the decisions of the planner.

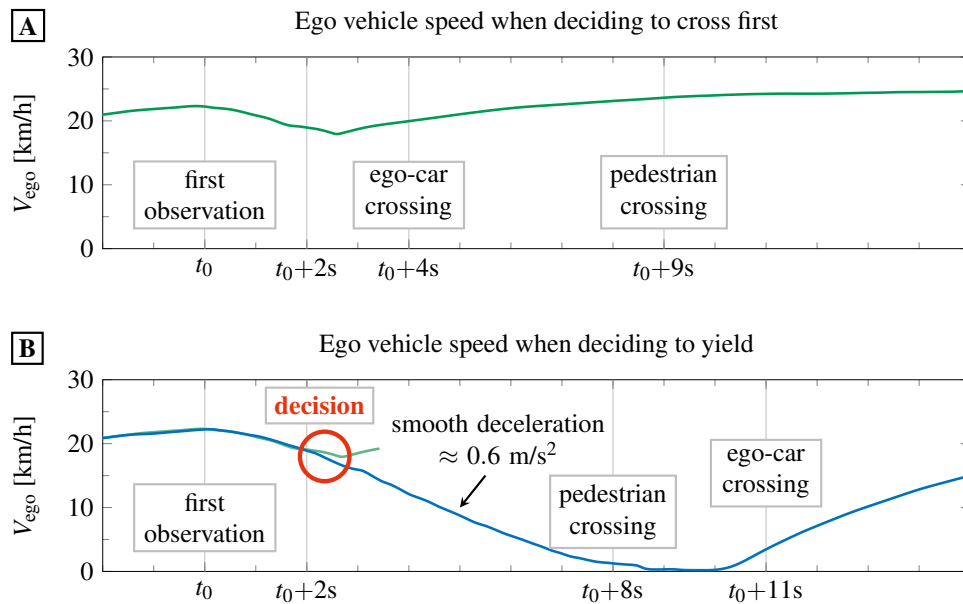


Figure 21 – Graph (A) shows the ego-vehicle speed in the simulated scenario which is depicted in fig.20. This scenario is repeated with a 1 s delay in the arrival of the car. This time, the planner decides to yield, the resulting speed is presented in graph (B). The divergence of the curves, at $t_0 + 2s$, reveals that this is the moment when the decision to continue or to yield is taken. The Predictive Collision Detection enables to anticipate the collision risk and make an early decision, so only smooth actions are applied.

have been presented in [SGSM⁺22] and [FSL09] and that consider similar tests scenarios. This test demonstrates that PCD grants the planner the ability to anticipate collision risks and to perform decisions.

Inferring safety distances during an overtaking maneuver A second example, in fig. 22, demonstrates that the PCD identifies all collision risks that arise during an overtaking maneuver. As the particle-based occupancy prediction considers the probabilities of agents to take various actions from braking, steering and accelerating, it forces the planner to observe forward, lateral and rear safety distances. Moreover the experiment shows that the safety distances that the proposed avoidance respects match with the RSS recommendations. This illustrates the accuracy of the predictive collision risk estimation and it demonstrates that the particle-based probabilistic occupancy prediction succeeds to encapsulate safety concepts that are similar to those of RSS. This behavior of our avoidance system qualitatively compares with what has been presented with similar tests scenarios in [MCT⁺17], [LK19] and [SCLC20]. As presented in section 3.3 and illustrated with fig.13 and fig.14, these works distinguish different cases of risk during the overtaking and use explicit definitions of the collision risk that depends on the relative position of the ego-vehicle to the agent. On the other hand, the method that is proposed with the PCD provides a single implicit stochastic risk definition that emerges from the particle-based approach and that is valid during the whole maneuver. This same risk definition was also operating with the pedestrian in the tests of fig.20 and with several pedestrians in section 3.5.2. So this risk definition is more versatile than those of the related works. This test demonstrates that PCD provides a generic stochastic risk estimation and that it induces reasonable safety distances in various situations.

Versatility of the PCD approach Among the state-of-art collision avoidance systems that integrate both perception and navigation under uncertainty on dynamic occupancy grids, on the one hand are [LCB20], [FSL09] that handle navigation in human-populated environments and on the other hand are [MCT⁺17], [LK19] and [SCLC20] that operate on structured multiple-lanes road scenarios and [SGSM⁺22] that considers collision mitigation in road scenarios. These solutions exploit prior knowledge of structure of the environment (some assume a multiple-lanes road structure) and of the obstacles (some assume that all obstacles are humans others that all obstacles are cars). Such simplifications leverage the complexity of computations and the constraints on perception but restrict these avoidance systems to operating domains they have been tailored for. Meanwhile, with the PCD, the same collision risk led the proposed avoidance system through the previous experiments (fig.19, fig.20 and fig.22) and the proposed collision avoidance qualitatively compares with each of the mentioned related works in their operating domain. This demonstrates the versatility of the proposed avoidance solution in ADS applications.

Real world experiments The proposed collision avoidance has also been tested in real world experiments on the [experimental automated car](#) where it demonstrated its collision

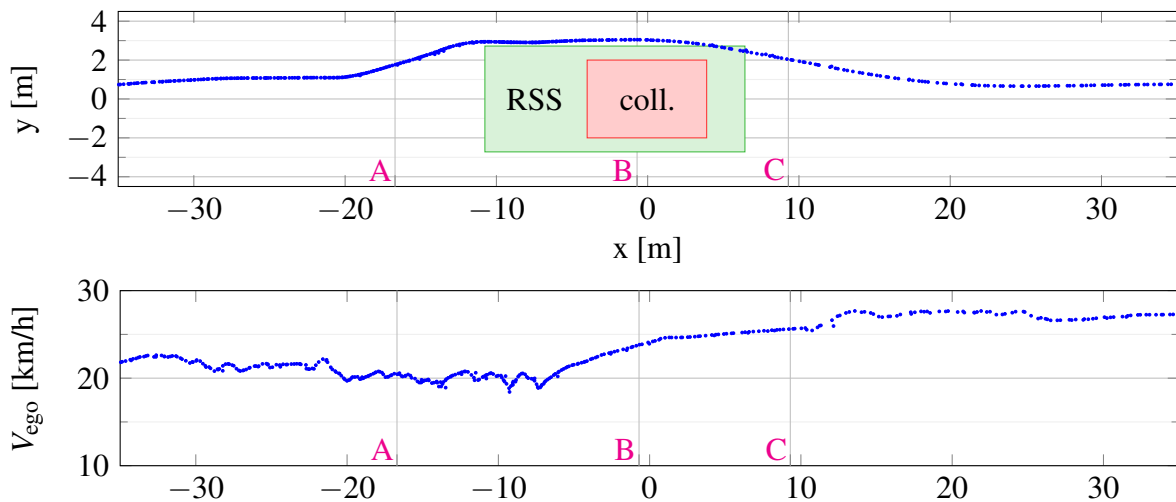
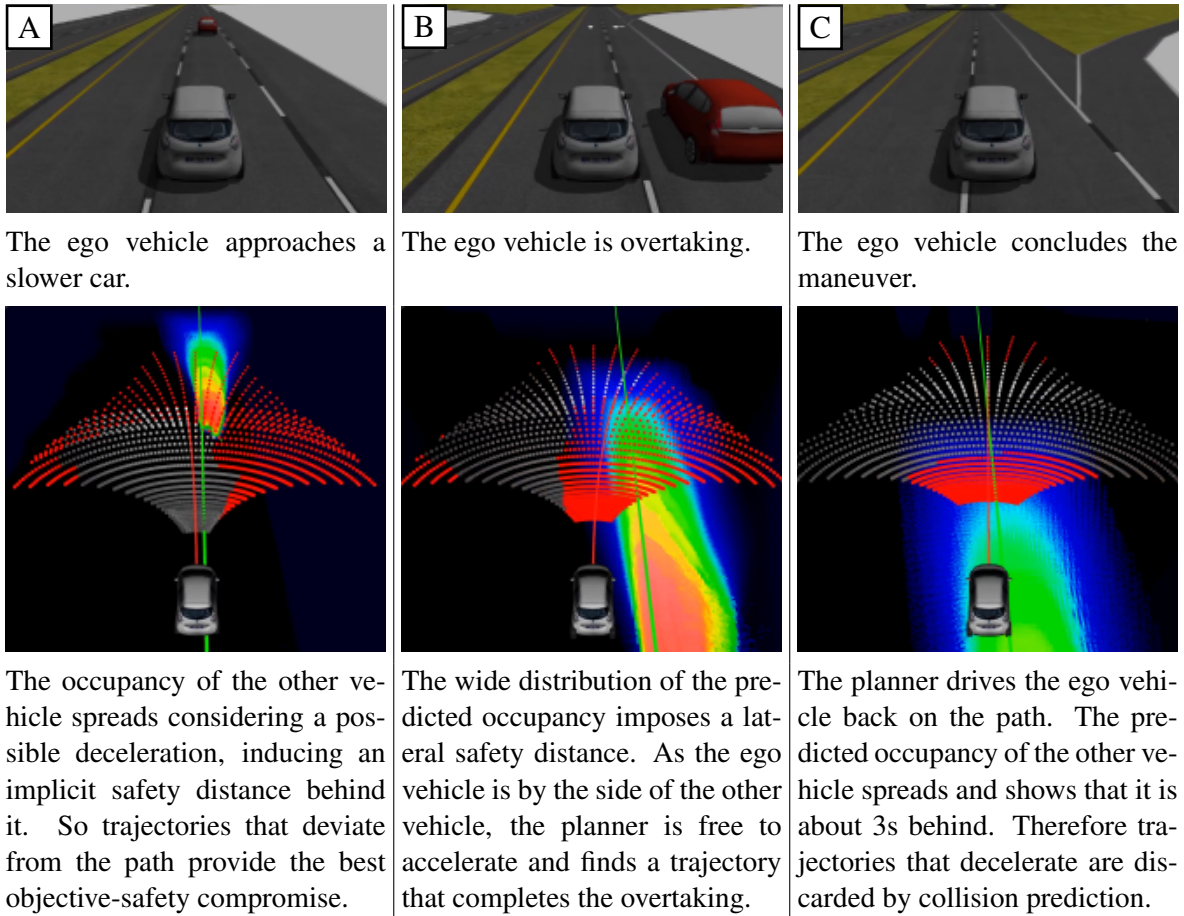


Figure 22 – Images of the top row illustrate a simulated overtaking scenario. Below are the predicted occupancy and the collision risk on sampled trajectories. The first graph shows the relative position of the ego vehicle with respect to the other during the maneuver. The second graph displays the ego-vehicle speed. The collision area is shown in red. The RSS risk area, in green, is estimated from definition in [SSSS17], parameters of (8), $V_{agent} = 20\text{km/h}$ and lateral acceleration of 0.4m/s^2 . It can be observed that the proposed particle-based occupancy prediction leads to the observance of the longitudinal and lateral RSS safety distances.

avoidance capabilities at low speed in diverse scenarios representing urban scenes with static obstacles, pedestrians and other vehicles. **Augmented Reality (AR)** on sensor data [GHRL22] has been used to safely test scenarios with pedestrians. Fig. 23 illustrates tests with static obstacles. Fig. 24 shows an avoidance in a scene with a pedestrian and static obstacles. This real world experiment used **AR** on sensor data to reproduce the scenario of fig.20, similar behaviors have been observed.

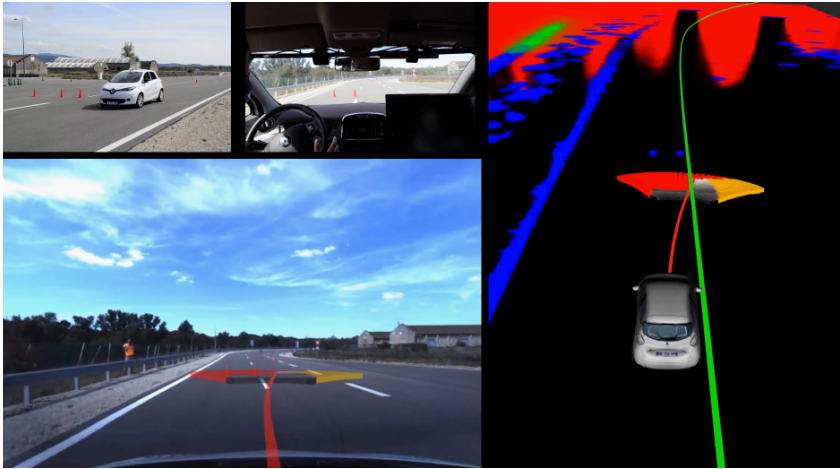


Figure 23 – Different points of view on the same scene of a real world test of the collision avoidance system on the *experimental automated car* with static obstacles. The occupancy grid and the planner outputs are displayed on the right.

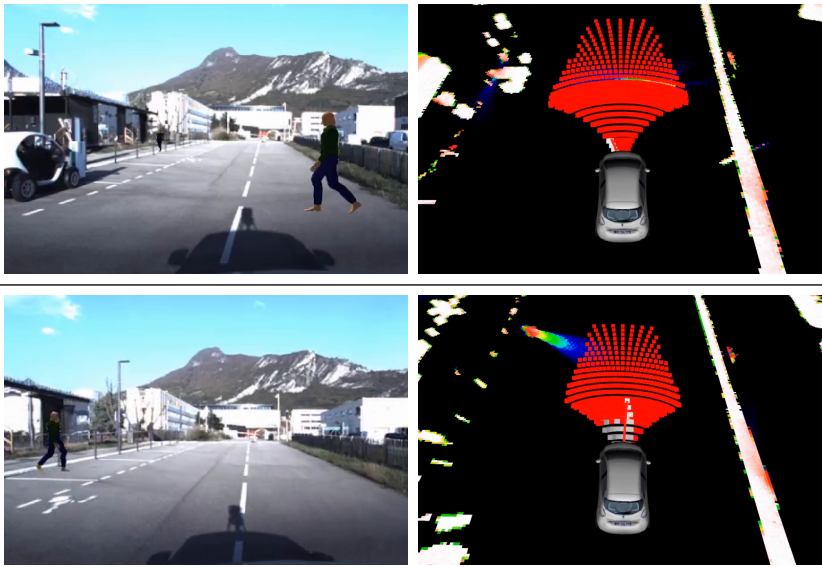


Figure 24 – Images from a test of the collision avoidance system with the *experimental automated car* using **AR** on LiDAR data to figure moving obstacles. On the left are images of the embedded camera. The corresponding predicted occupancy grids and the discretization of the trajectories are shown on the right. This test reproduced the behaviors that have been observed on fig.20.

3.5.4 Application to other vehicles

So the PCD collision avoidance framework has been successfully deployed on the [experimental automated car](#) within the project [IRT NANOEEC – SVA](#). Yet its versatility enables its use for other applications and with other vehicles. This same collision avoidance system has been deployed within the projects [IRT Nanoelec – Light Vehicle Demonstrator](#) and [R&D Booster MoovIT](#) respectively on an [urban light vehicle prototype](#) and an [industrial logistics robot](#). These projects are presented in section 6.2.3. The experimental platforms are described in section A. Realistic [HiL](#) simulations of these robots in their respective operating environment have been designed to evaluate the performance of the proposed collision avoidance. Real world tests with the [urban light vehicle prototype](#) have been executed.

Along with the robotic car, these experimental platforms present a complete manifold of challenges for collision avoidance. First, they present different environments. The automated car and the light urban vehicle have been tested in urban-like environments, either streets or shared space. These environments contain few static obstacles but may contain numerous [agents](#) that might move fast. The third vehicle is designed to operate indoor with less [agents](#) around but in cluttered environment. Thanks to a set of parameters, the collision avoidance system is adapted to each of these environments.

Also these robots have different sensing capabilities. While the car is equipped with a rich set of LiDAR sensors, the 2 other robots rely on a multimodal set of sensors, with cameras, LiDAR sensors and ultrasonic range sensors. The [CMCDOT](#) perception suite has been used in each case. Using the output of this fused perception enabled to naturally handle the different set of sensors regardless the modality. The probabilistic representation of obstacles is used to render complex sensing situations such as in the environment of project [R&D Booster MoovIT](#) that contains transparent obstacles that are invisible to LiDAR and cameras but can be sensed by ultrasonic range sensors.

These 3 robots also differ in their real-time constraints. The real-time constraint is shorter on the robotic car as the speeds of the ego vehicle and of the agents can be higher. Meanwhile, on the [urban light vehicle prototype](#) and on the [industrial logistics robot](#), the real-time constraint is relatively less challenging but the computing power is limited since these robots are only equipped with NVIDIA Jetson TX2 boards which were at the moment the lightest boards with GPU from NVIDIA. However, they sufficed to execute the perception suite and the collision avoidance within the real-time constraints. This demonstrates the frugality of the proposed avoidance solution.

Application to indoor robotics The proposed collision avoidance has also been adapted for an industrial logistics application for the project [R&D Booster MoovIT](#). The [industrial logistics robot](#) is designed to convey goods in a human-populated indoor environment. This robot differs from the [experimental automated car](#). It is equipped with a limited set of sensors, LiDARs and ultrasonic range sensors. Also its computing power is limited. [HiL](#) simulated experiments have been executed on its embedded computer, a NVIDIA Jetson TX2.

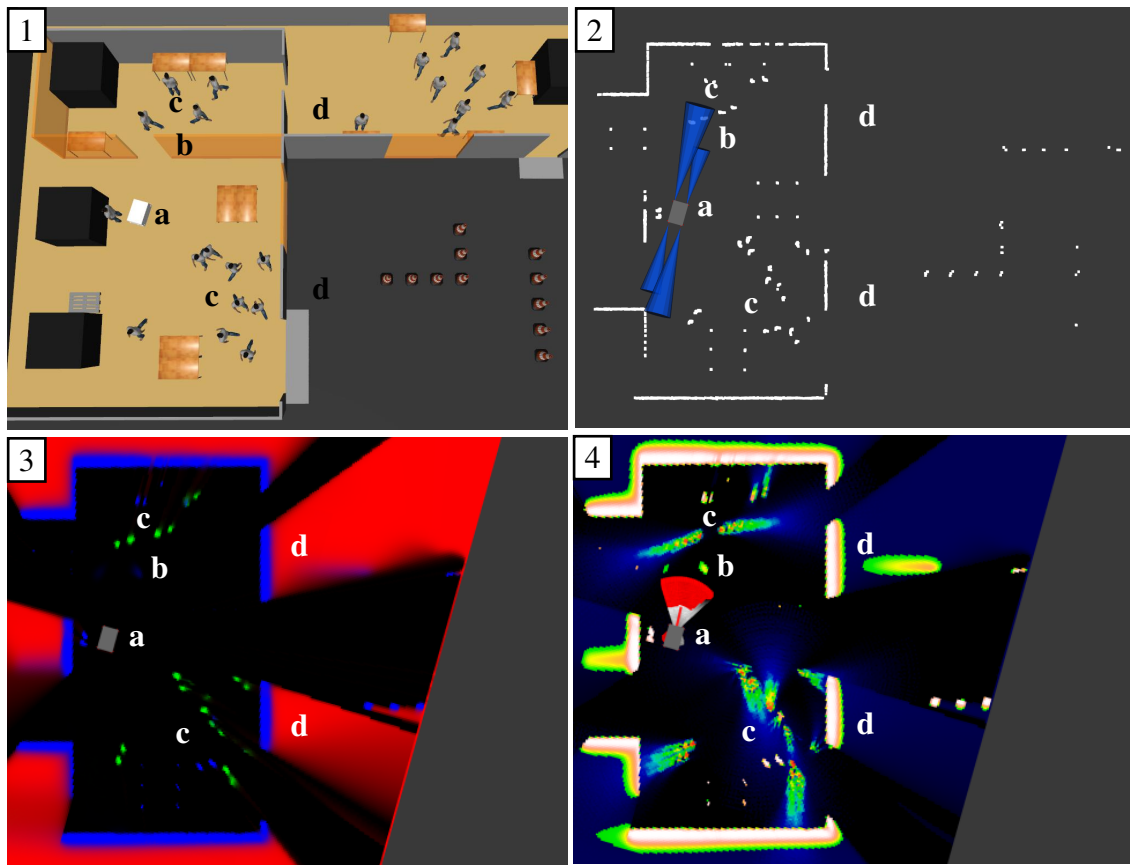


Figure 25 – The proposed method has been tested for indoor navigation of an *industrial logistics robot*. (1) displays the simulated environment with the robot at (a). (2) presents the outputs of its LiDAR and sonar sensors. (3) shows the corresponding dynamic occupancy grid. (4) shows the predicted occupancy grid and the current planning output. This grid showcases a global observation of the environment including, transparent walls that are visible only to sonars (b), pedestrian motion (c) and unobserved areas (d). The proposed collision avoidance method is able to consider all these elements to navigate safely in this environment.

Fig.25 illustrates these tests. The operational environment contains surfaces that are transparent to LiDAR and observable only to sonars. But the range of sonars is short. Moreover sonars only provide a single range observation over a large angular sector. With this robot, sensing the environment is more complex. Under these constraints and considering that only feet of humans are visible to LiDAR sensors, it would not be possible to generate an object-based perception from clustering and object tracking. So previously mentioned methods of the state of the art could not be applied. Yet, in the proposed approach, the *CMCDOT* perception suite enables the fusion of LiDARs and sonars, which is almost seamless for the *PCD* itself since the occupancy prediction processes the probabilistic occupancy at cell level so it represents faithfully the various obstacles of this scene and their motion. Then, with a map-based global planner and the proposed local collision avoidance, the robot has been able to navigate safely and complete several loops across the whole simulated environment. This demonstrates the versatility of the proposed method in a very different application. It also shows that the proposed solution is robust under the constraints of embedded hardware

and limited sensing capabilities, even in a complex environment. Such a result has not been observed in the studied related works.

Application to a light urban vehicle The proposed avoidance has not yet been tested in real world on the [industrial logistics robot](#). However it has been deployed on the [urban light vehicle prototype](#) that is equipped with the same computing hardware and a similar set of sensors. As the [CMCDOT](#) perception suite is used, the deployment of the [PCD](#) is similar to the above-mentioned applications. Even though this vehicle had a different and complex motion model, with a passive trailer, it has easily been taken in account in the proposed approach since the sampling-based [MPC](#) planner can handle this. Then global planner was used to compute a path to the goal while the proposed avoidance system was in charge of avoiding unexpected static and dynamic obstacles. This vehicle has been designed for navigation at very slow speed (below 10 km/h) in urban environments so tests mostly considered avoidance of static obstacles and pedestrians. For the tests of the local avoidance, obstacles were placed on purpose on the global path. Tests demonstrated the capacity of the vehicle to navigate safely in the considered environments. This different application demonstrates again the versatility of the proposed avoidance solution, this time with a real world application on a light robotic platform.

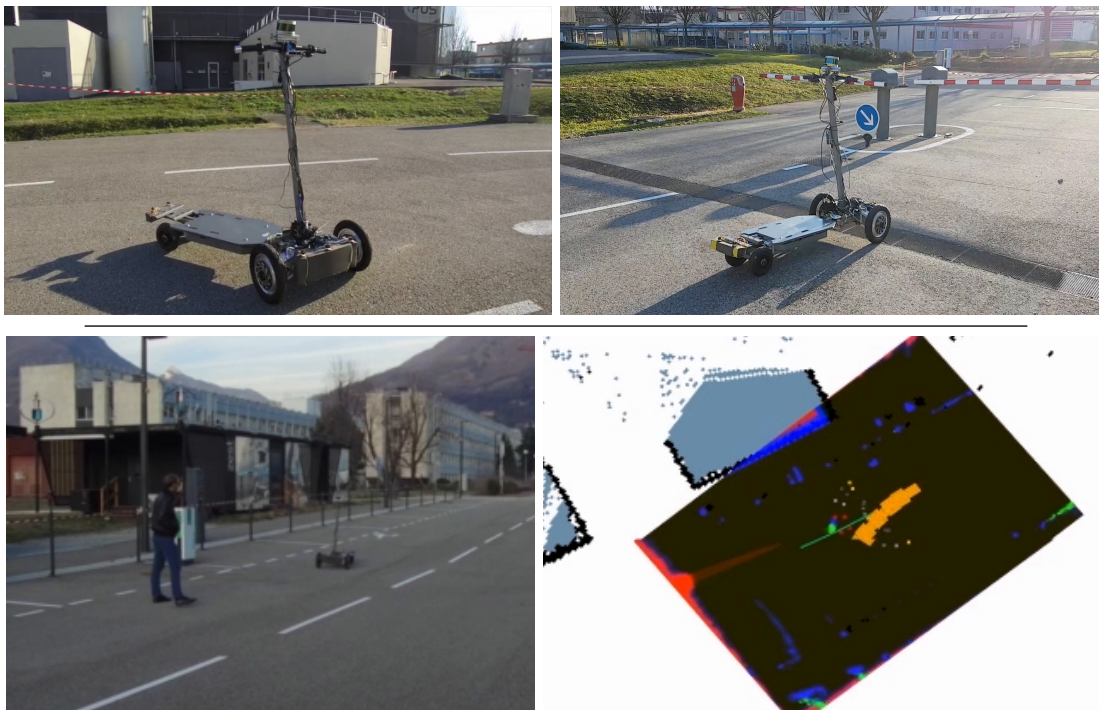


Figure 26 – These images illustrate the [urban light vehicle prototype](#) on which the proposed avoidance has been deployed. With the help of a map and a global planner, this vehicle successfully navigated in real world tests through simple urban scenarios that considered static obstacles and 1 pedestrian. This vehicle, with a multimodal set of sensors and a light computing power, strongly differs from the [experimental automated car](#). The application of the proposed avoidance to this different application demonstrates its versatility.

3.6 Chapter conclusion

In this chapter is proposed a collision avoidance system that has been tailored to operate from [Dynamic Occupancy Grid](#) based perception. It is designed for local avoidance under uncertainty in unstructured environments. The method relies on the [Predictive Collision Detector \(PCD\)](#) which is proposed as an interface between [Dynamic Occupancy Grid](#) based perception and sampling-based planners. While related works chose to compute the avoidance on object-based representations of the environment, the proposed approach with the [PCD](#) operates only on elementary spatial occupancy, at cell and particle level. It uses a probabilistic prediction of future occupancy with motion particles and the collision detection is computed on a grid representation. Parallelizing the computations in a [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#) implementation enabled the deployment on several robots, including an automated car, under real time constraints. This original approach leads to a different paradigm of collision detection and avoidance that has shown advanced driving behaviors such as anticipation, decision, observance of safety distances and navigation in a crowd. All of these behaviors arise from the particle-based approach of the [PCD](#) that provides a single stochastic risk estimation that is generic and can be applied in various situations. The proposed avoidance qualitatively compares with the performance of various state-of-art methods in their respective operating domains, proving that it is more versatile. The navigation safety as been proven in a quantitative evaluation according to the well-established standards of [Responsibility-Sensitive Safety \[SSSS17\] \(RSS\)](#). The robustness of the approach has also been demonstrated in complex scenarios and under limited sensing capabilities. As a broader use of state of art perception under uncertainty could improve safety and efficiency of [Automated Driving Systems \(ADS\)](#), the present thesis intends with this contribution to facilitate the use of [Dynamic Occupancy Grids](#) for collision avoidance. Moreover, as it has been observed that use of [Dynamic Occupancy Grids](#) is under-represented in local planning, this thesis also hopes to contribute to enhance the connectivity of the state-of-art of perception and navigation.

This contribution is an opportunity for several future works. First, the [PCD](#) could be a standard interface between grid-based perception systems and sampling-based planners. It has been designed for this purpose but had not been tested with several perception systems and different planners. This could be done to achieve a modular solution and compare the performance of different combinations of perception and planning systems. Also the occupancy prediction model could be improved. The kinodynamic unicycle motion model that is used to predict the motion of agents provides a conservative planning policy but a more accurate prediction model could result in a less conservative but more efficient planning. This has actually been considered within this thesis. An extension of the [PCD](#) framework has been proposed for more efficient human-aware navigation. It is presented in chapter 4.

Chapter 4. Application of the proposed method to human-aware collision avoidance

4.1 Chapter outline

While the previous chapter introduces a new collision avoidance framework whose originality is to operate directly on [Dynamic Occupancy Grids](#) without object-based representations of the environment, the present chapter introduces an extension of this collision avoidance system to consider cooperative navigation among humans. Section [4.2 Motivation for an extension of the PCD to consider its application to human-aware navigation](#) reminds that navigation of [Automated Driving Systems \(ADS\)](#) in shared spaces remains an open problem because current collision avoidance systems, including the previously proposed [Predictive Collision Detector \(PCD\)](#), are generally too conservative. This is the motivation for an extension of the [PCD](#) for interaction-aware collision avoidance with humans. Section [4.3 Related works in human-aware navigation and originality of the proposed solution with the extended PCD](#) reviews the limitations of state-of-art human-aware collision avoidance systems. Classical predictive approaches suffer from the [Freezing Robot Problem \[TK10\] \(FRP\)](#) while proactive planning approaches are restricted by their inherent complexity. Finally the proposed approach is presented, it relies on a particle-based interaction-aware occupancy prediction that derives from a state-of-art [Social Force Model \[MHG⁺09\] \(SFM\)](#) based pedestrian-vehicle interaction model. Section [4.4 Methodology of the extension of the Predictive Collision Detector \(PCD\)](#) describes how the [PCD](#) is extended to become aware of the probabilistic semantic classification of agents and to embed an interaction-aware prediction model of pedestrian motion within its particle-based occupancy prediction. Section [4.5 Deployment and evaluation of the extended PCD in the context of human-aware collision avoidance](#) describes the implementation of the novel human-aware collision avoidance system on an autonomous car prototype. Some qualitative results of occupancy prediction and navigation in dense crowds are presented. A quantitative analysis investigates experimental results in terms of navigation efficiency, safety and pedestrian comfort.

The contributions that are presented in this chapter are :

- the extension of the [Predictive Collision Detector \(PCD\)](#) that makes it aware of the semantic classification of agents
- the development of an interaction-aware occupancy prediction model specifically for pedestrians
- the qualitative and quantitative experimental validation of the method

These contributions have been published in :

Thomas Genevois, Anne Spalanzani, and Christian Laugier. Interaction-aware predictive collision detector for human-aware collision avoidance. In *IEEE Intelligent Vehicles Symposium (IV)*, 2023

A part of this work has also been presented in a invited talk at a workshop :
Thomas Genevois, Anne Spalanzani and Christian Laugier. Interaction-aware Motion Planning as an alternative to Proactive Planning for Human-aware Navigation. In *IV2023 Workshop on Interaction-driven Behavior Prediction and Planning for Autonomous Vehicles*.

4.2 Motivation for an extension of the PCD to consider its application to human-aware navigation

Mercedes-Benz Group, on their website, state that *"Automated vehicles will become part of everyday life in the near future – not only on clearly structured motorways and expressways, but also in urban areas. Yet urban traffic in particular poses huge challenges for the researchers and developers of the necessary technologies. The task is to master considerably more complex road layouts, processes and possible scenarios. In addition, towns and cities are places where many different road users interact in a relatively confined space."* [M23]. This observation is shared by other actors, observers and regulators of the autonomous driving systems community [W23, B19, C20, L23]. Even though [Automated Driving Systems \(ADS\)](#) have been deployed on public roads and are already used for transportation of goods and people, they are currently restricted to simple [Operational Design Domain \(ODD\)](#) where they evolve in structured environment and have limited interactions with only few other road users. But [ADS](#) should be progressively released in new [ODD](#) of increasing complexity [B19], such as city centers and shared spaces. In urban traffic, the road network is more complex and the density of road users is significantly higher. This issue has already been mentioned in section 1.1.

In order to face these incoming challenges, autonomous driving software should evolve to operate under a wide range of circumstances and be robust to unexpected events. For these reasons, in chapter 3, a novel collision avoidance framework has been proposed. This new method uses [Conditionnal Monte-Carlo Dense Occupancy Tracker \[RNL15\] \(CMCDOT\)](#) perception suite and the proposed [Predictive Collision Detector \(PCD\)](#) to enable the direct use of rich probabilistic perception for a versatile navigation system. As it is designed for operation in unstructured and potentially highly crowded environment, this [PCD](#) contributes to make [ADS](#) capable of navigation in urban traffic.

Among the many challenges that arise in urban environments, this chapter focuses on automotive navigation in human populated shared spaces. This scenario, that is often encountered in city centers, considers a car navigating at low speed in presence of many pedestrians without explicit intersections and priority rules. Various solutions [CCMZ18] have been proposed for such navigation in human crowds but they are often dedicated to restricted [ODD](#) (kinematic constraints, speed limitations, number and category of agents, ...) and only a few of them consider automotive applications. Navigation of a car in shared spaces remains an unsolved problem. It is particularly difficult because it presents a highly dynamic unstructured environment. Considering this, the previously proposed [PCD](#) appears as a likely candidate for this task. It has been already demonstrated in section 3.5.2 that this system is

capable of safe navigation in crowds of average density. Yet, the version of the **PCD** that has been proposed in chapter 3 does not comprehend the opportunities of cooperation navigation with humans. So it does not constitute an efficient solution for human-aware navigation. This is particularly visible in fig.27 with the example of the **experimental automated car** trying to navigate in a simulated very densely crowded shared space. The collision avoidance framework that has been proposed in chapter 3 is controlling the vehicle, with the **Predictive Collision Detector (PCD)**. But the conservative prediction approach does not consider that pedestrians are aware of the vehicle. So the predicted occupancy, which results from the motion of this crowd, covers the whole navigable space and the planner can not find any safe trajectory. The vehicle is stuck. This issue is not specific to the **PCD**. This is a common problem that most mobile robots encounter in dynamically crowded environments. It has been formalized and designated as **Freezing Robot Problem [TK10] (FRP)**. It occurs when *"the environment surpasses a certain level of complexity [such that] the planner decides that all forward paths are unsafe"*. In the previous example, the **PCD** is missing 2 things to be able to solve the Freezing Robot Problem. First it would need to be able to understand that the agents that it is facing are humans. Then it would require a human-vehicle interaction model.

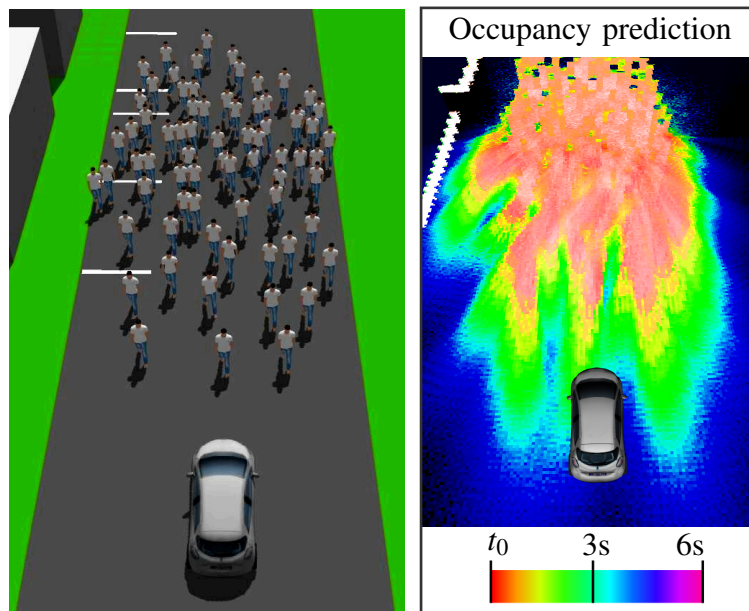


Figure 27 – Example of a scenario where the proposed **PCD** faces the **Freezing Robot Problem [TK10] (FRP)**. To try to navigate through the densely crowded simulated shared space, the **PCD** computes the predicted occupancy grid. However, as the prediction covers the whole space, the planner can not find any safe trajectory.

So, to address navigation in shared-spaces, it is first proposed to make the **PCD** aware of the semantic classification of the agents (pedestrian, cyclist, car, ...). For this, the extended **PCD** integrates a new classification input and combines several class specific prediction models. Such that, a less conservative and more accurate prediction can be adopted for each class of agent. Then, within this new classification-aware framework, the extended **PCD** is enriched with a prediction model for human motion. This model is inspired from expert knowl-

edge and predicts pedestrian-vehicle interactions. In this way, this new version of the PCD can anticipate cooperative navigation with pedestrians.

Therefore, in this chapter, this thesis proposes to extend the previous Predictive Collision Detector (PCD) for interaction-aware collision avoidance among humans. This is motivated by the intention to propose a method to efficiently address navigation in shared spaces as it appears needed for future development of Automated Driving Systems (ADS). As this approach relies on the initial PCD, it is also versatile, capable of navigation in complex environment under uncertainty and ready to be deployed with an available perception system. The addition of the interaction-aware model for pedestrian motion prediction significantly improves navigation in shared spaces. As it is less computationally demanding and more versatile than proactive planners but still able to benefit from cooperation with humans, the proposed approach offers an interesting compromise between classical planning approaches and human-aware proactive planners.

4.3 Related works in human-aware navigation and originality of the proposed solution with the extended PCD

4.3.1 Related works in human-aware navigation

The proposed extension of the PCD considers its application for human-aware navigation. Therefore this section reviews a selection of relevant research works in this field. Human-aware navigation, also called social navigation, is a wide research field in Robotics that covers decision making, global planning and local planning for robots in interaction with humans. The present thesis only addresses the task of local planning which is to ensure to reach a target at a local level while dynamically avoiding collisions. To perform human-aware collision avoidance, a robot has to predict the motion of nearby humans and plan its own motion accordingly. However humans are conscious agents that are likely to replan their motion depending on the actions of the robot. Therefore, for an efficient navigation, the robot needs to anticipate how its own actions will influence the other agents. This is why human-aware collision avoidance is a *joint problem of prediction and planning*. There are multiple ways to address or circumvent this problem. [CCMZ18] presents a survey of human-aware planners. [KA17] compares the performance of some such planners. Human-aware planners can be classified in different categories according to the types of interactions they can produce during navigation. There is not yet a general agreement of the research community on the designations and definitions of these categories of human-aware planners. The designations that are used in this thesis mostly refer to those that are used in [CCMZ18] and synthesize those that are used in the other research works that are cited in the present section.

Reactive planners The planners of this category only consider the current state of the environment for avoidance. Classical planners such as the Dynamic Window Approach [FBT97] or the Potential Fields method [K86] belong to this category. Without any prediction and

anticipation of the motion of [agents](#), these planners perform poorly in dynamic human-populated environment and are subject to the [FRP](#) even with low densities of agents but they provide a frugal approach and a simple behavior that can easily be understood by humans around the robot.

Kinodynamic predictive planners These planners estimate the future state of the environment to achieve a more accurate avoidance of moving obstacles. This prediction relies on conservation of the motion that has been perceived with eventual variations to model dynamic capabilities of the [agents](#). In this category, the common Velocity-Obstacle [[FS98](#)] approach is used in human-aware navigation, it is included as a fallback mode in [[TSFA21](#)]. The [PCD](#) that has been presented in chapter 3 falls in this category. These planners uncouple and consider sequentially the tasks of prediction planning. As a result, the planning computations remain rather simple and with limited complexity even in presence of numerous agents. The resulting behavior of the planner is also quite understandable by humans. However they do not capture the possible cooperation with humans so they encounter the [FRP](#) when in crowded environments.

Learning based planners This category refers to planners that rely on data-driven learning-based approaches instead of explicit models for prediction and planning. As the internal learnt interaction model is not accessible, these planners can not be directly categorized as the others can be. However it is possible to classify these planners among the other categories with an experimental analysis of the behavior of the planner. Various approaches have proposed, most of them rely on deep Reinforcement Learning, as in [[ECH21](#)] and in [[LCL⁺21](#)]. Learning-based planners are mainly limited by the availability of relevant and large enough data sets. Also they do not generalize as well as model based approaches for scenarios that differ from the training data [[CCMZ18](#)]. None of the previously mentioned learning-based approaches showed evidence of proactive navigation behavior.

Proactive planners This is the category of planners that address jointly prediction and planning. They estimate the future state of the environment and how human [agents](#) would react to the possible actions of the robot. So they are not subject to the [FRP](#) and they have the richest capabilities of interaction and provide the optimal solutions for efficient cooperative navigation of robots with humans. Yet, as proactive planning addresses simultaneously prediction, planning and their correlations, it conveys an high inherent complexity. This complexity is the main limitation of this type of planners. So they often require strong simplifications of the planning problem in order to operate in real-time. This complexity also limits the number of agents that can be considered at a time and it also prevents to consider additional real-world constraints, such as partial and uncertain perception, that would increase even more the complexity of the computations. The proactive planning approaches include [[KSM22](#)] which achieves proactive navigation of an autonomous car in a crowd but had to uncouple the longitudinal and steering planning. [[FS14](#)] presents a proactive planner

that plans local trajectories and uses an extended [Social Force Model \[MHG⁺09\]](#) (SFM) to estimate and minimize its impact on nearby humans. Although its complexity increases exponentially with the number of agents. [\[TSFA21\]](#) navigates proactively a robot in an indoor environment that can be densely populated however the cooperative planning is restricted to the 2 nearest humans only. Also [\[TSFA21\]](#) identifies the "entanglement issue" that arises when humans do not behave as expected in the prediction model of the planner. This issue is representative of a global side effect of using too rich planning approaches for human-aware navigation : too complex navigation behaviors can be difficult for nearby humans to understand and predict, resulting in missed cooperations. A simple and explainable behavior is sometimes needed. Several robotic applications actually require the robot to assume the responsibility of collision avoidance and let humans take the initiative of a cooperation. This is the case in automotive applications where it is generally considered that "*right-of-way is given, not taken*" [\[SSSS17\]](#). This is also a specific request that has been formulated by the industrial partner in the project [R&D Booster MoovIT](#).

Behavioral predictive planners Planners of this category predict the future state of the environment and then plan for the optimal trajectory. However, unlike classical predictive approaches, the prediction embeds behavioral human motion models that can be obtained either from expert knowledge or from data-driven learning. The interest of this approach is that it processes sequentially prediction and planning. Therefore the models and computations remain simple. Also the use of specific human motion models provides the planner with a broader understanding of human motion and enables richer interactions. While predictive planners do not suffice for cooperative navigation among humans and while the application of proactive planners are limited by their inherent complexity, the emerging category of behavioral predictive planners offers an interesting compromise. Among the behavioral predictive planners, [\[BAB⁺22\]](#) achieves navigation of multiple robots in an indoor human-shared environment thanks to a prediction of human motion which is based on a [SFM](#). [\[FBH⁺18\]](#) introduces a probabilistic approach with a variable confidence in a utility-based human motion prediction model and used it for quadcopter navigation in human vicinity. The 2 above mentioned works considered that human trajectories are agnostic of the presence of robots. This is a conservative assumption that simplifies computations and ensures that the robot will not drive too aggressively. It is effective for small robots interacting with few persons but would lead to the [FRP](#) in more complex situations. Instead, the prediction process may consider that humans are aware of the robot and will react to it. This approach defines the category of planners that will be designated as *interaction-aware planners* in this thesis. The ability to anticipate interactions with humans generates rich navigation behaviors among humans and solves the [FRP](#). An example is PORCA [\[LCB⁺18\]](#) that uses Partially Observable Markov Decision Process to encode pedestrian intentions and an improved model of ORCA as pedestrian motion model. It is demonstrated on a robot scooter that has been able to navigate in real-world in a dense crowd. However PORCA remains limited because it applies social navigation only on longitudinal planning and can not model non-holonomic constraints

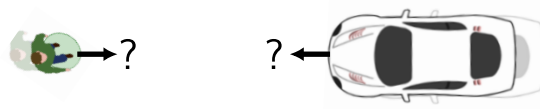
of vehicles.

In conclusion, it appears that the most promising approaches for human-aware navigation without facing the FRP are proactive planning and interaction-aware planning. Proactive planners generate the most efficient interactions and fluid navigation among pedestrians but they are limited by their complexity when it comes to handle large number of agents and real-world constraints such as uncertainty or complex motion models. On the other hand, interaction-aware planning relies on more simple prediction and planning models that can embed such real-world constraints and consider numerous agents but the interactions are not as good as with proactive approaches. The tables in fig.28 illustrate the different paradigms of predictive, interaction-aware and proactive planning and proposes a summary of their pros and cons. With the example of the frontal encounter with a pedestrians, it provides expected planning behaviors with these approaches. Based on these considerations, fig.29 proposes optimal domains of application of these approaches depending on the density of the crowd through which to navigate and the expected complexity of interactions. This analysis is based on the reviewed state of the art and the results of section 4.5.

The aim of the extension of the PCD to human-aware navigation is to address navigation in shared spaces where an autonomous vehicle may face dense crowds. Therefore, considering the already mentioned pros and cons of the different approaches, interaction-aware planning appears more relevant for the desired use case. Moreover, since the interaction-aware approach relies on a sequential process of prediction and then planning, it is compatible with the framework of the PCD (section 3.4.1) and only requires an evolution of the prediction module.

4.3.2 Proposed approach to extend the PCD for human-aware collision avoidance

So, the proposed approach is to extend the PCD for interaction-aware behavioral predictive planning with pedestrians, in potentially dense shared spaces while preserving its previous behavior for non-pedestrian agents and its ability to operate under uncertainty, in unstructured environments with simultaneous longitudinal and steering control of non-holonomic motion. It preserves most of the original framework. It relies on a similar perception system with the CMCDOT software suite. The new PCD also achieves a stochastic occupancy prediction. And the same sampling-based Model Predictive Control (MPC) planner is used to select the optimal trajectory and provide the control commands. However, the original PCD makes a prediction that is not aware of the type and of the behavior of agents. So it produces inaccurate, yet conservative, predictions that result in the planner being too cautious in avoidance maneuvers. It causes the FRP and inefficient navigation in crowded environments. In this, this previously proposed planner belongs to the category of kinodynamic predictive planners that has been defined in section 4.3.1.



Joint problem of prediction and planning

Planner types for Human-aware navigation

Predictive	Interaction aware	Proactive
<p>Brake and wait</p>	<p>Continue & expect cooperation</p>	<p>Take initiative & expect reaction</p>
Agents are not aware of me. I plan for myself.	Agents are aware of my status. I plan for myself.	I plan for agents and myself.
Uncoupled prediction & planning	Uncoupled prediction & planning	Coupled prediction & planning
Highly conservative	Suitable compromise	Very complex

Pros & cons of different types of HA-planners

	Predictive	Interaction aware	Proactive
Richness of interaction	Limited Only reacting	Average Anticipating cooperation	Best Taking initiatives
Performance in crowds	Poor Frozen robot	Best Good scalability	Average Limited scalability
Constraints integrability	Best Uncertainty, complexity, real-time	Best Uncertainty, complexity, real-time	Limited Rare in literature

Figure 28 – These tables illustrate the different paradigms of predictive, interaction-aware and proactive planning. With the example of the frontal encounter with a pedestrians, it provides expected planning behaviors with these approaches. The main features and pros and cons of these approaches are also summarized.

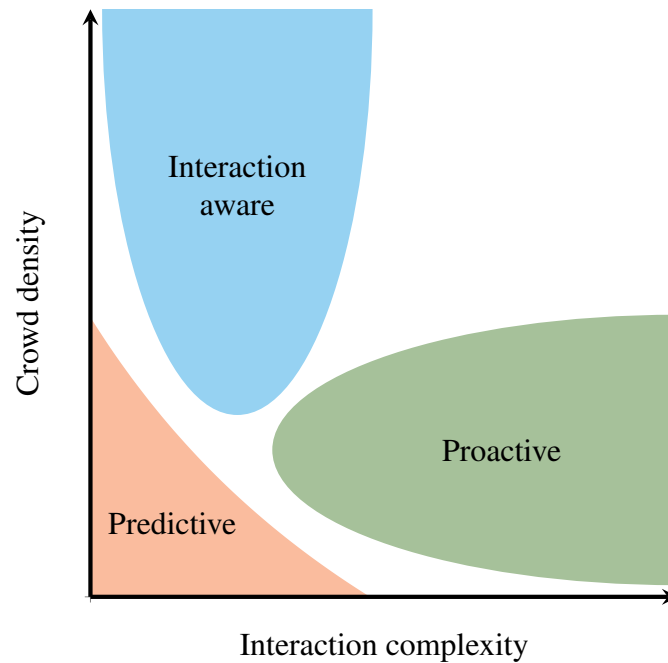


Figure 29 – This graph proposes optimal application domains for predictive, interaction-aware and proactive human-aware planners depending on the density of the crowd through which to navigate and the expected complexity of interactions. This proposition results from considerations that are presented in section 4.3 and in fig.28.

Opening the PCD for agents class specific prediction models To improve the **PCD**, this thesis proposes first to distinguish the prediction according to the semantic classification of **agents**. For this, the new framework relies on a deep learning based approach to extract semantic classification of agents and construct a semantic occupancy grid as in [EWL19] that identifies the probability of motion particles to refer to a car, a pedestrian, a cyclist or another type of agent. Then the **CMCDOT** perception suite is used not only to filter the occupancy probability and extract speed estimation but also to filter the classification probability jointly [RDGML22]. Such that it generates a Dynamic and Semantic Occupancy Grid. Then, within the **PCD**, instead of propagating the occupancy probability in time with only one global model, it is propagated according to the semantic classification. Class-specific prediction models redefine the prediction according to a continuous distribution of probabilities among classes. The previous kinodynamic model still holds for unclassified agents. So the main innovation of this new approach lies in the occupancy prediction module. While preserving the features of the original **PCD**, this extension enables to simultaneously model more accurately some classes of agents and plan more efficient collision avoidance. Since this approach handles a continuous classification probability, it is robust to uncertain classification and versatile enough to handle simultaneously agents of different classes.

Definition of a prediction model for pedestrian agents The proposed prediction model is inspired by a [SFM](#)-based pedestrian-vehicle interaction model that has been proposed for simulation in [[PMDS21](#)]. This model has been proven to represent accurately various actual human behaviors thanks to a comparison with real data. Since this original model requires the knowledge of pedestrians' intentions, it needs to be adapted for its use in prediction within the [PCD](#) that can only rely on observable data. The present thesis proposes to adapt this model for probabilistic occupancy prediction and to use it as a model for the class of pedestrian agents within a new version of the [PCD](#). This new prediction model embeds empirical knowledge of human behavior during motion and interaction with a vehicle. Thanks to this addition, the extended [PCD](#) can be regarded as a behavioral and interaction-aware predictive planner.

As the prediction is aware of humans and interactions with them, the improved collision avoidance system becomes capable of navigation in crowded shared spaces with smooth maneuvers around humans and does not suffer from the [FRP](#). Also, it relies on an explicit human motion prediction model. Unlike with learning based approaches, this model can be analyzed to prove that the beliefs of the planner are reasonable and produce a safe and human-friendly behavior. For example, the proposed model states that the robot should not expect humans to react to its actions and that it should comply with human intentions that are observable. Therefore the robot does not take the initiative in an interaction and also it respects human initiatives to force the right-of-way, yield or make a detour. Finally, as this is an extension of the previous [PCD](#), the new avoidance system is as well capable of navigation under uncertainty, in complex and unstructured environments, with a realistic ego-motion model.

4.4 Methodology of the extension of the Predictive Collision Detector (PCD)

4.4.1 Reminder of the Predictive Collision Detector framework

The new proposed collision avoidance system is an extension of the original [PCD](#) framework. As illustrated in [fig.30](#), its structure only differs from the previous [PCD](#) by the addition of the semantic classification as a new input and the modification of the prediction module. The rest of the software framework which is illustrated [fig.16](#) remains unchanged. So the new framework consists of the following modules :

- A *Bayesian Occupancy Filter* [[RDGML22](#)], within [CMCDOT](#) software suite, which fuses and filters sensor data. It produces a Dynamic Occupancy Grid i.e. a discrete 2D representation of the likelihood of space to be occupied together with estimated speed of the observed motions. In the extended version, the [Dynamic Occupancy Grid](#) also provides semantic classification as a grid-based probabilist representation of an [agent](#) lying in an defined area to belong to one class (pedestrian, cyclist, car, unknown, ...).
- A *Prediction module* that projects the current occupancy grid along the motion particles and estimates the evolution of the occupancy grid over the next seconds. The main contribution of this chapter is an evolution of this module, from the simple kin-

odynamic prediction to a class-specific and behavioral prediction.

- A **Collision Detector** that investigates the predicted occupancy grid and returns the probability of collision in each position that the planner queries. It remains unchanged from the previously proposed **PCD**.
- A **Sampling-based Planner**. Considering the collision probability in sampled positions, a Sampling-based Model Predictive Control planner [DCJC08] estimates the collision risk over trajectory candidates. For our application, a cost function helps to find the best trajectory, among the safe ones, to follow a predefined path. Our solution is theoretically compatible with other sampling-based planners. This module is the same as in the previous system.

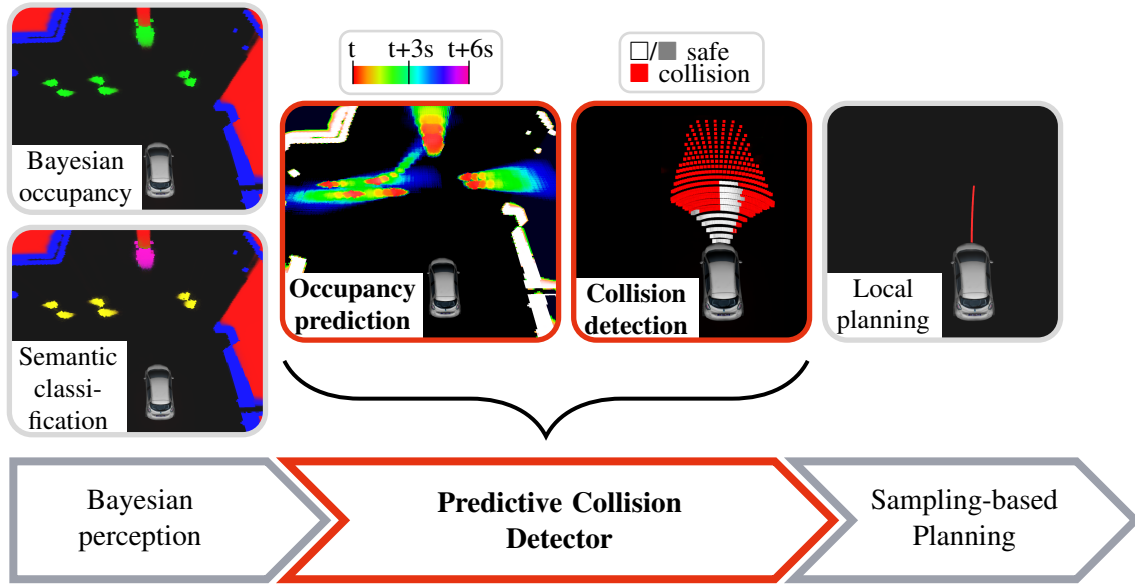


Figure 30 – Illustration of the major steps of the proposed framework for human-aware collision avoidance, from the perceived Semantic Dynamic Occupancy Grid until the planner selects a trajectory.

4.4.2 Opening the PCD for agents class specific prediction models

The first proposed improvement of the **PCD** is to make the occupancy prediction capable to apply different models for different classes of **agents**. The previous model is presented in section 3.4.2. Similarly with the previous system, each particle has a speed vector and carries a probability of occupancy that represents the probability of existence of a moving obstacle in the cell in which is the particle. But the perception also informs of the probability of the particle to refer to an agent of each class. These classes include but are not limited to pedestrians, cyclists, cars and unknown agents. As t_0 refers to the current time, let \mathcal{P} be a particle with a speed $V_{\mathcal{P}}(t_0)$ and an occupancy probability p . Let $X_{\mathcal{P}}(t_0)$ be its position. Then $P(C_k), k \in \llbracket 0, n \rrbracket$ represents the probability of \mathcal{P} to belong to an agent of class k among n classes. In the previous **PCD**, a prediction model is defined by \mathcal{U} , a set of actions that the agent could do, $P(u)$, the likelihood of each action $u \in \mathcal{U}$ and $X_{\mathcal{P}}(t) | u$ the future position

of the particle given that action u is realized. The new **PCD** proposes to extend this by considering a specific prediction model for each class of agent. So, for each class k , a prediction model is defined by its action set, \mathcal{U}_k , the likelihood of its actions, $P(u | C_k)$, $u \in \mathcal{U}_k$ and the resulting motion, $X_{\mathcal{P}}(t) | u \in \mathcal{U}_k$. Finally \mathcal{P} is split into a set of sub-particles, one for each action of each model. The sub-particles are then propagated according to the models and they carry an occupancy probability given by (10).

$$\forall k \in \llbracket 0, n \rrbracket, \forall u \in \mathcal{U}_k, P(u) = 1 - (1 - p)^{P(u|C_k) \times P(C_k)} \quad (10)$$

The total occupancy probability is preserved given that $\sum_{k \in \llbracket 0, n \rrbracket} P(C_k) = 1$ and $\sum_{u \in \mathcal{U}_k} P(u | C_k) = 1$. This distribution of the occupancy probability across multiple classes enables several prediction models to coexist and participate in the same prediction. It also makes the system robust to uncertain classification. Another element of robustness against uncertain classification is to consider a class for unidentified agents, C_0 . The previous kinodynamic prediction model is maintained for this class. This enforces the global and defensive approach to collision avoidance of the previous **PCD** as a default behavior for unidentified agents. Even though the division in many sub-particles increases the complexity of the computations, this approach leads to an efficient implementation in **Global Purpose computing on Graphics Processing Unit (GP-GPU)** since the computations are individually simple and parallelizable.

4.4.3 Definition of a prediction model for pedestrian agents

The second modification of this extension of the **PCD** is the addition of a dedicated prediction model for human motion. It is embedded in the functioning with multiple classes that is described in section 4.4.2. It redefines the occupancy prediction motion that is used for particles of agents that are classified as pedestrians. The proposed model is inspired from [PMDS21] which presents a model to simulate the motion of pedestrians in interaction with a vehicle. This work relies on the **Social Force Model [MHG⁺09] (SFM)** that models the motion intention and interactions with other agents as forces acting on a physical system. It also uses a state machine to represent decisions such as crossing before a car, stopping or stepping back. This model has been designed for simulation, with exact knowledge of the state and intentions of all agents, thus it requires adaptations before to be used as a prediction model. First, this thesis proposes to ignore interactions between the pedestrian and agents and obstacles other than the ego-vehicle. This simplifies the model by considering only the interaction between the pedestrian and the ego-vehicle. Second, as the decisions of pedestrians are not known to the prediction system, the set of states of the state machine is reduced to consider only states that can be inferred from observations. Namely, the adapted model uses the states "stopping"(S), "walking"(W), "running"(R) plus a new state, denoted as "unidentified motion pattern"(U), which covers both the case of insufficient observations to identify a state and the case of unidentified state. Then it is assumed that the observed current speed results from a **Gaussian Mixture Model (GMM)** which is composed by the speed distributions of the 4 states. Such that the state distribution for each particle is estimated by (11).

The parameters of the **GMM** have been found in surveys on human walking and running speeds [PDS19, PMDS21, BW17, HYE07]. In the absence of perception of the intentions of humans, this model is used to infer their state within the state machine of the model. This estimation is stochastic and continuous. A graph of the state distribution as a function of the estimated speed is presented in fig.31. These states refer to those of the internal state machine of the prediction model of human motion. They define the parameters of the **SFM** that is used for motion prediction.

$$\forall s \in \{S, W, R, U\}, P(s | \|V_{\mathcal{D}}(t_0)\|) = \frac{\mathcal{N}(\|V_{\mathcal{D}}(t_0)\| | \{\mu_s, \sigma_s\})}{\sum_{x \in \{S, W, R, U\}} \mathcal{N}(\|V_{\mathcal{D}}(t_0)\| | \{\mu_x, \sigma_x\})} \quad (11)$$

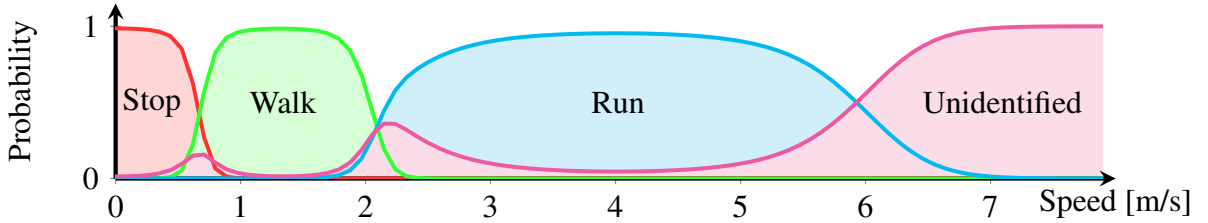


Figure 31 – This graph presents the estimated probability of the state of a pedestrian as a function of its estimated speed, according to the proposed **GMM**. The states refer to those of the internal state machine of the model of human motion. In the absence of direct observation of the intention of pedestrian, this approach provides a stochastic and continuous estimation.

The different states define the value of the parameters of the prediction model. For states S , W and R , these parameters are the same as for the simulation model in [PMDS21]. According to the Social Force approach, the human is modeled as a physical body reacting to 2 forces, its internal motivation and the interaction with the vehicle. [MHG⁺09] defines the model (12).

$$\begin{aligned} E &= X_{\text{vehicle}} - X_{\mathcal{D}}, \quad D = \lambda(V_{\mathcal{D}} - V_{\text{vehicle}}) + \frac{E}{\|E\|} \\ \theta &= \langle E, D \rangle, \quad T = \frac{D}{\|D\|}, \quad N = \perp T, \quad B = \gamma\|D\| \\ F_{\text{motivation}} &= \frac{V_{\text{desired}} - V_{\mathcal{D}}}{\tau} \\ F_{\text{interaction}} &= -Ae^{-\frac{\|E\|}{B}} (e^{-(n'B\theta)^2} T + e^{-(nB\theta)^2} N) + \varepsilon \\ \frac{dV_{\mathcal{D}}}{dt} &= F_{\text{motivation}} + F_{\text{interaction}} \end{aligned} \quad (12)$$

As in [PMDS21], A and $\|V_d\|$ are parameters that depend on the current state of the state machine. Similarly $F_{\text{interaction}}$ is assumed to be null in states S and R . $X_{\mathcal{D}}(t_0)$, $V_{\mathcal{D}}(t_0)$ are known from the occupancy grid and X_{vehicle} , V_{vehicle} from the odometry. The other parameters for pedestrian-vehicle interaction are defined in [PMDS21].

To consider the uncertainties of human motion that are not observable to this prediction model, this thesis proposes to add ε , the interaction model error which is assumed to follow

a Gaussian distribution. The intended pedestrian speed which also is not observable, V_{desired} , is assumed to follow a Gaussian distribution which is centered on a speed vector whose norm depends on the current state and whose direction is the one of the current observed motion. Within this context, ε and V_{desired} represents the unobservable intention of the pedestrian and his/her willingness to interact. The distribution of these 2 variables forms the action set \mathcal{U} of this prediction model. $P(U)$ is given as well by the Gaussian distribution of each of these 2 variables. So \mathcal{P} can be divided in sub-particles with different values of ε and V_{desired} . And then the motion model, $X_{\mathcal{P}} | u$, is obtained by integration of $\frac{dV_{\mathcal{P}}}{dt} | u$ (with a speed bound to prevent unrealistic speed estimations in critical avoidances) and $V_{\mathcal{P}} | u$. In this way, the interaction-aware pedestrian motion prediction model is completely defined. At last, in state U , the prediction follows the previously defined global kinodynamic model. This fallback mode ensures a defensive prediction when no motion pattern is recognized.

Fig. 32 and 33 show simulated and real-world experimental examples of predictions grids that have been generated with this model. Fig. 32 considers a simulated scene where the ego vehicles is moving at about 10km/h, 3 groups of pedestrians are walking at different positions around the vehicle. This figure comparatively displays the predicted occupancy that were computed with the kinodynamic unicycle model and with the behavioral interaction-aware model. Both approach produce similar predictions for a pedestrian who is on the side of the car, not interacting with it anymore. Predictions differ slightly regarding a group of pedestrians who are currently moving away from the car's trajectories. Finally predictions are very different for the last group of people who is further away but walking towards the car. It can be observed that SFM model expects them to cooperate and deviate from their current path but it is still unclear in which direction the person on the right will go. This illustrates a fragment of the diversity of behaviors that can be modeled by the proposed behavioral interaction-aware prediction.

Fig.33 presents a results that has been computed in real time on the on-board computer of a car with the interaction-aware prediction model in a real world scene. In this scene, the vehicle is driven at very low speed. The predicted occupancy grid shows the persons around the car and the static obstacles. The persons on the right are currently interacting with the car. This demonstrates the applicability of the proposed prediction method with real world data.

As the proposed model is similar to the expert one [PMDS21] that has been validated against real-world data, a realistic prediction can be expected. Also, the defines explicitly the beliefs of the planner regarding human motion, so it can be verified that they are reasonable and will produce a human-friendly behavior. For example, as $F_{\text{interaction}}$ is null for stopping and running humans (states S and R), the SFM model states that a pedestrian in this state is not going to react to the vehicle. Therefore the planner is not expecting cooperation from these persons. The robot will carefully avoid them. The interaction is considered only with pedestrians at walking speed. So the planner might expect cooperation only from walking people. They may contribute to the avoidance or in some cases let the vehicle pass. Here too the consideration of an explicit desired walking speed which is associated to a bound on the predicted speed of humans makes sure that the predicted motion is reasonable for a

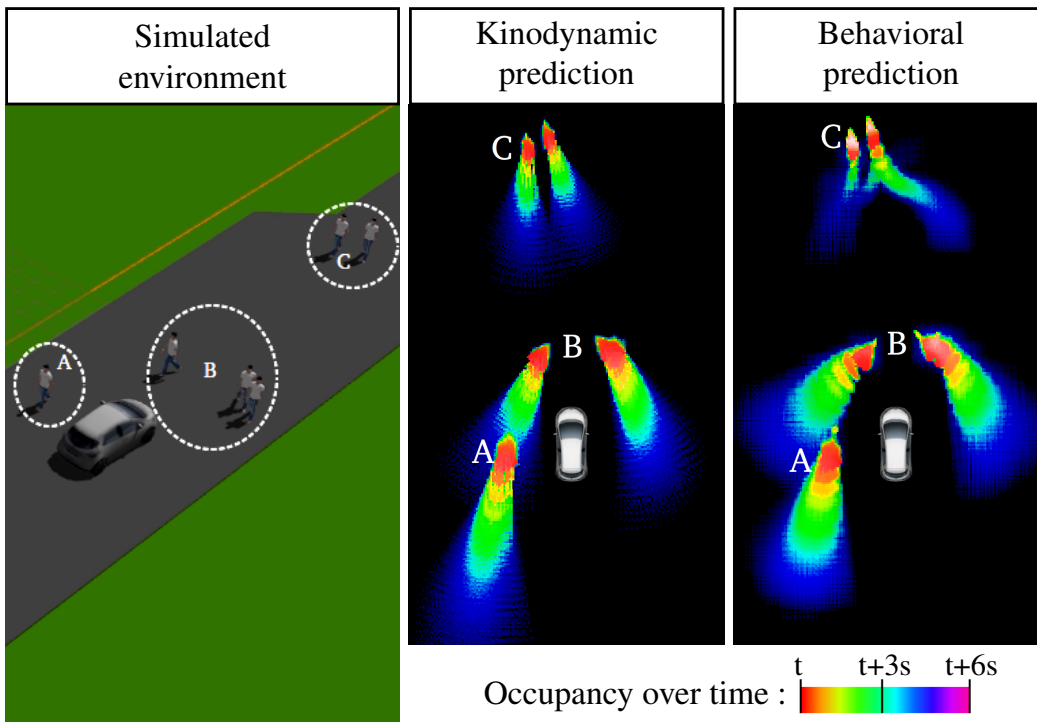


Figure 32 – In a simulated scene, this figure displays a comparison of the predictions with the kinodynamic unicycle model and with the behavioral interaction-aware model. For example, the behavioral interaction-aware prediction distinguishes a person who is not interacting with the car anymore (A), people that are currently avoiding the car’s trajectories (B) and people that are still far away from the vehicle but likely to cooperate and to deviate from their current path (C).

pedestrian. Furthermore the proposed model specifies that humans do not react to the actions of the vehicle but only to its current position and speed (X_{vehicle} and V_{vehicle} in the model). This ensures that the robot does not take the initiative in an interaction but let humans decide first. This is a conservative behavior that is desired for the use cases of the [PCD](#).

4.5 Deployment and evaluation of the extended PCD in the context of human-aware collision avoidance

4.5.1 Deployment of the extended PCD on a real vehicle and transferability to real-world applications

Test conditions The proposed extension of the [PCD](#) intends to enable efficient navigation of autonomous vehicles in urban shared spaces. To validate this with experimental results, the extended [PCD](#) has been deployed on the [experimental automated car](#) and on the [industrial logistics robot](#). Yet most validation tests have been conducted on the [experimental automated car](#) which at the same time more challenging and more representative of the intended use case. The test environments that have been considered represent urban environments with few static obstacles, crowds of different densities, mostly on 2 lanes roads. As for the tests of the previous [PCD](#) in section 3.5, the planner is given a path to follow approximately but

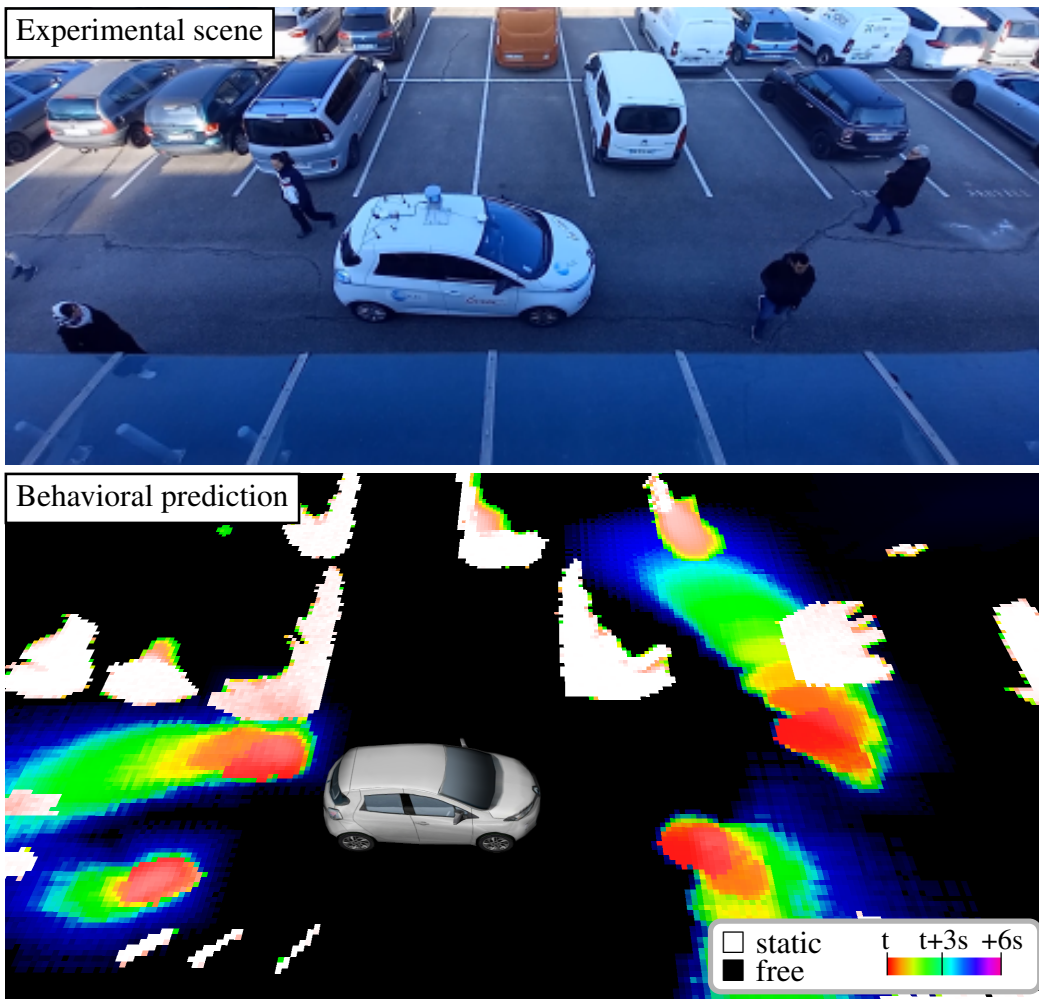


Figure 33 – The top image shows the experimental vehicle, driving at low speed near several persons. The second image displays the predicted occupancy grid that has been computed in real time on the on-board computer with the interaction-aware prediction model in this scene.

the constraint on the path following is much weaker than the collision avoidance. The planner is given a speed limit of 30km/h but it can freely adapt the longitudinal speed. The path following itself is not evaluated, the focus is made on the safety and efficiency of the collision avoidance.

Implementation and real-time execution The proposed human-aware collision avoidance system, with the extended PCD, has been deployed for real-world applications on the [experimental automated car](#). From LiDAR and camera observations, the [CMCDOT](#) software suite provides a fused and filtered perception as a Semantic Dynamic Occupancy Grid [[RDGML22](#)]. This grid is used for occupancy prediction and collision detection within the proposed PCD. Finally a sampling-based [MPC](#) planner [[DCJC08](#)] is used to select the optimal safe trajectory according to the risk evaluation that the PCD provides and the objective of path following. As this extended version of the PCD is also optimized in [NVIDIA Compute Unified Device Architecture \[L08\]](#) (CUDA) for execution on [GP-GPU](#), it is executed in

a maximum of 15 ms for a densely crowded area, on a NVIDIA GeForce GTX 1080. This allows to meet the real-time constraints and the whole collision avoidance runs at 15 Hz.

Testing the prediction in real world The proposed avoidance systems relies only on perception data, without prior knowledge of the scene. So, just as the previous version of the PCD, the CMCDOT perception suite provided a sufficient perception for real world deployment of this enhanced collision avoidance system an its real-time execution on-board of the [experimental automated car](#). However it has not been technically possible to guarantee experimental safety during autonomous navigation of the [experimental automated car](#) near humans. For this reason, this thesis presents only prediction results with real pedestrians, as in fig. 33.

Testing collision avoidance in simulation Then the proposed avoidance system has been extensively tested in [Software-in-the-Loop \(SiL\)](#) simulation of urban scenarios and navigation in densely crowded shared spaces. In this environment, the virtual ego-vehicle is a twin of the experimental vehicle that provides similar sensor data, responds to the same command messages, has similar motion model, shape and real-time constraints. Therefore, the tests in simulation represent quite well what could happen in the real-world. As in the validation of the previous PCD, the pedestrian simulator [Simulation of Pedestrians and an Autonomous Car in Shared Spaces \[P21\] \(SPACISS\)](#) has been used for simulation of crowds of pedestrians. It aims to reproduce realistic behavior of humans in their social interactions among themselves and with the car. Randomized initialization and decisions of simulated pedestrians generate a large diversity of behaviors to challenge the collision avoidance. These simulated pedestrians are mostly cooperative with the car but they also happen to be selfish. Extensive tests in simulation demonstrate the ability of the proposed navigation system to drive safely in crowds of such simulated pedestrians even in presence of non cooperative pedestrians.

Testing collision avoidance on a real vehicle with Augmented Reality With the help of [Augmented Reality \(AR\)](#) on the data of the on-board sensors, this navigation system has also been tested in real-world, on the experimental vehicle, on-board and in real-time, but for safety concerns with simulated pedestrians. Fig.34 and its linked video show the result of a test of real-world navigation of the [experimental automated car](#) with the proposed avoidance method in a dense crowd in [AR](#). The video first displays the augmented point cloud which is the input of the proposed collision avoidance system. Then in displays successively the Dynamic Occupancy grid, the predicted occupancy and the collision risk on the trajectory sampling. All of these have been computed in real time on board of the vehicle and the proposed collision avoidance system drove the vehicle accordingly to these outputs. If not the safety concerns, this experiment could have been realized with real people. This deployment and the whole validation demonstrates that the proposed avoidance system is able to navigate a real car in dense crowds, in real time, with simultaneous lateral and longitudinal control, with a non-holonomic motion model and latency in the command, without prior knowledge

of the scene and only using real sensor data. While none of the above-mentioned related works in automotive human-aware navigation considered these real-world constraints and such a deployment on a real vehicle, it appears that the collision avoidance system that this thesis proposes could be transferred to real world applications.

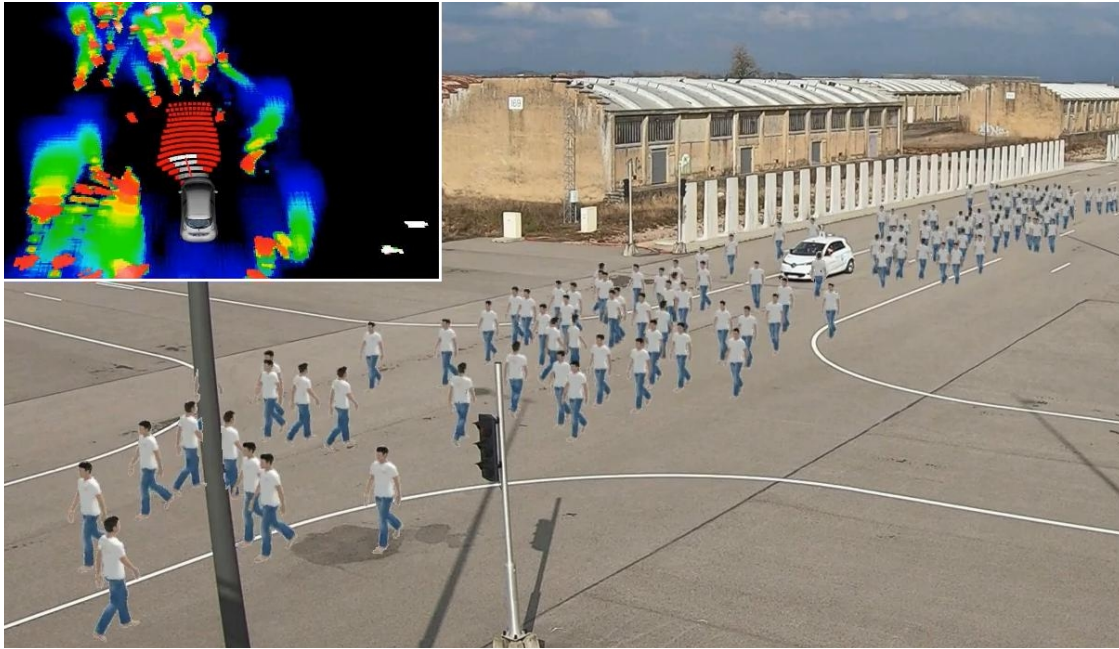


Figure 34 – Experimental navigation of a car in a crowd with the proposed human-aware collision avoidance system. The system is deployed on an actual vehicle but, for experimental safety matters, it operates with virtual pedestrians that are generated in Augmented Reality on LiDAR sensors [GHRL22]. The top left image displays the predicted occupancy grid and the trajectory candidates that were generated during this test. A video of this experiment is available at <https://inria.hal.science/hal-04073269>

4.5.2 Qualitative evaluation of the extended PCD

The proposed behavioral interaction-aware prediction model that is used in the extended PCD relies on an expert model that has been developed and validated with real human motion data in [PMDS21]. So this thesis does not consider to evaluate the accuracy of this motion prediction since it has already been validated. Instead this thesis aims to assess the benefits of embedding this prediction model in the PCD collision avoidance framework. In this purpose, multiple SiL simulated tests are executed under realistic constraints such as perception only from sensor available data, uncertainty and partial knowledge of the scene, non-holonomic motion, actual shapes of ego-vehicle and obstacles, no restriction on the number and types of agents and obstacles. None of the proactive and interaction-aware predictive planners that have been presented in section 4.3.1 operates under these constraints. Therefore the results of the proposed method can not be directly compared with these approaches. However qualitative results can demonstrate that embedding an interaction-aware prediction in the PCD has granted the planner with some typical features of proactive planners.

Ability to perform a cooperative avoidance As it is mentioned in [CCMZ18], the ability to anticipate and negotiate simultaneous avoidance with agents is a feature which is commonly associated to proactive planners. The same survey also observes that predictive planners are not able to solve simple human-aware navigation tasks such as mutual avoidance in narrow corridors because the planning for the ego is independent of the navigation of the other agents.

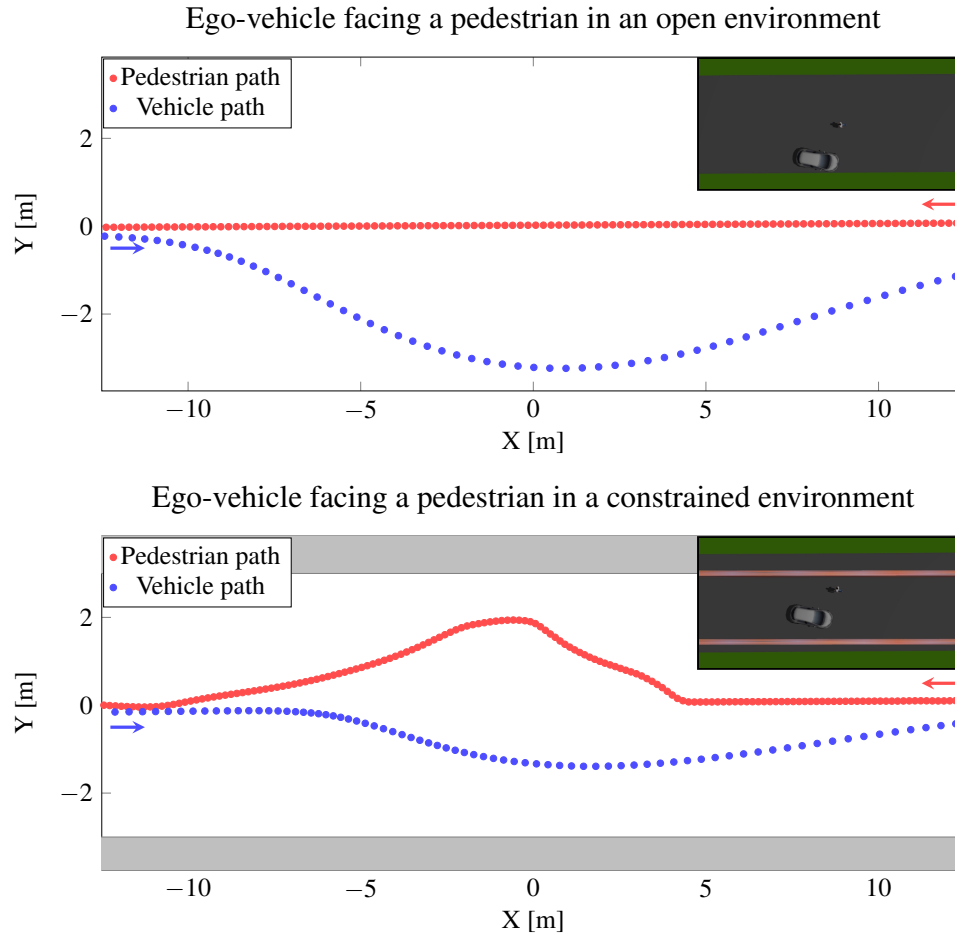


Figure 35 – These graphs presents the successive positions of the ego-vehicle and a pedestrian in a mutual avoidance situation. In the first simulated test, there is no restriction on the lane width. In the second test, the lane width is restricted to 5m. This exemple demonstrate the ability of the proposed planner to mitigate its impact on human agents in function of the scenario. Both tests are illustrated by bird-eye-view images, the video of the tests is included in <https://inria.hal.science/hal-04073269>

Fig.35 displays the paths of the ego-vehicle and a pedestrian in a mutual avoidance situation when they are face to face on the same path. The ego-vehicle is driven by the proposed planner with the behavioral PCD. A first test shows that, when facing the pedestrian in an wide open area, the proposed collision avoidance chooses to drives far from him/her. In this way, the ego-vehicle takes all the burden of the avoidance and causes reduced disturbance to the pedestrian. This is a desirable behavior. Moreover, when this encounter occurs in a narrow space, the proposed avoidance system with the extended PCD cooperates with the human

to negotiate a simultaneous avoidance in which both of them take a part of the burden of the avoidance. The avoidance is not symmetric because the car and the human do not have the same kinematics. This example demonstrates the ability of the proposed planner to mitigate its impact on human agents in function of the scenario. However as the proposed planner is not proactive, it is not planning the actions of the agent. Therefore it waits for the human to initiate the cooperation by showing a clear intention. This results in a delayed and less smooth interaction than what a proactive planner could achieve. Yet this behavior exemplifies the superiority of the proposed approach as compared to regular predictive planners.

Ability to handle simultaneously different classes of agents Fig. 36 provides an example where the proposed collision avoidance system deals simultaneously with several classes of agents. In the example, the ego-vehicle follows a preceding car and is approaching a group of people about to cross the road. The proposed occupancy prediction models the motion of people with the human-aware prediction model and the car with the default kinodynamic model. Both models contribute to the same predicted occupancy grid that is then used for collision detection and planning. Thanks to the extended prediction module that is aware of the semantic classification of agents, the planner still considers the specificity of each agent and predicts human motion with the appropriate model. This occurs without any ODD transition. For this, in the presence of humans, the proposed extended PCD-based avoidance solution is more effective than global purpose collision avoidance and it is also more versatile than most human-aware navigation systems that do not consider other agents than humans.

4.5.3 Quantitative evaluation of the extended PCD : efficiency, safety and pedestrian comfort

Efficiency A first set of tests aims to demonstrate that the extension of the PCD, with its new behavioral interaction-aware prediction model, improves the navigation efficiency of the whole collision avoidance system. So these tests consider the experimental automated car in a SiL simulation. Driving on an 8m wide road, the vehicle has to safely cross a crowd of pedestrians that are simulated by SPACISS. The tests consider different densities of crowd, up to 67 people/m² which can be regarded as an extremely dense crowd for the navigation of a car. These crowds follow an unorganized flow of people simultaneously in 2 ways, in the same direction as the car and in the opposite direction. Each test is repeated 10 times with a random initialization of the crowd. The test compares the navigation efficiency of the proposed avoidance solution with the extended PCD with the navigation efficiency of the original PCD that has been proposed in chapter 3. The extended PCD relies on the behavioral interaction-aware prediction model while the original PCD relies on the kinodynamic prediction model. However they use the same perception and planning systems. So a direct quantitative comparison of their navigation behaviors is possible and reflects the impact of the different prediction models.

As mentioned in section 4.5.2, none of the proactive and interaction-aware predictive

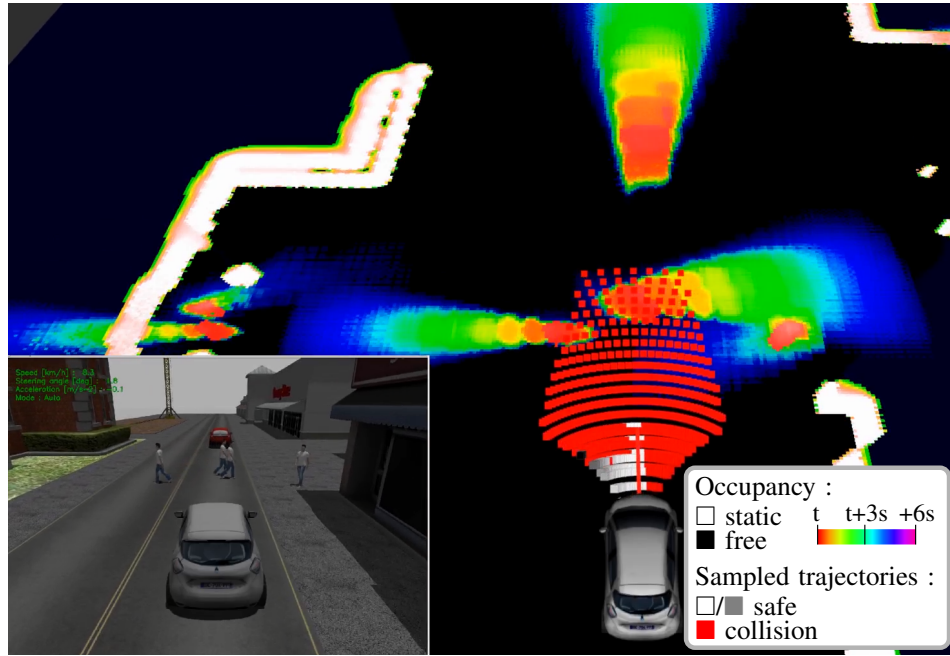


Figure 36 – This image presents an autonomous navigation simulation with different types of agents, a car and several pedestrians. A view of the scene is in the bottom left corner. The predicted occupancy grid is displayed. Each colored square illustrates a sampled trajectory, it is located at the 3s future position of the center of rear axle of the ego-vehicle if the commands of acceleration and steering that correspond to this trajectory are applied. These squares are red when the PCD identifies a collision risk on this trajectory. In such a situation, the extended PCD represents the motion of cars with kinodynamic prediction while the behavior of pedestrians is modeled with behavioral interaction-aware prediction.

planners that have been presented in the related works operate under the constraints of densely crowded environment, with limited sensing and realistic ego-motion model that are used in this SiL simulation. For this, a quantitative comparison with methods from the state of the art is not possible. As per metric, the navigation duration, Δt_{nav} , is used to evaluate the navigation efficiency. As in section 3.5.3, it is defined as the amount of time that the vehicle needs to complete the given path. It is assumed that this metric captures the fluidity of the different avoidances that the vehicles must do to cross the crowd. As it is mentioned in [CCMZ18] and used in [FBH⁺18, BAB⁺22, LCB⁺18, TSFA21, KSM22], such a navigation duration based metric is common for evaluation of the navigation efficiency of human-aware navigation systems.

Fig.37 plots comparatively the results of these tests. First, it can be observed that globally, in presence of humans, the system with the extended PCD completes the path faster than the one with original PCD. The performance of both systems is comparable in low crowd density but the improvement is significant in dense crowds. And even more, while the vehicle with the original kinodynamic PCD faces the FRP and is not able to cross the densest crowds, the vehicle that is controlled by the interaction-aware PCD is still able to navigate through these same crowds. These results prove that, in presence of humans, the integration of the

interaction-aware prediction model in the **PCD** collision avoidance framework leads to a significant improvement of the navigation efficiency. These results also reveal that, as proactive planners, the planner with the extended **PCD** is not subject to the **FRP**.

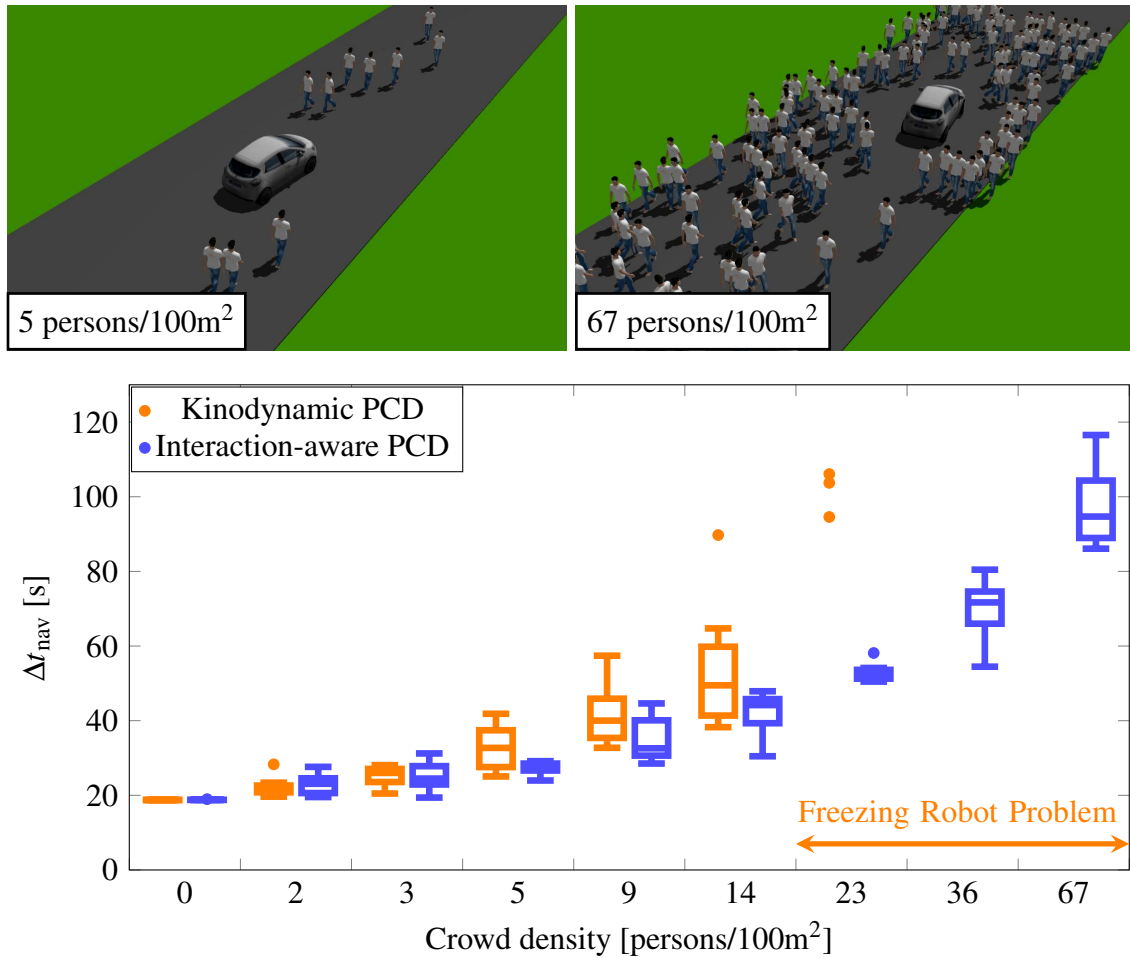


Figure 37 – 180 simulations are run in a randomly initialized scenario where the car has to cross a 25m long and 8m wide shared space with an unorganized longitudinal 2 ways flow of people. The top images illustrate this scenario. The graph presents, Δt_{nav} , the required time to complete the path, depending on crowd density. Each boxplot displays the median, quartiles, extrema and outliers over 10 simulations. The absence of some boxplots traduces the fact that the car was not able to cross in a reasonable amount of time. Results show that the planner with the extended **PCD** performs better than original **PCD** when the density of the crowd increases. Moreover this interaction-aware planner is not subject to the **FRP** even in extremely dense crowds.

Safety The first part of this validation shows that planning with the extended **PCD** is more efficient in crowded environment, although it is crucial to verify that it is still safe. The original **PCD** operates with a kinodynamic prediction model that always considers worst cases of acceleration, braking and steering. So it is highly conservative and overly conservative if it is compared to human drivers. Therefore it does not make sense to compare the safety of navigation with the extended and with the original **PCD**. Instead this thesis proposes to verify that the collision avoidance with the extended **PCD** is safe whatever the behavior of

humans, even if they are not cooperative. For this purpose, navigation in partly cooperative crowds has been identified as a particularly challenging task. If the majority of pedestrians in a crowd do cooperate, the autonomous vehicle can drive rather fast. Then the presence of few non cooperative pedestrians that do not cooperate is very challenging since the vehicle has to distinguish them from cooperative ones early enough to slow down and avoid these people carefully. This challenging scenario is referred to as partly cooperative crowd in this thesis. In the experiments, the cooperative part of the crowd is simulated with the regular [SPACISS](#). The non cooperative humans are also simulated with SPACISS but their field of view is virtually restricted to 1m, as if they were distracted, not aware of the presence of the vehicle until it comes closer than 1m. After experimenting different values, a proportion of 25% of distracted pedestrians is chosen. It guarantees at the same time that the crowd is mostly cooperative such that the vehicle can drive rather fast and that the vehicle often encounters and has to avoid distracted people. Experimentally, it has been observed that, with the proposed navigation system with the extended PCD, the navigation duration, Δt_{nav} , is increased in average by 18% as compared to a cooperative crowd of the same density and the shortest distance to pedestrians during avoidance maneuvers is reduced by 38%.

So a new set of tests proposes to evaluate and compare the safety of the collision avoidance system with the extended PCD, in cooperative crowds and in partly cooperative crowds. In simulations, the [experimental automated car](#) has to cross unorganized crowds of different densities that go in the opposite direction on a 8m wide road. Each test is repeated 10 times with a random initialization of the crowd.

The review of the state of the art does not reveal any consensus on a safety metric for human-aware navigation. Among the approaches that are listed in [[CCMZ18](#)], counting occurrences of collisions appears too simplistic since it would not reveal most of the dangerous events. Also defining a safety zone around pedestrians is not feasible since it would not capture the effects of speed (it is safe for pedestrians to be very near the car at low speed but not at high speed) and the effects of the kinematic constraints of the car (it is safe for pedestrians to stand on the side of the car but not in front of it). For these reasons, this thesis proposes to use the RSS satisfaction ratio, Q_{RSS} , as a measure of the navigation safety. This metric is defined and used in chapter [3.5.2](#).

So [fig.38](#) presents the comparative results of the safety evaluation of the navigation with the extended [PCD](#) in cooperative and partly cooperative crowds. It is important to notice that, contrary to the results of the safety evaluation of the original PCD in [fig.19](#), the navigation with the extended PCD results in a RSS satisfaction ratio which is below 1, meaning that the safety standard of [Responsibility-Sensitive Safety \[SSSS17\] \(RSS\)](#) are not satisfied. This is understandable since the new prediction model is less conservative and does not consider the worst cases as the [RSS](#) approach does. The [RSS](#) approach to safety is generally considered as overly conservative. So Q_{RSS} being lower than 1 should not be interpreted as an observation of a dangerous behavior. With both cooperative and partly cooperative crowds, no collision occurred and actually the RSS satisfaction ratio is almost unchanged. Despite partly cooperative crowds that visibly affect the navigation and significantly lower Δt_{nav} and reduce the

distance to pedestrians during tests, there is no noticeable impact on the safety metric.

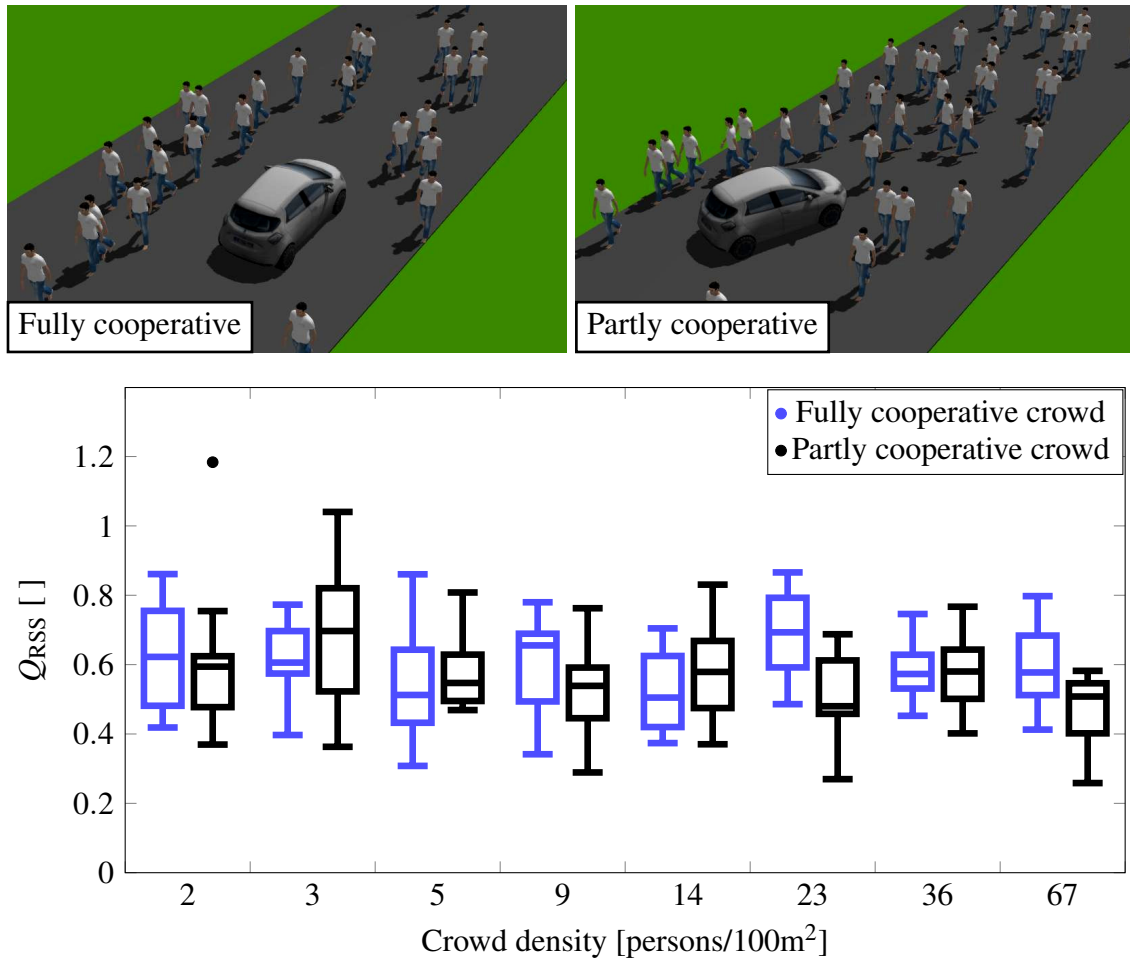


Figure 38 – 160 simulations are run in a randomly initialized scenario where the car has to cross a 25m long and 8m wide shared space with an unorganized longitudinal opposite flow of people. The top images illustrate this scenario with 23 persons/100m². The graph presents Q_{RSS} , the worst RSS satisfaction ratio during the scenario, depending on crowd density. Each boxplot displays the median, quartiles, extrema and outliers over 10 simulations. Half of the simulations use a non-cooperative crowd. The interaction-aware planner is used in all cases. Results show that the non-cooperation of some individuals of the crowd has no visible effect on the safety metric while it significantly affects the navigation efficiency. This proves that the planner with the extended PCD adapts its drive to guarantee similar safety standards whatever the driving conditions.

This experiment demonstrates that the proposed interaction-aware planner actively maneuvers and avoids dangerous situations to maintain a constant level of safety even if some people do not cooperate. This study shows that, even if the proposed planner benefits from the cooperation of humans, it actively avoids collisions with humans whatever their behavior. Therefore, the proposed collision avoidance solution with the extended PCD guarantees safety, even in dense crowds.

Pedestrian comfort Even though the approach that is proposed in this thesis does not include an explicit minimization of the disturbance it causes to humans, the comfort of pedes-

trians that interact with the vehicle is an important aspect of human-aware navigation. It is necessary for the acceptability of a navigation solution in a real-world application. Unfortunately this aspect is rarely validated with experimental results in the related research works. The disturbance that a vehicle causes to nearby pedestrians is not only related to the risk of collisions. The concept of disturbance encapsulates all undesired modifications of the trajectories of pedestrians i.e. the detours and speed changes that they make to avoid the vehicle. If the safety of pedestrians is the absence of any dangerous event, the comfort of pedestrians resides in the overall smoothness of all interactions over time.

In the literature, there is not yet a consensus on a metric to evaluate this comfort [CCMZ18]. Though the promising concept of Social Work has been introduced by [GS12] and reused in [FGS13] and [FS14]. Within a [Social Force Model](#) [MHG⁺09] (SFM) that represents the interactions between the vehicle and nearby humans, the social work is defined as the cumulative energy variation of pedestrians which is caused by the social interaction forces, by analogy with the definition of the work of a force in physics. It represents the sum of all the disturbances that a vehicle causes to nearby humans. Considering q humans $H_i, i \in \llbracket 1, q \rrbracket$ whose speed vectors are V_{H_i} , then the average Social Work per person, W , is defined by (13).

In order to apply this metric to the experiments of this thesis, the computation of the Social Work has been embedded in [SPACISS](#) and is based on its underlying [SFM](#). Even though the extended [PCD](#) also relies on a [SFM](#), it is not related in any way to one of this metric. This metric could actually also be used to evaluate planning methods that do not use a [SFM](#) based approach. The Social Work has the advantage to integrate on all the interactions that occur over time and to summarize them in one single measurement. The use of the Social Force Model also provides a well defined basis for this metric.

$$W = \frac{1}{q} \sum_{i=1}^q \int |F_{\text{interaction}} \cdot V_{H_i}| dt \quad (13)$$

In a new set of simulations, the [experimental automated car](#) has to drive through unorganized crowds of different densities that go in the opposite direction on a 8m wide road. Each test is repeated 10 times with a random initialization of the crowd. With the use of the Social Work, these tests aim to compare the disturbances that are caused to pedestrians by the vehicle when it is driven with the extended [PCD](#) and with the original [PCD](#). As the original [PCD](#) is highly conservative it is expected to be more human-friendly than the new proposed version of it. The purpose of this experiment is to determine by which amount the discomfort is increased. Unfortunately, for this experiment too, no comparison with the related works is possible. [Fig.39](#) displays the results of this comparison between the original kinodynamic [PCD](#) and the extended interaction-aware [PCD](#). In the least dense crowds, the planner with the interaction-aware [PCD](#) causes twice more disturbance than its kinodynamic counterpart. This is reasonable since the kinodynamic planner drives very conservatively while the interaction-aware planner tries to take advantage of the cooperation of humans. However they appear to cause the same amount of discomfort to crowds of average density. Surprisingly, experiments

in the densest crowds reveal that the interaction-aware planner causes less disturbance to them than the kinodynamic planner.

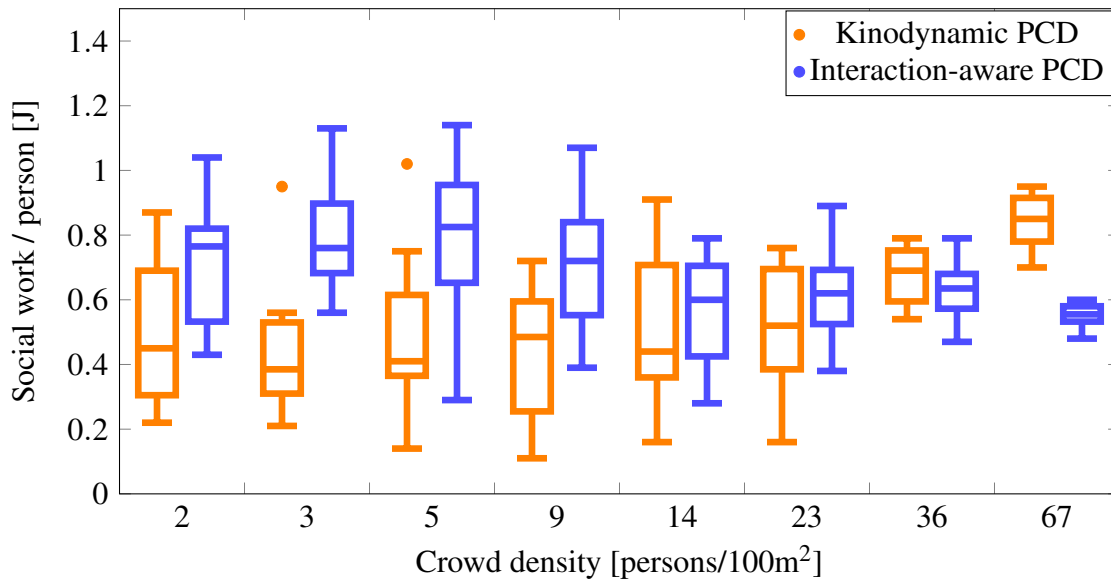


Figure 39 – 160 simulations are run in a randomly initialized scenario where the car has to cross a 25m long and 8m wide shared space with an unorganized longitudinal opposite flow of people. This scenario is the same as the one described in fig.37. The graph presents the social work which reflects the disturbance that both the interaction-aware PCD and the kinodynamic PCD planners cause to pedestrians. Each boxplot displays the median, quartiles, extrema and outliers over 10 simulations. The results show that the interaction-aware planner causes more disturbances to low density crowds than its kinodynamic counterpart. This is logical since the kinodynamic PCD is highly conservative. Yet the interaction-aware planner causes a reduced disturbance to humans in high density crowds. This is a surprisingly good result. It comes from the ability of this new proposed planner to merge into the flow of the crowd and to clear space faster.

The fact that the planner with the extended PCD is more human-friendly than the original planner can be explained by the fluidity of interactions. Visual observations of the experiments reveal that, when driven with the original PCD, the vehicle drives in average very slowly in dense crowds, resulting in bottlenecks for people trying to go around the vehicle. On the other hand, when driven with the interaction-aware PCD, the vehicle drives slightly faster, goes towards less densely crowded areas and naturally follows people that move in the same direction. So this experiment demonstrates the ability of the proposed collision avoidance with extended PCD to drive a car smoothly in very dense crowds, merging into the flow of humans and causing reduced disturbance to humans, in comparison with a conservative planning approach.

4.6 Chapter conclusion

In this chapter is proposed an extension of the previously proposed Predictive Collision Detector (PCD) based collision avoidance system. This extension aims to make the proposed solution capable of human-aware navigation. It is designed for navigation of an autonomous

car in densely crowded shared spaces with humans. This improvement relies in a modification of the occupancy prediction module of the [PCD](#) that makes it aware of the semantic classification of [agents](#) and that provides a specific model for human motion prediction.

The prediction model relies on a preexisting [Social Force Model \[MHG⁺09\] \(SFM\)](#) that has been adapted for human-vehicle interactions. While related works in human-aware navigation for cars chose proactive approaches that are limited by their own complexity, this thesis proposes a not proactive but interaction-aware approach which is not limited in the number of agents that it can simultaneously handle.

Furthermore, as the proposed approach is embedded in the [PCD](#) framework and benefits from the [Conditionnal Monte-Carlo Dense Occupancy Tracker \[RNL15\] \(CMCDOT\)](#) perception suite, it can operate under several real-world constraints that are not considered in comparable related works, such as perception only from sensor available data, uncertainty and partial knowledge of the scene, non-holonomic motion, actual shapes of ego-vehicle and obstacles, no restriction on the number and types of agents and obstacles.

The method is deployed on a prototype of autonomous car. While the new occupancy prediction is tested with real data, the collision avoidance is tested on the real car but with simulated humans via [Augmented Reality \(AR\)](#) on sensor data. These experiment prove that the proposed method could technically be transferred to real-world applications although it has not been done for experimental safety matters. Qualitative experimental results in [Software-in-the-Loop \(SiL\)](#) simulations show that the new planner is able to consider simultaneously different classes of agents. It is also able in some situations to minimize its impact on nearby humans but also to cooperate with them if needed, as proactive planners can do. A quantitative study measures the performance of the proposed extended [PCD](#) through series of [SiL](#) simulations. It demonstrate that the new version of the [PCD](#) is not subject to the [Freezing Robot Problem \[TK10\] \(FRP\)](#) and navigates more efficiently in dense crowds. It is also verified that this new avoidance solution actively contributes to safety and avoids collisions with pedestrian whatever their behavior. Finally, it is observed that the new planner causes a reduced disturbance to humans in highly dense crowds since it merges more smoothly in the flow of humans.

With this contribution, this thesis intends to expand the possibilities of usage of [Dynamic Occupancy Grids](#) for collision avoidance by introducing this perception technique in human-aware navigation. Besides, while proactive planners are difficult to deploy in real-world due to their inherent complexity and predictive planners are not efficient in crowded environments, the proposed solution stands as a compromise that is capable of efficient human-aware navigation and could be transferred to real-world applications.

This contribution is an opportunity for several future works. First, for safety matters, the method could not be evaluated with actual human behaviors. A validation in presence of actual humans is an important future step. Second, the human motion prediction model assumes that human intentions are unknown. Yet these intentions could be detected, it would drastically improve the accuracy of the interaction-aware prediction. Finally, this extension of the [PCD](#) opens way for development of class specific prediction models but only a model

for pedestrians has been proposed in this thesis. This framework could be enriched with specific prediction models for other classes of [agents](#), such as cars or cyclists.

Chapter 5. Proposed Augmented Reality test framework

5.1 Chapter outline

Simulated tests were not sufficient to validate the [Predictive Collision Detector \(PCD\)](#) collision avoidance framework. Real-world tests on the [experimental automated car](#) were limited too, due to experimental constraints. Hence this thesis proposes a new approach for testing automotive software. This approach provides an intermediate between simulation and real-world experiment. It has been used in the validation of the previously proposed collision avoidance system. This new approach is designated as [Augmented Reality \(AR\)](#) on sensor data, it is presented in this chapter.

The first section, [5.2 Testing automotive collision avoidance software requires new test frameworks](#), explains why there is a need to bridge the gap between simulation and real-world testing. Then section [5.3.1 State of the art of mixed reality testing for automotive software](#) reviews the different approaches to combine simulation and reality for testing automotive and robotic software. An analysis of the state of the art reveals that the most relevant of these approaches actually bypass the perception software and produce object-based representations of the mixed-reality environment. It would not make sense to use a such method to challenge the [PCD](#) which is dedicated to operate with a perception system and without object-based representations. Therefore a [AR](#) new testing approach is proposed and explained in section [5.3.2 Proposed approach of Augmented Reality at sensor level](#). Its framework and its key component, the LiDAR data augmentation function, are described in section [5.4 Methodology of the proposed Augmented Reality \(AR\) solution](#) with some hints to optimize its implementation in [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#). After presenting the details of the deployment of this test framework on the [experimental automated car](#), the last section [5.5 Implementation and experimental validation of the proposed AR framework](#) validates the approach with experimental qualitative and quantitative results. Qualitative results show that sensor data augmentation does not affect the shapes of objects and does not alter real data. A quantitative comparison of real and augmented data demonstrate that the augmentation is seamless for the software under test, the [PCD](#) avoidance system.

The contributions that are presented in this chapter are :

- the design of a new test framework for automotive software that enables [Augmented Reality \(AR\)](#) directly on sensor data
- the data fusion methodology that allows real-time augmentation of LiDAR sensor data
- both qualitative and quantitative experimental results to validate the applicability of the proposed method

These contributions have been published in :

Thomas Genevois, Jean-Baptiste Horel, Alessandro Renzaglia, and Christian Laugier. Augmented reality on lidar data: Going beyond vehicle-in-the-loop for automotive software validation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2022

5.2 Testing automotive collision avoidance software requires new test frameworks

Automated Driving Systems (ADS) and Advanced Driver Assistance Systems (ADAS) software have shown an outstanding development in the last decade. The performance of this software is nowadays reaching and maybe going beyond the level of human drivers' skills. This is bringing a drastic change in the automotive industry. Despite this outstanding progress, few of the recent research developments have been applied to prototype vehicles and extremely few have been transferred to commercial applications. This is mainly due to the lack of convenient testing tools and validation procedures.

The automotive industry commonly uses the V model development. This software development procedure associates every step in the conception of a software with a dedicated test. Hence the software is validated when all of its submodules have passed the required tests. The automotive industry also adopts well defined tests and validation procedures such as the ones of Euro NCAP [S19]. These validation procedures guarantees standards of safety, reliability and quality. ADS are critical systems that require such a validation. Under the European Artificial Intelligence Act [M21], ADS are classified as high-risk systems. Hence, for their commercial deployment, it will be mandatory to demonstrate through testing that they satisfy safety requirements. However, the complexity of recent AI-based algorithms poses new challenges in terms of testing and validation. Namely, deep learning software can not be formally proven since it remains a black box even for its designers and the V model development does not apply to such software since it is not modular. In the absence of formal proof and modular testing, software validation requires extensive testing.

Simulation environments are a commonly adopted solution for testing. Nowadays simulators are getting more and more realistic and they enable repeatable, fast and cheap testing in diverse scenarios. However, the **representability** of tests in simulation to validate real systems is not proven. The simulation models do not reflect exactly real-world phenomena and there exist numerous unexpected or not modeled events that make reality significantly different from simulation. This difference is often designated as the reality gap. Because of this reality gap, pure simulation does not suffice for the validation of critical automotive software.

So there is a need for real-world testing of automotive software. Yet this real-world testing is time-consuming. It is also costly. It can be dangerous since critical scenarios, where it is more relevant to test the systems, involve potentially dangerous situations, such as unexpected behaviors of other road actors (cars, pedestrians, bikes, etc.) and may lead to collisions. To test the systems without risking fatalities during the tests, other road actors are represented by actuated dummies and other soft targets [GAS17, A23]. Fig.40 presents examples of such test targets, a fake car, a fake scooter and a pedestrian dummy with its actuator. But dummies and soft targets can not be deployed in large amounts that would be needed to figure crowded scenarios such as what has been presented in the validation of the **Predictive Collision Detector (PCD)**. Dummies and soft targets do not always guarantee the repeatability of tests. Moreover they do not reproduce realistic motion of humans and cars

and are not versatile enough to generate a significant diversity of scenarios.



Figure 40 – These pictures display some examples of actuated dummies and soft targets that are used to test **ADS** and **ADAS**. These images are from www.abdynamics.com [A23].

As a result, nowadays the most common validation procedure currently used in industry is to drive a fleet of prototype vehicles for millions of kilometers and the main metric to assess the performance remains the average distance run without an error or a disengagement. New approaches [RFNV21] emerge but automotive software validation still needs more convenient testing tools to enable a proper validation procedure and quality assessment.

This chapter proposes a new framework based on **Augmented Reality (AR)** for testing advanced automotive software. The key aspect of this method is the design of a merge function allowing a real-time augmentation of LiDAR data with virtual elements. Fig.41 illustrates this concept. From an initial point cloud from the sensors of the **experimental automated car** and with a virtual point cloud from simulation, the AR system generates in real time a fused point cloud that corresponds to what could have been perceived in an augmented scene. For a better understanding, a visualization of the augmented scene is also computed online and displayed in this figure.

Augmented Reality opens new possibilities for testing. A test site can very easily be populated with many and diverse virtual elements in order to create complex test scenarios. Virtual pedestrians or cars are easier to operate and offer rich and active behaviors (e.g. reacting to the ego-vehicle's motion). Furthermore, all elements of the test scenario that may induce a collision risk can be replaced by their virtual counterpart to secure the tests in the early stages of development or to test the system in critical situations. Virtual scenarios are also repeatable and this is a key feature to reproduce experiments.

The proposed **AR** testing implementation accurately represents the virtual scenes, preserves real sensor data and guarantees a consistent fusion of real and virtual worlds. So **AR** tests produce meaningful results that can be used to infer the behavior of the vehicle in the real world. Finally, as any element can be either real or virtual, **AR** testing offers a smooth transition from simulation to actual testing. For these reasons, the proposed **Augmented Reality (AR)** framework can be a fundamental testing solution for the validation of advanced automotive software.

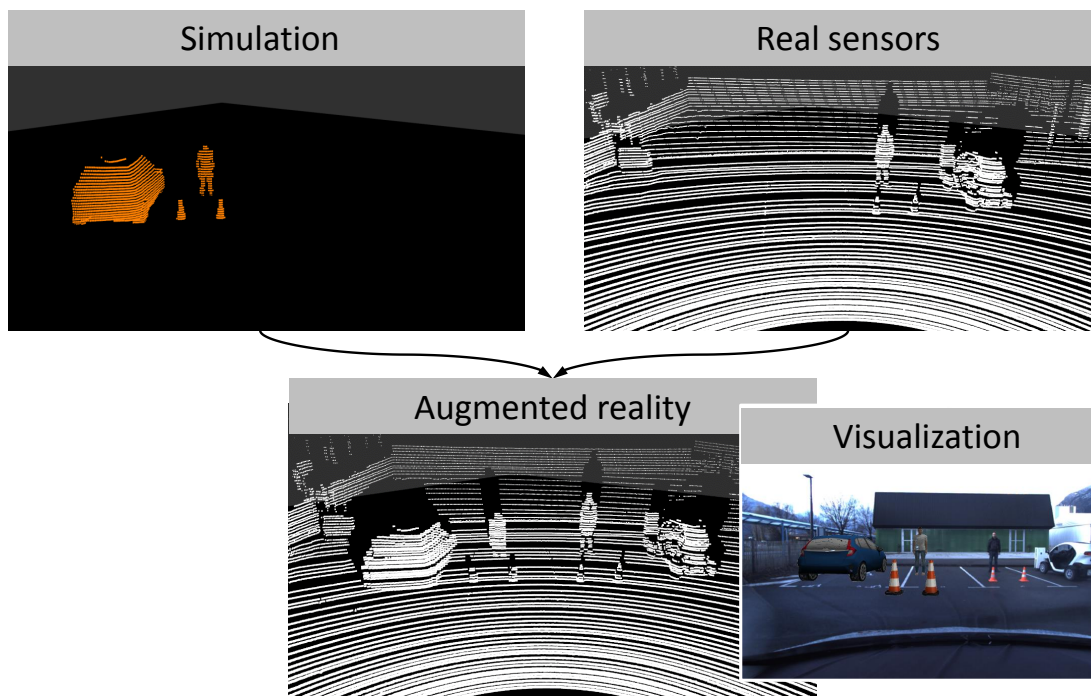


Figure 41 – Illustration of the principle of LiDAR point cloud augmentation with an example that has been recorded on the *experimental automated car*.

5.3 State of the art of automotive mixed reality testing and originality of the proposed approach

5.3.1 State of the art of mixed reality testing for automotive software

On the one hand, the reality gap is a limit to the [representability](#) of tests in simulation. On the other hand, real-world tests are costly, time consuming, they might be dangerous and they may not even cover a sufficiently diverse set of scenarios. Over the years, the industrial and research communities have worked to bridge the gap between simulation and reality in order to obtain hybrid test tools that would overcome these limitations.

Augmenting simulation with elements from the real-world A first approach for this is to embed real components in a simulation framework. In this way, the concept of [Software-in-the-Loop \(SiL\)](#) consists of the execution of the software under test in a virtual environment. The software under test is executed as it would be on the real-world system ; its inputs are provided by emulated sensors and its outputs affect the virtual world through virtual actuators. So the software under test can be tested in a realistic software environment. A large amount of diverse tests scenarios can be simulated, without any risk and at a reduced cost. Yet [SiL](#) does not accurately represents physical systems, sensing and computing hardware.

A more realistic approach, [Hardware-in-the-Loop \(HiL\)](#) involves the software under test with the actual computing hardware in simulated tests. The interest of this approach is that it challenges the software under test in a realistic computing environment. It reveals potential

communication, synchronization and multi-threading issues. It shows the actual usage of memory and computing power. It can be used to optimize the execution of the software on its dedicated hardware. However the system remains virtual in a virtual environment. So all of the sensing, internal and external physical systems and other agents remain approximated by simulation models. **SiL** and **HiL** do contribute to the validation of automotive software but do not suffice since vehicles contain several physical systems that are not accurately modeled in simulators.

Hence the concept of **Vehicle-in-the-Loop (ViL)** proposes a more realistic testing than **SiL** and **HiL** for automotive applications. This method involves the whole real vehicle under test in an entirely virtual environment. **ViL** has been widely used for automotive testing over the last decade. Various implementations have been realized and designed for different purposes but a common structure can be observed [LLG18]. The base idea of **ViL** is to execute the software under test, on the actual computing hardware, on-board of the real vehicle in motion. The state and actions of the actual vehicle are updated in a virtual environment which is then perceived by emulated sensors that replace the actual sensors. While the test happens entirely in the virtual environment, the actual vehicle may be standing on a test bench [AD10] or driving on an empty road [BMF07]. Under the **ViL** approach all the internal physical systems, such as for example the powertrain or the tire-road contact, are actually operating and do not need to be approximated by simulation models. So **ViL** is way more realistic than **HiL** for automotive testing. Still, the environment remains purely virtual, the testing is limited by the accuracy of the sensor emulation, the simulator realism and the diversity of realistic test scenarios. Even though **ViL** is a highly effective tool that contributes to the validation of automotive software, the gap between **ViL** and actual vehicle testing is too large for **ViL** to replace an extensive real-world testing in a validation procedure.

Augmenting the real-world with simulated elements The other approach to bridge the reality gap is to bring virtual elements to the real world. With this approach, several recent works have achieved a significant improvement to go beyond **ViL** by combining virtual and real worlds to provide versatile and realistic hybrid test environments. The fundamental concept of these methods is to introduce, in real-time, virtual elements in the actual **perception** of the vehicle. It is an augmentation of real test scenes with virtual elements, in the internal representation of the navigation system. So these methods are referred to as **Augmented Reality (AR)**. In the literature, the same concept is also sometimes named mixed reality. A major constraint of **AR** is that the augmentation process must be executed in real-time, without adding any significant delay to the perception process. While **ViL** consists in having the real vehicle in a virtual world, **AR** proposes to have the real vehicle in a real environment in which some virtual elements are added. So **AR** offers simultaneously safe, efficient and repeatable testing with some virtual elements but also rich, dense and realistic environments. **AR** is extremely versatile since any **agent** or element of the background can either be real, for representability and realism, or virtual, for repeatability and safety. In this, **AR** is a significant step towards realism when compared to **ViL**. Yet there are different approaches to **AR**.

Augmented Reality at object level The most common approach in recent works is [AR](#) at object level. This approach proposes to introduce virtual elements in an object-based perception of the surroundings of the vehicle, directly as objects without sensor emulation. It relies on a representation of the world as a list of objects with their position, speed, status, classification and main characteristics. The first advantage is that the augmentation process is straightforward and directly applicable for navigation systems that use object-based perception. Also the object-based representation of virtual elements is frugal enough to be shared through [Vehicle-to-everything communication \(V2X\)](#) communications to have multiple vehicles and devices perceiving the same augmented environment. In [[CCX⁺20](#)], Chen et al. present a unified fusion data format to represent the augmented scene at object level. Also, an implementation of object level [AR](#) has been realized at the MCity facility [[FYX⁺18](#)], it relies on the [V2X](#) communication protocol to introduce and share virtual objects. Another implementation has been developed for the ZalaZONE proving ground, it introduced the concept of Scenario-in-the-Loop [[S21](#)]. SciL offers a mixed reality framework where virtual objects are introduced on a communication protocol based on 5G. However [AR](#) at object level does not accurately challenge navigation systems since it bypasses the perception process which is probably the most critical part of [Automated Driving Systems \(ADS\)](#). The direct introduction of virtual objects in the perception ignores the difficulty to sense and classify these objects, potential occlusions, limited range of sensors and all the potential errors, misclassifications and uncertainties that come out of the perception process. Moreover [AR](#) at object level is meaningless to navigation systems that do not use object-based perception, such as for example end-to-end learning-based approaches or grid-based approaches like the proposed [PCD](#) framework. A different approach is possible.

Augmented Reality at sensor level Instead of introducing virtual elements as objects after the perception process, it is possible to emulate virtual sensors to perceive them and introduce them in sensor messages before the perception process. This recent approach that is designated as [AR](#) at sensor level consists in augmenting the observations of the sensors with virtual elements, in the sensor messages as they come out of the sensor drivers. The real detections and the format of these sensor messages are preserved so this form of [AR](#) is entirely seamless for the software under test. Virtual elements follow the same perception process, just as real elements. It also enables to test the whole navigation system, including the perception. Another major feature is that [AR](#) at sensor level enables the introduction of virtual elements that can not be modeled in object-based representations but are relevant in testing, such as for example, elements of the background, ground irregularities, weather effects on sensing and sensing noise. Yet [AR](#) at sensor level implies heavy computations that must be executed in real-time for the augmentation process, and still it may introduce sensing delays. Also it requires an strict synchronization of the virtual world otherwise observations of virtual elements would be misplaced or delayed. Finally [AR](#) at sensor level depends on the existence of a realistic and real-time augmentation function for every sensing modality of the vehicle. There are many common automotive sensing modalities for which the current state of the art

of AR at sensor level does not propose any such process to merge the virtual data into sensor messages. In the implementation of Scenario-in-the-Loop at the ZalaZONE, [S21] describes the emulation and augmentation of a very simple range sensor, it can be regarded as AR at sensor level. Hildebrandt et al. introduced a sensor level AR approach with the concept of World-in-the-Loop [HE21]. WiL has been implemented on a drone, it augments the images of an on-board monocular camera. More recently Argui et al. deployed a similar approach for a depth camera on a wheeled mobile robot [AGA23]. In another recent work [SCL⁺23], Shen et al. considered a realistic rendering of lighting conditions for camera image augmentation. In the Sleepwalker framework [ZEF⁺18], Zofka et al. have proposed a solution for AR with smart LiDAR sensors. However, due to requirement for real-time performance during live signal augmentation, the augmentation has not been carried out on the raw point clouds but on the simplified feature sets that result from an internal black box process of these smart sensors. Therefore the realism of the augmentation process can not be guaranteed and this approach could not be applied to test perception systems that directly process point clouds.

Offline augmentation of sensor data Other research works propose offline augmentation LiDAR point clouds. In [FZY⁺20], the authors use real LiDAR data recording as background to add annotated and simulated point clouds of virtual objects. In [VRGBC21], the authors simulate adversarial weather conditions by adding impacts on simulated water sprays to real LiDAR point clouds. Yet these 2 last approaches can not be regarded as AR since they operate offline and not in real-time.

5.3.2 Proposed approach of Augmented Reality at sensor level

The collision avoidance system with PCD framework that is proposed in chapters 3 and 4 is dedicated to operate in complex and crowded environment with a real perception system under partial observability and uncertainty. The perception system on the experimental automated car uses LiDAR sensors. Therefore its proper validation requires its testing in realistic perception conditions with LiDAR sensors. For this, this thesis proposes a new framework to realize Augmented Reality (AR) at sensor level on LiDAR data. The originality of this framework is that, unlike the related works it relies on the structure of a ViL system. That is why, if it considers only virtual elements, the proposed solution is equivalent to ViL. Conversely, if it does not consider any virtual element, it is equivalent to real-world testing. So, via addition and removal of virtual elements, the proposed framework provides a unique smooth transition from ViL to real-world testing. This offers new opportunities for testing and validation. Also, since the proposed framework does not require neither a mapping neither a classification of virtual and real objects, it is highly versatile. Any element of the real world is directly taken in account and any simulated element is introduced in AR. Another difference of the proposed approach is that it relies on an accurate localization. The localization is used to bond the virtual environment to the real world. This strong connection enables realistic sensor emulation, without filtering. Regarding the augmentation of LiDAR

data, the proposed solution differs from what has been presented in [ZEF⁺18]. Instead of augmenting feature sets, the processed output smart LiDAR sensors, this thesis proposes the augmentation of large unprocessed point clouds. This contribution also differs from the other above-mentioned works in offline LiDAR data augmentation because this thesis considers this augmentation in real-time and on-board to generate AR.

5.4 Methodology of the proposed Augmented Reality (AR) solution

5.4.1 Structure of the proposed AR framework

This section presents the proposed framework to introduce AR in real-time in the actual sensing of the vehicle. Even if the proposed AR system does much more than ViL, it has the typical architecture of a ViL system and several modules in common. Reusing the ViL topology which is described in [PCL20], this thesis proposes an AR framework which consists of the four following modules:

- a virtual environment which contains a twin of the experimental vehicle
- a synchronization module which updates the position and state of the virtual twin
- a sensor emulation which generates outputs from the virtual sensors and integrates them in the actual sensors' outputs
- a visualization which helps testers to understand the AR scene.

Compared to a ViL system, the originality of the proposed approach lies in the sensor emulation process that merges real and virtual data in order to augment the real scene with virtual elements. Fig.42 proposes a schematic representation of this software framework. The periodic messages of the sensors of the real vehicle give rhythm to the virtual world. So all modules must run in real time, their execution duration must be short compared to the period of the sensors. This is a heavy constraint on the design and implementation of each module.

Virtual environment A virtual environment is generated in simulation. It is anchored to a real world position with a reference in GPS coordinates. Then, this virtual environment contains only a virtual twin of the vehicle under test and the virtual elements that are desired in the Augmented Reality scene. The virtual twin must be a faithful representation of the real vehicle with an accurate positioning of the sensors. There is no restriction on the type, nature and number of virtual elements in the augmented scene, as long as they can be simulated and perceived by the emulated sensors. The scene can be as complex as required by the test and include any type of object, road actor, static element, background or ground irregularities ; the only limits are the ones of the simulator. Apart of the virtual vehicle and the test elements, the virtual environment is empty. The proposed method does not require a background, a ground plane nor any representation of the actual test site. This makes this method versatile and easy to deploy in a new place.

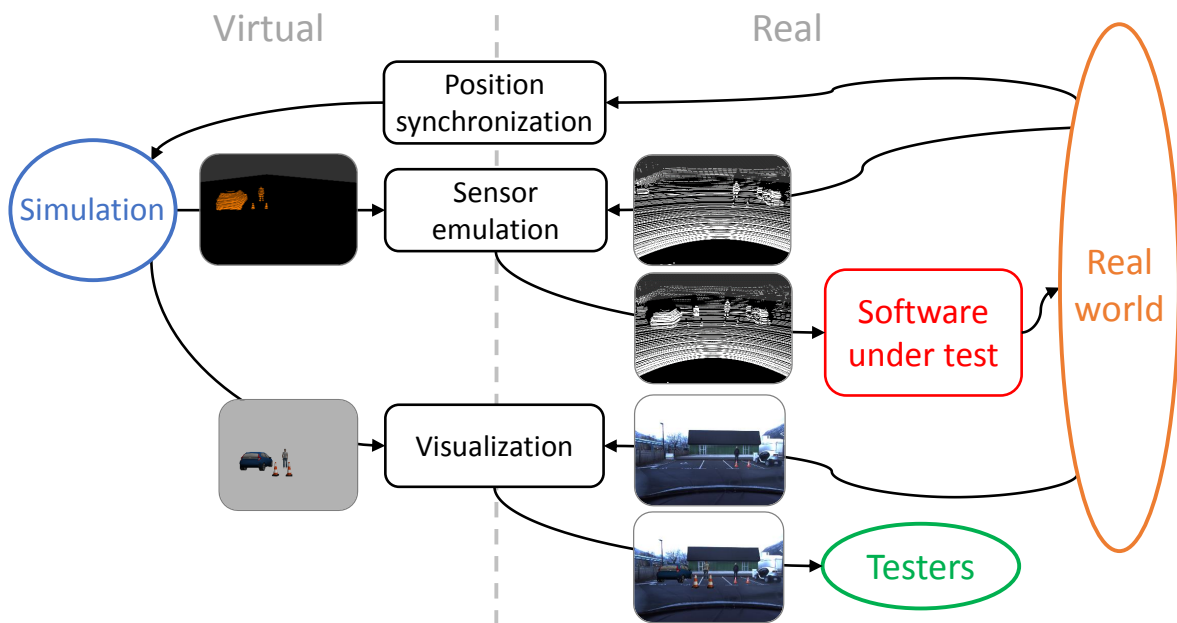


Figure 42 – Schematic representation of the structure of the proposed Augmented Reality framework.

Synchronization The absolute position of the vehicle under test must be constantly estimated by an accurate localization system. Then the estimated position is used to set the position of the virtual twin of the vehicle under test in the virtual environment such that the observations of the twin match those of the real vehicle. This straightforward synchronization gives a great flexibility. The AR system can be deployed without any installation. Yet the correct placement and motion of virtual elements depend on the accuracy of the localization system.

Sensor emulation The virtual twin of the vehicle is equipped with a set of sensors that mimic the sensors of the actual vehicle. An accurate, realistic and real-time emulation of the sensors is needed. Although the framework is generic, this thesis focuses here on LiDAR sensors. It considers LiDARs sensors that provide their observations as unprocessed point clouds. This is the most common format for this sensing modality. So the emulated LiDARs must return the detection of the virtual objects under a point cloud format. The point clouds are then merged with those returned by each corresponding actual sensor. This augmentation process is a key component of the proposed AR framework: it must be real time despite the amount of data to process ; it must consider a realistic sensor model ; it must reproduce all occlusions between real and virtual world. For each sensor, the augmentation produces a new point cloud that represents the perception of the Augmented Reality scene. It can then be processed by the software under test in place of the actual sensor's point cloud. Thanks to this, the use of AR is seamless for the software under test. Even if it has been designed for LiDAR point clouds, the proposed augmentation function could be easily adapted to process data under the format of LiDAR laser scans instead.

Visualization The virtual twin of the vehicle is also equipped with a set of cameras that mimic the ones of the actual vehicle. Thanks to the simulator, the virtual cameras return images of the virtual objects. These images are then merged with those of each corresponding camera. For each camera, this produces a new image that represents the AR perception. Even though it is just like AR on camera data, the proposed camera augmentation has only been done to provide the testers with a convenient insight of the AR scene. Due to the poor quality of the image rendering, this augmentation has not been used to feed the software under test. But if using a photo-realistic simulator and a realistic image merge function, this visualization could be used as AR for perception with cameras. This has been considered in several research works [HE21, AGA23, SCL⁺23]. However, a simulator with approximate graphics and a simple merge procedure suffice for the purpose of visualization.

5.4.2 LiDAR point cloud augmentation for AR at sensor level

The actual augmentation process of LiDAR point clouds, the integration of virtual LiDAR data into the point clouds from real LiDAR sensors, is one key component of this contribution. It is what makes the difference between ViL and AR. The ViL approach is to replace real sensors' observations by what virtual sensors perceive of the simulated environment. The sensor level AR approach is instead to introduce observations of the simulated environment into the outputs of the real sensors, preserving the sensing of the real-world and the format of the original sensor message.

Let us consider one of the actual LiDARs and its corresponding virtual sensor. Both periodically produce point clouds of their detections but they are generally not synchronized. After a manual calibration, the virtual sensor is assumed to have the same position on the vehicle that its corresponding actual sensor. The position of the virtual vehicle is updated at a high frequency thanks to a fast and accurate localization system. So, at any instant, it is assumed that the position of the actual and virtual sensor in a global frame match each other.

The real-time emulation of the virtual sensor is required to produce the point cloud of the virtual scene yet, for a high resolution 3D LiDAR sensor, this process is very demanding in computing resources. Hence it may be needed to use a virtual LiDAR sensor with a different resolution, not as good as the actual sensor. Therefore the requirements for the augmentation function are that it should merge two point clouds, sensed from the same position but that are not necessarily synchronized and may have different resolutions, the resolution of the real sensor being the highest.

There is also a hard real time constraint on this augmentation function. It should not introduce any significant delay in the perception process. So its execution duration must be short compared to the period of sensor. LiDAR point clouds consist of large amount of data, about a million points to process every second for a single dense 3D sensor. For sake of real time, the augmentation function is designed to be executed in [Global Purpose computing on Graphics Processing Unit \(GP-GPU\)](#) and parallelized over the numerous points with as few memory transfer as possible.

Let \mathcal{P}_s be the point cloud that the real sensor produces. \mathcal{P}_v denotes the point cloud coming from the virtual sensor. And \mathcal{P}_f represents the fused point cloud which results from the augmentation process. Let p be a point whose Cartesian coordinates are (x, y, z) and spherical coordinates are (r, θ, ϕ) . For every point cloud provided by the virtual sensor, the system computes \mathcal{R}_v , the array of equivalent spherical coordinates. The implementation of this computation requires it to be sampled. If the resolution of the virtual sensor is known and uniform, the sampling can be at the same resolution. Otherwise the sampling resolution must be small compared to the resolution of the virtual sensor.

$$\mathcal{R}_v = \{(r_v, \theta_v, \phi_v), \forall p_v \in \mathcal{P}_v\} \quad (14)$$

Let R be the function that returns the range coordinate of a point. Then it is possible to define the function R_v that returns the range of the point in \mathcal{R}_v which is the closest to the given angular coordinates (θ, ϕ) . In its implementation, the computation of this function does not require a optimization search. Instead it exploits the structure and the ordering of the sampling of (14).

$$R_v(\theta, \phi) = R \left(\arg \min_{(r_v, \theta_v, \phi_v) \in \mathcal{R}_v} \left\| \begin{bmatrix} \theta_v - \theta \\ \phi_v - \phi \end{bmatrix} \right\| \right) \quad (15)$$

The frequency of the virtual sensor must be set such that \mathcal{R}_v is updated faster than the actual sensor's frequency. (14) and (15) are implemented as two sequential operations in a single GP-GPU kernel which is executed at every point cloud received from the virtual LiDAR. Then, for every point cloud that the actual sensor provides, the fused point cloud is computed, first in spherical coordinates, \mathcal{R}_f :

$$\mathcal{R}_f = \{(\min(r, R_v(\theta, \phi)), \theta, \phi), \forall p \in \mathcal{P}_s\} \quad (16)$$

(16) defines the fused point cloud such that it considers all occlusions between real and virtual objects. This augmented point cloud represents what would have been detected by a LiDAR sensor in the AR scene. (16) enables virtual objects to correctly occlude actual objects even though the point cloud does not have the same density. It also preserves all points from the actual point cloud, as long as they are not occluded.

This is an exact model to represent all occlusions, real elements occluding virtual ones and virtual elements occluding real ones. This management of the occlusion does not require the occluding object, either real or virtual, to be referenced or mapped in the framework. In this way, road actors (cars, pedestrians), but also, static elements (trees, walls, fences), ground irregularities (heaps, slopes) and weather effects (rain, snow, fog), either real or simulated, can be occluded or create occlusions in the augmented scene. Moreover (16) guarantees that real measurements are unchanged unless they are occluded. Furthermore, since it returns a value for every point of the sensor's point cloud, it maintains the exact format of the real sensor data, number of points, resolution and layers structure. Finally, with (17), \mathcal{R}_f is

converted in Cartesian coordinates to get \mathcal{P}_f . Then \mathcal{P}_f is sent to the software of the vehicle under test instead of \mathcal{P}_s . This simple rerouting of the sensor message is enough to make AR happen, it is seamless for the software under test.

$$\mathcal{P}_f = \{(x, y, z), \forall p \in \mathcal{R}_f\} \quad (17)$$

(16) and (17) are implemented together in a second GP-GPU kernel which is executed at every point cloud received from the actual LiDAR.

Fig. 43 illustrates the principle of sampling and occlusion introduced by (15) and (16). In the whole process, the array of ranges that corresponds to the sampling of $R_v(\theta, \phi)$ is the only element which is stored in memory. It is created at the execution of the first GP-GPU kernel with (14) and (15) and it is read in the second kernel in (16). There is no other memory transfer. Even though this algorithm involves the manipulation of large arrays, the computations are individually simple and parallelizable. So the implementation in GP-GPU is very efficient and it makes this application frugal and feasible in real time.

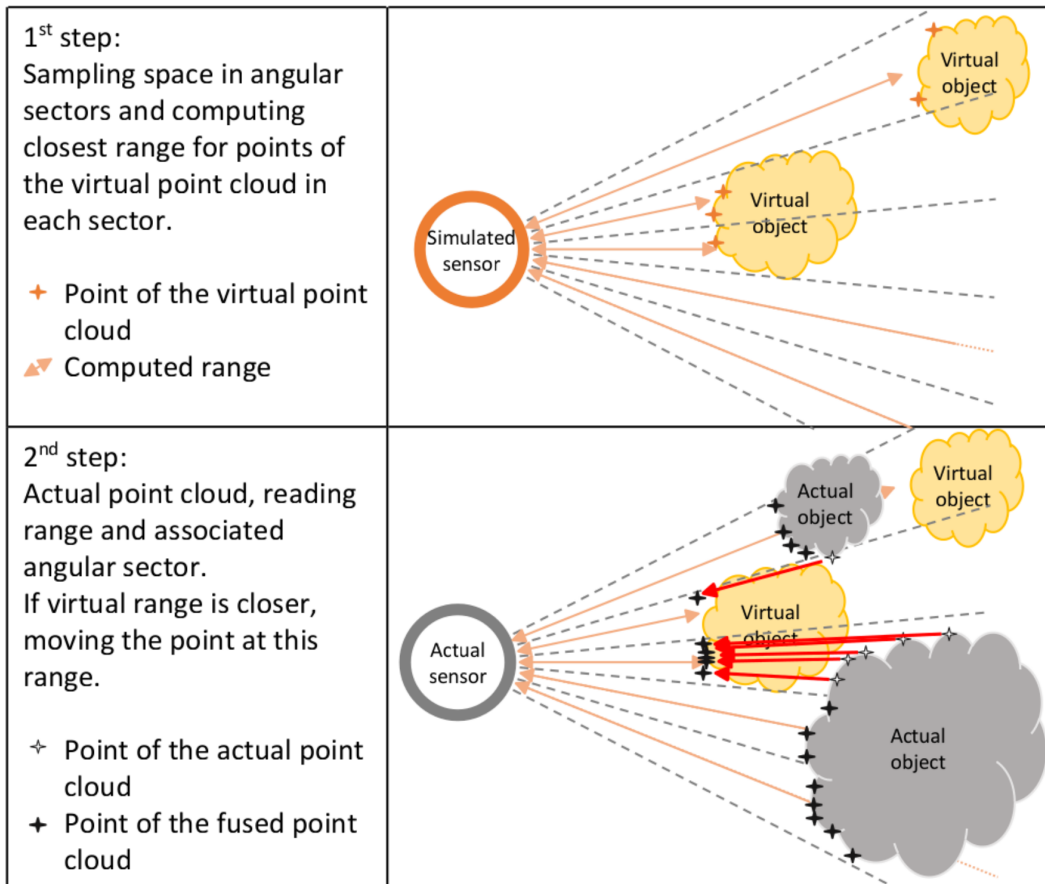


Figure 43 – Illustration of the principle of the proposed point cloud augmentation process. This approach preserves the angular density of the actual sensor message. It also accurately represents occlusions of both virtual elements by real ones and real elements by virtual ones. The first step, which corresponds to (14) and (15), is parallelizable over the angular sectors. The second step, which corresponds to (16) and (17), is parallelizable over the real sensor’s point cloud.

The augmentation process has been described for point clouds, although it could be immediately applied to laser scans that would correspond to the spherical representations of point clouds.

5.5 Implementation and experimental validation of the proposed AR framework

5.5.1 Procedure for the validation of the proposed AR framework

This section describes the deployment and validation of the proposed AR framework on an experimental automated car. The representability of the Augmented Reality system is defined as the capability of the system to reproduce the conditions of real-world tests. If the AR system is representable, the tests that are conducted in the augmented environment generate the same outcome than real-world equivalent tests. Therefore this does not mean that the augmented data should be equal to what it would have been in a real test. But the representability implies that the augmentation must be transparent to the software under test. So the goal of the present validation procedure is to demonstrate that the implementation of the proposed AR framework is representable i.e. that AR generated scenes correctly represent real test scenarios. In this purpose, the present section provides an analysis of the representability through 3 criteria. First, it is verified that the data augmentation is fast enough to be real time and that it does not introduce any significant delay in the perception system. Second, it will be qualitatively observed that the point cloud augmentation preserves the integrity of the shapes of some perceived objects. Third, there is a risk that inaccurate position synchronization or faulty point cloud augmentation introduce wrong positioning and displacements of the virtual objects in the augmented scene. So it will be experimentally demonstrated with a quantitative survey that the augmentation process guarantees the consistency of the position and motion of virtual objects with respect to the vehicle under test. This validation does not investigate how the real objects are transposed to the augmented data because the proposed augmentation principle does not alter the sensors' data unless there is an occlusion. Since this principle and its implementation are straightforward, it is assumed that it does not require an extensive validation but only a visual verification. Altogether, this validation aims to prove that the augmentation process is real time and that it does respect the shape, position and motion of the perceived objects, either real or virtual.

5.5.2 Implementation and real time execution of the proposed approach

Implementation The proposed Augmented Reality (AR) system is deployed on the experimental automated car. This vehicle is equipped with 1 high resolution 3D LiDAR sensor, a Velodyne HDL64, and 3 other LiDAR sensors, Ibeo Lux, with a lower resolution and limited field-of-view. The AR is simultaneously applied on these 4 sensors such that they perceive the same augmented environment. An instance of the LiDAR data augmentation function runs for each sensor. The whole AR framework is executed along with the software under test on

the on-board computer which is equipped with a NVIDIA Titan X GPU. [NVIDIA Compute Unified Device Architecture \[L08\] \(CUDA\)](#) is used to implement the method in [GP-GPU](#). The visualization of the AR scene is generated from an on-board monocular camera. The Gazebo simulator [\[O23\]](#) is used to generate the virtual environment. The test scenarios are constituted of common urban road scenes with pedestrians, cars, constructions cones that have been brought either to the real environment either to the virtual one. The car is driven at various speeds below 30km/h.

Real time performance of the augmentation process As it has been mentioned in section [5.5.1](#), there is a hard real time constraint on the process of augmentation of sensor data. In fact, this process introduces a delay between the acquisition of data by the sensors and the beginning of the perception process. To prevent any significant alteration of the tests and guarantee representability of the system, the augmentation processing delay must be negligible compared to the sensing period. On the [experimental automated car](#), the Velodyne HDL64 LiDAR sensor provides an observation every 100ms. Experiments show that this dense point cloud of about 150000 points is augmented with the proposed [AR](#) framework in a [CUDA](#) implementation in 10ms. In parallel, the 3 Ibeo Lux LiDAR sensors provide a sensor message every 40ms. Experimental result show that these point clouds are augmented in 0.6ms. Therefore it can be assumed that the additional delay that the augmentation process causes is negligible. These experiments have been realized with a realistic implementation of the proposed [AR](#) system, with a balanced use of the CPU and GPU resources, such that most of the computing power remains available for the software under test.

Real time performance of the simulation After demonstrating that the augmentation of the sensors' data is real time, it is necessary to make sure that the evolution of the virtual environment and the emulation of the virtual sensors follows the real time too. In their classical use cases, simulations do not have to follow the real time. Simulators often take advantage of this and slow down the simulated time when they can not execute all their computations to satisfy a real time execution. However this is not acceptable for an [AR](#) application. In the proposed [AR](#) framework, the simulator must run in real time. This constraint might be hard to satisfy since the simulation of complex scenes and especially the emulation of the sensors cause heavy computations.

The proposed implementation has been realized with the Gazebo simulator [\[O23\]](#). In this simulator, a naive straightforward simulation of the experimental car and its sensors would not run in real time. However it is possible to achieve a dramatic acceleration of this simulation with few adaptations. First, while Gazebo proposes a CPU-based emulation of LiDAR sensors, it is possible to emulate these sensors in GPU. It significantly reduces the computing time of the emulation of high-resolution LiDAR sensors. Second, as the virtual scene only includes elements whose motion is only kinematic, predefined or scripted, the physics engine of the simulator can be disabled. Finally, in the absence of the physics engine, there is no need for a fine time sampling of the simulation. The simulation rate can be adjusted to cor-

respond to the period of the virtual sensors. Altogether this produces a major acceleration of all the computations that are required for the simulation of the virtual environment and the emulation of the sensors.

With this, it became possible to simulate in real time very complex and crowded virtual scenes and to introduce them in Augmented Reality on the LiDAR data. For example, the scene which is displayed in fig.34 involved 340 pedestrians that were introduced in AR on the sensors of the car. These virtual humans were interacting among them and with the car, their behavior was simulated with [Simulation of Pedestrians and an Autonomous Car in Shared Spaces \[P21\] \(SPACISS\)](#). The simulation was real time without using too much of the computer's resources. This AR enriched experiment contributed to the validation the human-aware collision avoidance system that has been proposed in chapter 4.

5.5.3 Integrity of the shapes objects in AR augmented scenes

Since the proposed augmentation process introduces new data in the point cloud, it must be verified that the new data looks realistic. So several experiments have been realized to observe that virtually introduced elements do look as they should and do not present any noticeable distortion.

A first experiment considers a real pedestrian and a virtual pedestrian crossing a street. Several construction cones, some real, some virtual, are placed around them. The scene is captured and augmented on the [experimental automated car](#), the resulting point cloud is displayed in fig.44. The virtual elements, the cones and the pedestrian appear very similar to their real counterpart in this augmented point cloud. They present the same density of points. The irregular spacing of the actual sensor's layers is visible on the sensor's real observations but also on the augmented data. Also the virtual objects correctly produce shadows and occlude real elements behind them. All of this make that virtual elements that were introduced can almost not be distinguished from the real ones. Apart of the occlusions, the perception of the real elements does not appear to be altered by the augmentation. In conclusion, it appears qualitatively that, for this example, the augmentation produces realistic scenes.

More examples are presented in a video that is available in the media section at this link <https://ieeexplore.ieee.org/document/9827351/media#media>. However, in these early results, for sake of a real time simulation, the resolution of the virtual sensor had to be reduced. Because of this, virtual objects tend to appear like squares at longer range. Still this effect is limited and it has only been observed for objects farther than 10m. After optimizations of the simulator, it became possible to simulate the virtual sensor at the same resolution than the actual sensor. So this "square" effect is not visible on recent results. Despite, this imprecision that affected the early results, the propose LiDAR point cloud augmentation process visibly preserves the integrity of the shapes of the virtual objects.

The fact that the augmented data looks realistic is an important qualitative result but the most important is that it is realistic to the software under test. To certify the [representability](#) of the [AR](#) test framework, it must be verified that the data augmentation is transparent to the

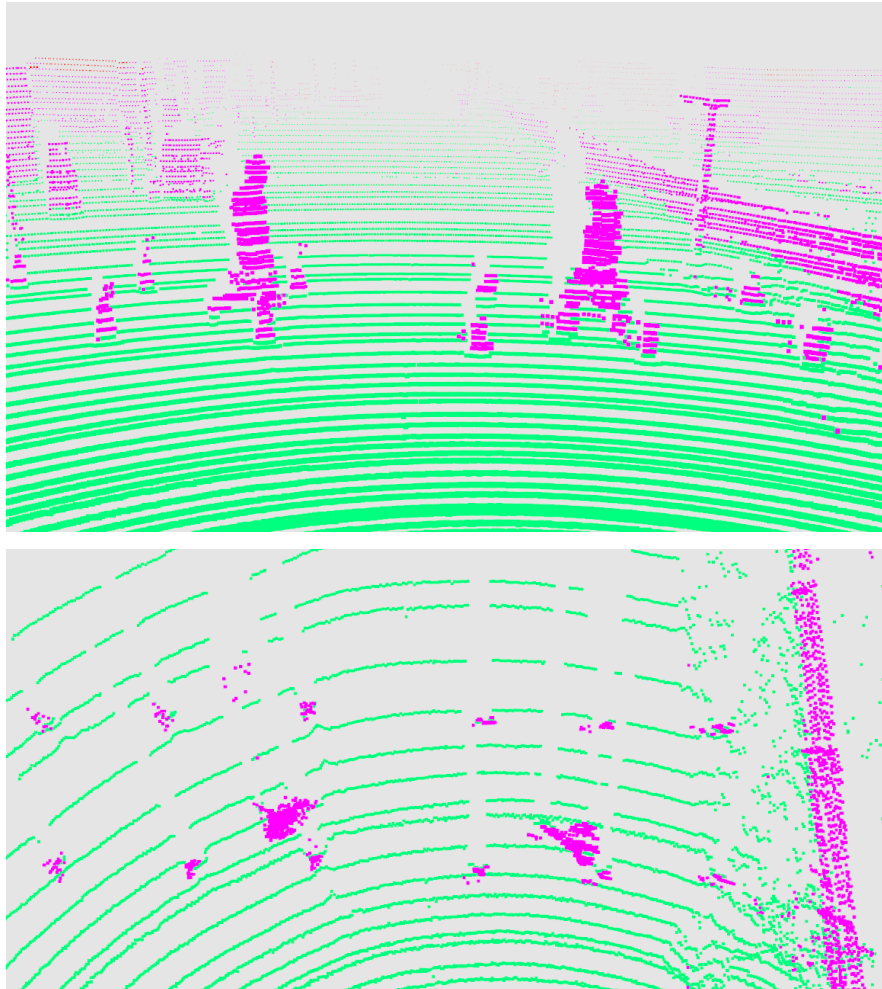


Figure 44 – Example of a point cloud that has been obtained by AR with the proposed method. The same point cloud is seen from the vehicle perspective and from above. The scene consists of an actual pedestrian and 6 actual construction cones on the left plus a virtual pedestrian and the same set of cones on the right. Pedestrians are approximately 5m away from the vehicle. After the augmentation, the [Ground Elevation and Occupancy Grid Estimator \[RPNL17\] \(GEO\) \[RPNL17\]](#) has been used to classify the points that belong to the ground. So the points are displayed in green for those of the ground and magenta for the other points.

software under test. In this purpose, the collision avoidance system that is presented in chapter 3 constitutes the software under test. In this software, the perception is achieved by the [GEO](#) and the [Conditionnal Monte-Carlo Dense Occupancy Tracker \[RNL15\] \(CMCDOT\)](#) that, respectively, segment out points that belong to the ground surface and filter and track the occupancy to produce a [Dynamic Occupancy Grid](#). In this context, several experiments with various AR scenes have been done to verify that the data augmentation is seamless in the [Dynamic Occupancy Grid](#), after these 2 steps of perception.

Fig.45 provides an example of this verification. This experiment displays an augmented scene with static real elements, a car, a pedestrian and some construction cones. The virtual counterpart of each of these elements is placed in a mirror position. After the augmentation and the perception, the virtual and real elements can not be distinguished on the occupancy

grid. This proves that, in their representation in the augmented point cloud, virtual elements do not have any distortion that affects the perception system. As the introduction of virtual elements in the sensor messages is seamless in the perception output, it will also be seamless to the navigation system that uses this perception. Therefore, regarding the shapes of objects, the data augmentation appears to be transparent to the software under test.

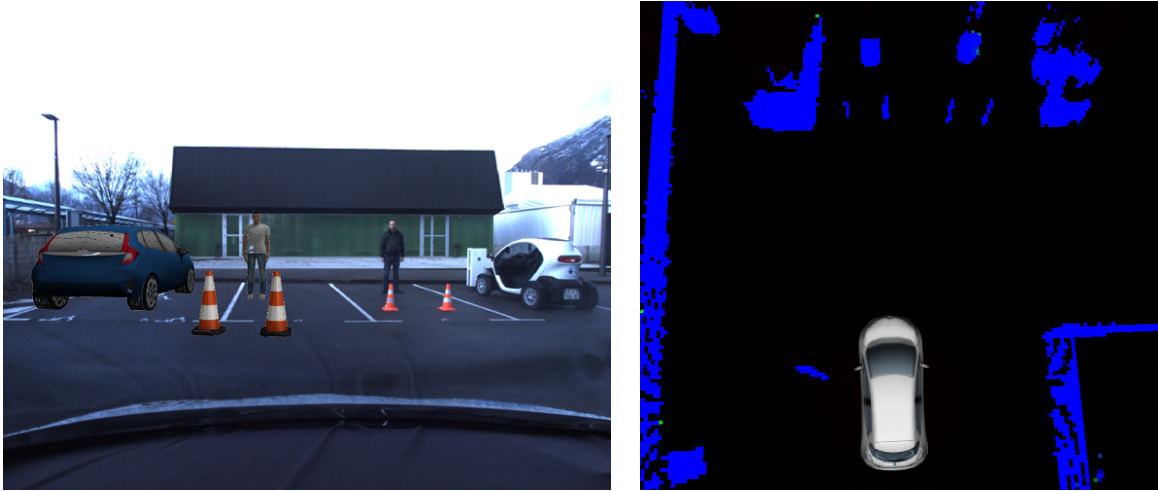


Figure 45 – In this experiment, 4 virtual elements, a car, a standing person and 2 construction cones are added via AR to a scene that contains in mirror the real counterparts of these elements. The visualization of the AR scene is shown on the left and the corresponding occupancy grid on the right. The free space is displayed in black and the occupied space in blue. Due to an imprecise calibration of the camera, the position of the objects in the visualization is not accurate. Yet, in the occupancy grid, all of the present virtual elements do appear similar to their real counterpart.

5.5.4 Relative position and speed consistency of virtual objects

Although the preservation of the shapes of virtual elements is crucial, it is of an equal importance to make sure that these elements are introduced in the right position and have a consistent motion over time. The positioning of virtual objects in the augmentation process depends on the synchronization of the virtual environment to the real world which itself largely relies on the localization of the actual vehicle. Due to the eventuality of inaccuracy of the localization system, there is a significant risk that virtual objects can be misplaced and have inconsistent motion. To verify that this is not the case, the perceived motion of virtual objects in AR data has been investigated with several experiments. Again, the purpose is to prove that, under this criterion of motion consistency, the Augmented Reality is seamless to the software under test. So, a first set of experiments investigates the predicted occupancy grid, one of the outputs of the proposed **PCD** collision avoidance framework. The predicted occupancy results from occupancy filtering and tracking and speed estimation within the **CMCDOT** and probabilistic kinodynamic motion prediction within the **PCD**. So, for it to be correct, the predicted occupancy requires the observation of a consistent motion over time.

Fig. 46 displays a predicted occupancy grid that has been computed by the kinodynamic

occupancy prediction of the [PCD](#). The scene contains a virtual and an actual pedestrian walking side by side at the same speed. Several construction cones are placed around the pedestrians, half of them are virtual. In this predicted occupancy grid, the virtual pedestrian can not be distinguished from the real one, neither by its shape neither by its predicted motion. They appear to walk at the same speed and cover the same area of the grid. This suggests that the shape and the motion of virtual human seen as a cluster of cells of the dynamic occupancy grid are consistent and representative of a real human.

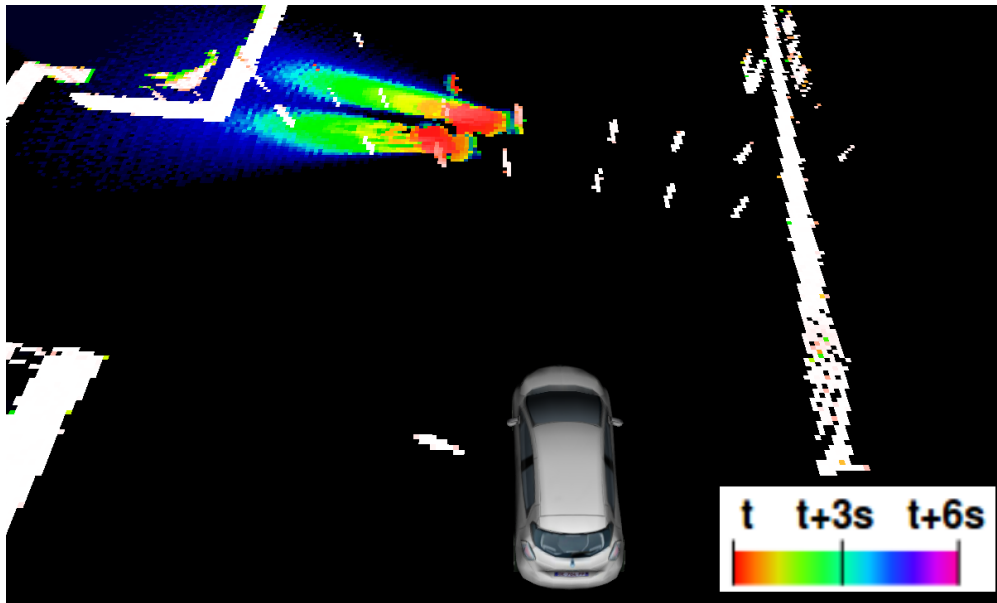


Figure 46 – This image displays a predicted occupancy grid that corresponds to a scene containing a virtual and an actual pedestrian walking side by side at the same speed. The color of each cell of the grid denotes at which time an occupancy is predicted in it. Cells with static detected occupancy are white and empty cells remain black. In the grid, the virtual pedestrian can not be distinguished from the real one, neither by its shape neither by its predicted motion.

Various other experiments have shown that the use Augmented Reality on LiDAR data produces the same predicted occupancy grid that what could have been obtained in real-world scenes. The predicted occupancy grid is highly relevant to whole collision avoidance framework since this grid is the very last Cartesian representation of obstacles and their motion before they are interpreted as collision risks in the [configuration](#) space. Therefore, as the motion of objects that have been introduced via AR appears to be seamless in predicted occupancy grids, it suggests that this new testing approach is seamless for the whole [PCD](#)-based collision avoidance.

To go further on the validation of the consistency of the motion of virtual objects in augmented data, this thesis proposes also a quantitative survey. This survey considers another output of the [PCD](#), the estimated [Time To Collision \(TTC\)](#). For this survey, only the estimated [TTC](#) on the current trajectory (constant steering and null acceleration) is considered. The [TTC](#) estimation is very likely to be severely affected if the virtual targets have inconsistent motion. Therefore the accuracy of the [TTC](#) is assumed as an indicator of the

representability of the motion consistency. The **TTC** is also relevant for its crucial role in the collision avoidance system since it is the final value of risk estimation that the **PCD** produces. This value is then directly used in the sampling based **Model Predictive Control (MPC)** planner. So it can be considered as an indicator of representability for this specific software under test that is studied in this thesis. The experiments consider a simple scenario. A pedestrian is either standing in the middle of a street either walking across it. The vehicle under test is manually driven at an almost constant speed between 10 and 25km/h along the street until it collides with the pedestrian. For half of the experiments, the pedestrian is figured by a dummy which is hanging on a Tyrolean traverse above the street. The other half of the experiments consider a pedestrian which is introduced in AR thanks to the proposed system. 30 such experiments are executed. Fig.47 depicts the test scenario.

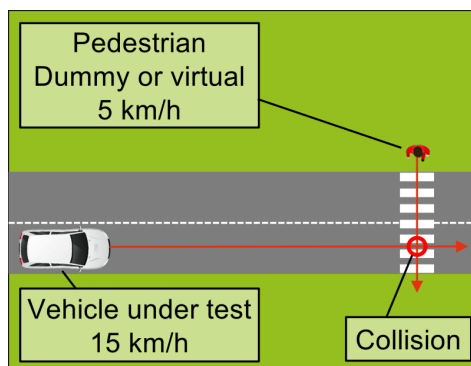


Figure 47 – Schematic representation of the test scenario for the quantitative validation of the consistency of the motion of obstacles that have been introduced via AR on LiDAR data.

Fig.48 displays the graphs of the **TTC** predictions for the last seconds before the collisions with walking targets, real and virtual. In all of the experiments, the estimated **TTC** globally decreases from the prediction horizon which had been set to 5.5s and at a rate of about -1. This is the expected result since the **TTC** is meant to predict the moment of collision. As the vehicle has been manually driven to maintain an almost constant speed, it is logical that this prediction is accurate. What is important is that this behavior is observed with both real and virtual targets. There are some experimental variations that make the result slightly differ from this behavior but they appear to affect similarly both experiments with real and virtual targets. Therefore these variations are not caused by the LiDAR data augmentation. The curves also reveal that the estimated **TTC** with real targets presents some fast variations of small amplitude that are not observed with virtual targets. This could be due to the nature of the motion of these targets. On the one hand, the virtual pedestrian moves at an exactly constant speed, this is not realistic. On the other hand, the dummy on the Tyrolean traverse has an irregular motion which is not realistic neither. In spite of these experimental variations, the graph reveal that the **PCD** is able to anticipate the collision 5s before it occurs and the moment of collision is accurately predicted 4s before the collision. This result is observed with both real and virtual targets.

In the last seconds before the collision, the **TTC** estimation is expected to be more accu-

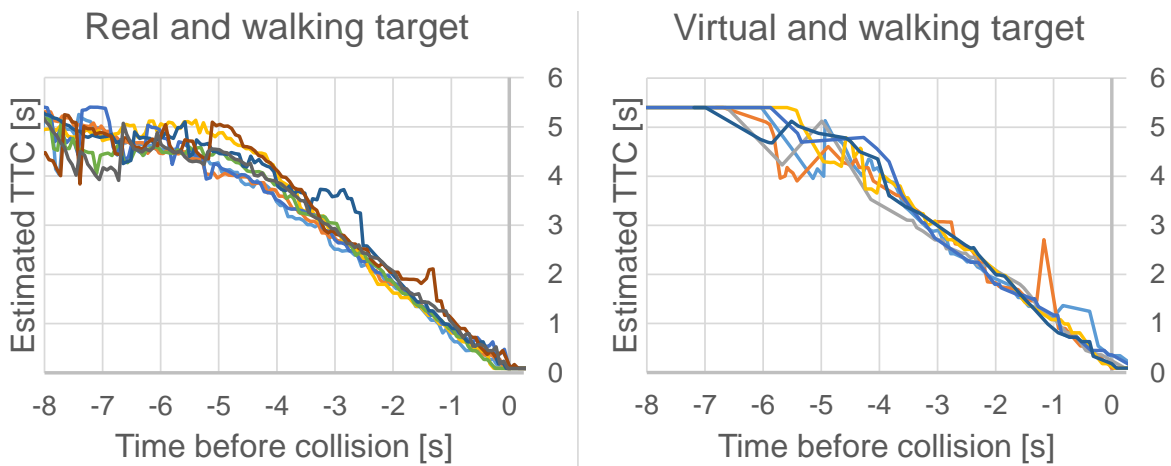


Figure 48 – Estimation of *Time To Collision (TTC)* with the *PCD* during the few seconds before collision over several experiments. The real target is a dummy hanging on a Tyrolean traverse in order to mimic a pedestrian crossing the street. The virtual target is a pedestrian introduced through AR.

rate. Therefore a closer investigation on these last seconds could present less experimental variations and reveal potential differences of the experiments with real and virtual targets. For this, fig.49 shows the statistic distribution of TTC predictions over series of experiments, at different instants before collision. It reveals that the TTC estimation is slightly pessimistic. When the car is 3s before the collision, the TTC estimation is in average 2.7s with a walking target and 2.85s for a static target. At 2s before collision, the TTC is estimated in average between 1.85s and 1.95s.

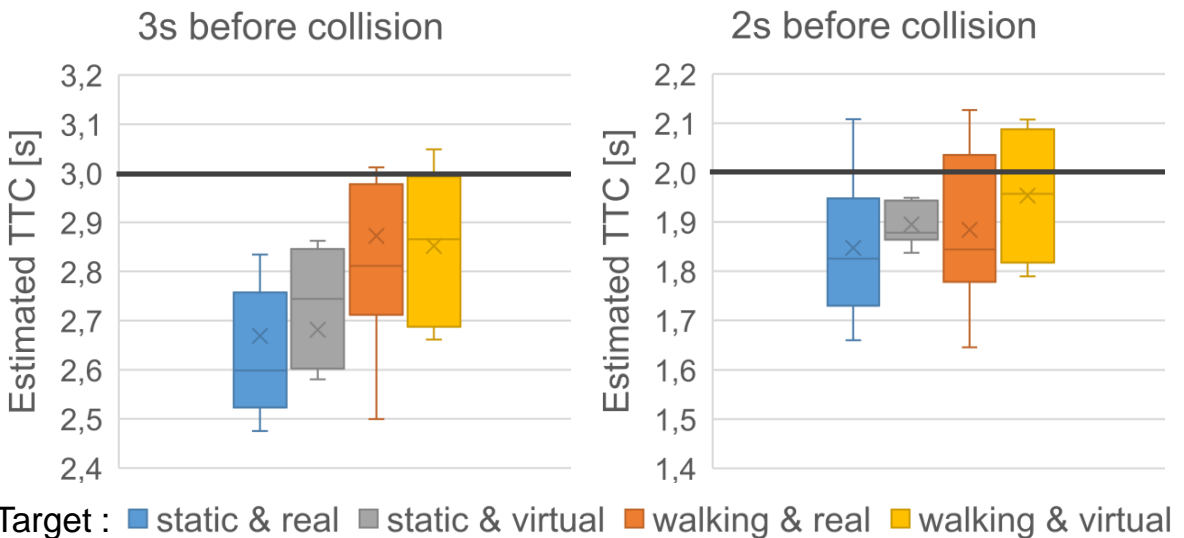


Figure 49 – Distribution of the estimated time to collision, at 3s and 2s before collision. The average, the median, the quartiles and the extrema are displayed for each series.

The fact that the *PCD* provides such a slightly pessimistic estimation is good because it indicates a conservative behavior. But the most important is that this phenomenon and the average values remain very similar when a virtual target is used. In some cases, a slight

difference can be observed between some results with a virtual target and the corresponding results with a real target. However such differences appear negligible compared to the variations of the result that occur within a test series, when the test parameters are unchanged. In all cases, the disparities among a series of test with the real target are much wider than the difference with series of tests using the virtual counterpart. It proves that random experimental variations have more effect on the TTC predictions than the use of AR. Moreover, in most cases, the amplitude of these experimental variations is the same with a real and with a virtual target. This suggests that these variations are not due to target itself neither to the sensing process but might be caused by the system under test i.e. the vehicle or its software. So the use of AR to introduce virtual targets does not have a significant impact on the accuracy of TTC prediction with PCD. Considering that the accuracy of the TTC estimation has been identified as an indicator of the consistency of the motion of virtual objects, these experiments prove the representability of the motion of AR introduced objects.

So across these experiments, the representability of the proposed Augmented Reality test framework has been challenged and it has been assessed with various outputs of the PCD-based collision avoidance which constitutes the software under test. First, it has been observed that this new test framework does not introduce any significant delay to the perception. It also preserves the shapes of the virtual objects that it introduces in LiDAR data and these objects appear similar to real ones in the perception output. Then the motion of these virtual object is also consistent enough such that they also look similar to their real counterparts in the occupancy prediction output. Finally it has been verified that the prediction of the collision risk is accurate and not disturbed by the use of virtual targets. This last result also implies that virtual targets have a consistent motion over time. Altogether this demonstrates that the LiDAR data is seamless for the proposed collision avoidance system. It proves the representability of the proposed test framework and that it is suitable for the evaluation of the PCD-based collision avoidance that is proposed in chapters 3 and 4.

5.6 Applications of the proposed Augmented Reality Framework

5.6.1 New opportunities for automotive software testing

The representability of the proposed Augmented Reality (AR) test framework has been proven in the previous section. Therefore this method is suitable for testing and evaluation of automotive software. This new testing approach is significantly different from the other test environments. It is more realistic than simulation and ViL. At the same time, it is more flexible than real world testing. This originality creates new opportunities for automotive software testing.

First, Augmented Reality facilitates real world testing. Using mixed reality testing scenes provides safer test conditions because it allows to replace collision targets with virtual elements. Also, it reduces testing time and costs since virtual elements can be added instantly and for free. Finally it makes the test more configurable and repeatable with virtual agents

that can be programmed. Certainly, the more the scene contains virtual elements, the more it loses its realism. But, unlike ViL, the proposed AR framework lets the user control how many virtual elements should be introduced in the testing. So, for more realism, the virtual part of the environment may be restricted to non critical elements of the testing scenario.

More than simply facilitating tests, the proposed AR framework opens new possibilities for testing. It enables to test vehicles in critical scenarios where collisions could not be avoided. These scenarios are too dangerous to be tested in real world but are needed to challenge software. Also Augmented Reality allows to go beyond the limitations of actuated fake targets and dummies. Such physical fake targets are too costly to be deployed in large amounts and their actuation is limited. In the meanwhile, virtual agents can be introduced in very large amounts in AR and can be freely programmed for advanced motion and even interactive behaviors. Finally, Augmented Reality enables to directly compare virtual elements and their real counterpart within the same environment. This can be used to evaluate the representativity of simulation components.

Among these various new opportunities that Augmented Reality offers, the proposed AR framework has been actually used for testing some automotive software within the present thesis. This occurred for 2 different applications that are presented in the next sections.

5.6.2 Application to test collision mitigation

In [SGSM⁺22], the authors have proposed for road vehicles a motion planner to mitigate the risk of injuries in scenarios where a collision is unavoidable. This software is meant to operate in critical scenarios that simultaneously present high probabilities of dangerous collisions with multiple agents and obstacles. In such a scenario, the purpose of this mitigation planner is to plan and execute the least dangerous trajectory. Therefore testing this software implies to have the ego-vehicle facing high collision risks with multiple various road users. For safety reasons, such a test is not acceptable in real world but it becomes possible with [Augmented Reality \(AR\)](#).

The proposed AR framework has been used on the [experimental automated car](#) to contribute to the experimental validation of this collision mitigation planner. Thanks to this, the authors have evaluated the response of their software in different scenarios that involved several cars and pedestrians. The planner has controlled the vehicle and drove it according to the computed mitigation trajectories. Fig.50 presents the visualization from the driver's perspective, the occupancy grid and the mitigation trajectory during such test. In this example, the ego-vehicle is assumed to be going too fast to brake while it faces 2 arriving cars and a pedestrian. In order to mitigate the overall risk of injuries the planner drives the ego to a collision with a car which is parked on the right side. Augmented Reality made this test possible on an real vehicle. This helped the authors of this software to debug and improve their perception filter against realistic sensor data and their motion control against the actual physics and actuators of the vehicle. Most of all, this Augmented Reality testing has demonstrated that the method can applied in real conditions.

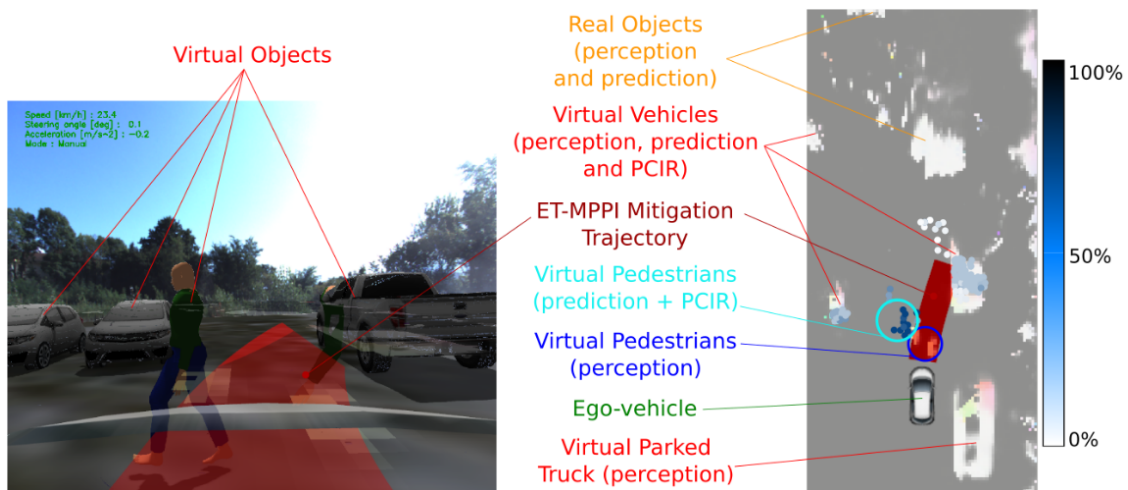


Figure 50 – Illustration of a test of ET-MPPI [SGSM⁺22] collision mitigation planner in an augmented reality scene. On the left is a visualization of the augmented scene. On the right is the corresponding Dynamic Occupancy Grid with the estimated probability of collision with injury risk and the planned mitigation trajectory.

5.6.3 Application to test navigation in crowded environments

The evaluation of the [Predictive Collision Detector \(PCD\)](#) and its application to collision avoidance, that have been presented in chapters 3 and 4, have also required the use of [Augmented Reality \(AR\)](#). This software is meant to drive a car in diverse complex, unstructured and potentially crowded urban scenarios. Therefore its evaluation involved tests scenarios that present potential collision risks with pedestrians, cyclists and other cars. So [AR](#) has been used to ensure safety during the testing. But [AR](#) also permitted to conceive rich tests scenarios with numerous agents. Also agents have been programmed with rich motion patterns and advanced interactive behaviors. This could not have been done with actuated dummies and physical fake targets. And even, such large numbers and such a wide diversity of actors would have been extremely difficult to achieve in real world testing. [AR](#) enabled to test on the real vehicle many scenarios that otherwise could have only been feasible in simulation, such as the scenarios of navigation in crowds with the human-aware [PCD](#).

Fig.51 displays the vehicle during a test of navigation in a dense crowd in Augmented reality. The crowd consists of 340 people. The figure shows the augmented point cloud, the predicted occupancy grid and visualizations from different points of view. In spite of this highly populated scene, the whole [AR](#) system did generate the augmented point cloud in real time, according to the sensing frequency and with a minor usage of the computing resources. It is visible on the point cloud that the many people in this crowd occlude each other. Then it is a real challenge for the perception to filter it, identify each agent and track its motion. These occlusions are correctly rendered by the proposed sensor level [AR](#) approach but would have been ignored by object-based mixed reality approaches.

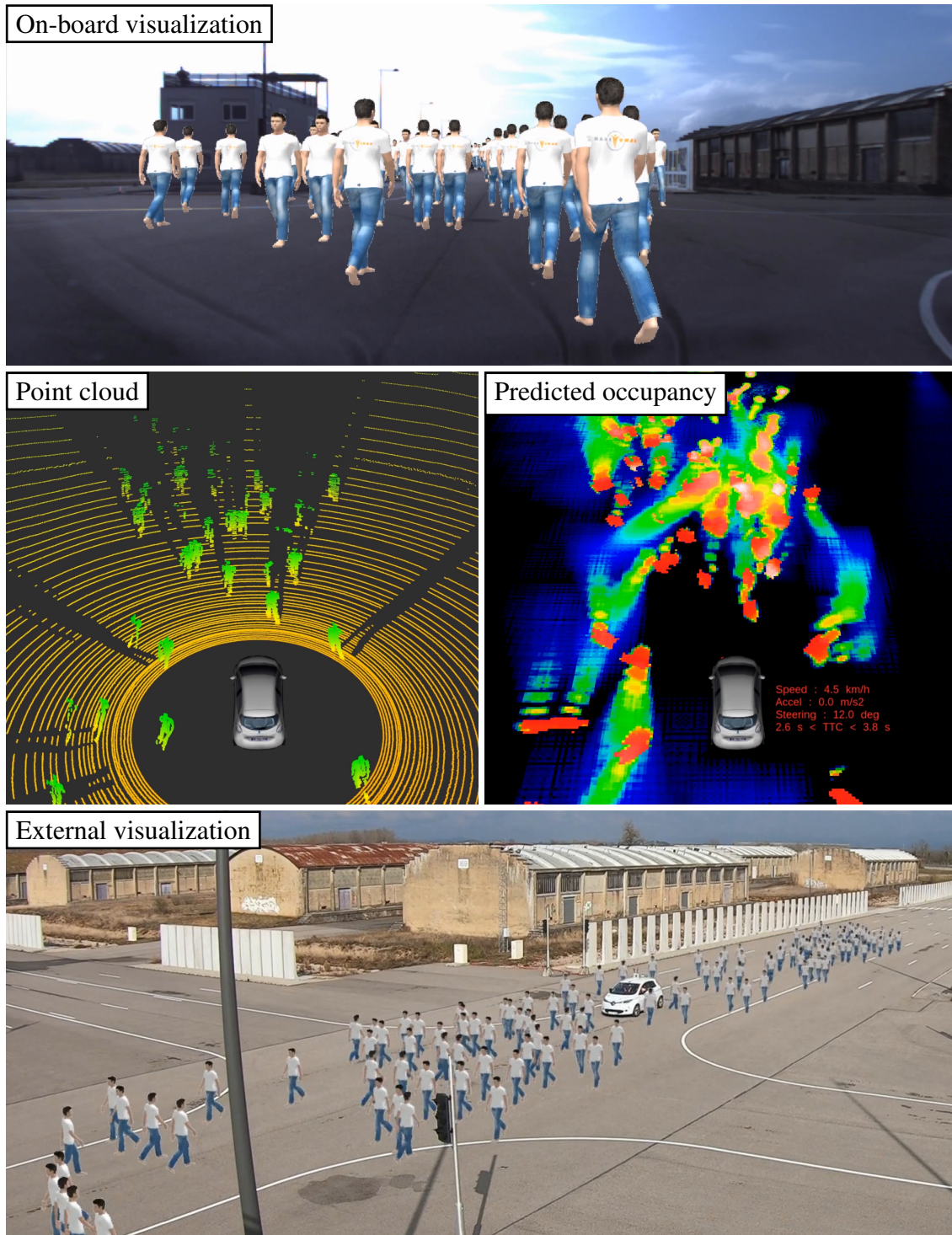


Figure 51 – Illustration of a test of navigation in a dense crowd with the human-aware PCD in an augmented reality scene. The augmented point cloud and the on-board visualization have been generated in real time by the proposed AR framework during this experiment. The predicted occupancy has been computed in real time by the human-aware PCD after it has been fed with the augmented point cloud. The external visualization has been reconstructed offline. These displays correspond to the same moment of the same test.

In this test, the planner drove the real car through this crowd and the pedestrians reacted to it. A video of this experiment is available at <https://inria.hal.science/hal-04073269>. This test with an actual automated car driving through such a large dense crowd is unprecedented in the related works. Overall, the use of **AR** for the validation of the **PCD** allowed to involve the actual vehicle in a wide diversity of test scenarios. It contributed to evaluate and demonstrate that the proposed collision avoidance system could be transferred to real world applications.

5.7 Chapter conclusion

This chapter describes a new framework for **Augmented Reality (AR)** on sensor data along with its key component, the LiDAR data augmentation function. This **AR** system is proposed as a test environment for the validation of complex automotive software such as **Automated Driving Systems (ADS)** and **Advanced Driver Assistance Systems (ADAS)**. This thesis has studied the application of **Dynamic Occupancy Grids** to collision avoidance and proposed a new collision avoidance system which includes an advanced perception system and relies on an original representation of the obstacles without object-based representations. Simulation was not realistic enough to suffice for the validation of this collision avoidance system. Due to safety, economical and time constraints, real-world testing was also too limited. **Augmented Reality (AR)** appeared as complementary test environment that could be more realistic than simulation and produce richer and more diverse tests conditions than real-world testing.

However most of recent research in this field propose object-level **AR** approaches that bypass the perception and produce object-based representations of the scene that are meaningless to the collision avoidance system of this thesis. Few other research works present solutions for **AR** on sensor data. But none of the works that have been reviewed in the proposed state of the art considered the real-time augmentation of LiDAR point clouds such as the ones that constitute the input of the proposed collision avoidance system. Hence this thesis proposes a new framework for **Augmented Reality (AR)** testing and the required real-time LiDAR data augmentation function.

The proposed framework differs from the state of the art since it relies on the structure of a **Vehicle-in-the-Loop (ViL)** system such that it enables a smooth transition of the tests from **ViL** to **AR** and from **AR** to real-world. It is a promising test environment that bridges the gap between simulation and reality and provides new test opportunities including dense, safe, cheap and reconfigurable test environments but also numerous test actors with rich behaviors that can be safely collided such as crowds of virtual pedestrians. The LiDAR point cloud augmentation relies on a massive parallelization of individually simple computations on a **Global Purpose computing on Graphics Processing Unit (GP-GPU)** architecture. This enables a real time a frugal execution. This augmentation accurately models all occlusions between virtual and real elements of the scene. It produces an augmented point cloud that respects the angular density and the layer structure of the actual sensor. It also preserves all

the real sensor observations as long as they are not occluded.

The proposed [AR](#) test framework is deployed on the [experimental automated car](#) and applied to each of its 4 LiDAR sensors. It is experimentally verified that the data augmentation does not add any significant delay to the perception, that the point cloud augmentation preserves the shape of the introduced virtual objects and that the motion of these objects is consistent. An analysis of several outputs of the perception and risk estimation of the collision avoidance system demonstrate that the use of [AR](#) is seamless to this software. This all proves the representability of the proposed Augmented Reality test framework i.e. its capability to produce realistic test conditions for the collision avoidance system that is studied in this thesis.

Based on these results, this new testing method has been used to test the [Predictive Collision Detector \(PCD\)](#) collision avoidance framework and its extension for human-aware collision avoidance. Several such tests are presented and discussed in chapters 3 and 4. This testing method has also been used to evaluate a collision mitigation driver [[SGSM⁺22](#)]. More recently, the proposed testing approach and its implementation on the experimental vehicle are currently studied and experimented in the project [PRISSMA](#) which considers to associate [AR](#) with contextual automated generation of critical test scenarios [[HLM⁺22](#)] and new evaluation metrics for perception in order to constitute a complete validation framework.

This contribution to Augmented Reality on sensor data calls for future developments. First, there is a need for real-time and realistic augmentation of sensing modalities other than LiDAR. Some of the mentioned related works do consider the augmentation of camera images. Still this augmentation is not realistic enough to be considered as seamless and be trusted for validation purposes. There are also several other common sensing modalities, such as radar, for which there is no known solution for data augmentation. Besides, Augmented Reality on sensor data is a promising testing tool that provides new testing opportunities. Yet it does not suffice on its own for validation of automotive software. It is a complementary tool that must be used along with simulation and real world testing. Therefore, to get the best out Augmented Reality testing, it is necessary to design new validation strategies that combine tests in simulation, in Augmented Reality and in reality. In future validation approaches, Augmented Reality could be seen as an opportunity to facilitate tests in the real world and also as a solution to collate and compare simulation and reality in order to certify the realism of a simulation.

Chapter 6. Conclusion and future works

6.1 Global conclusion

In this thesis, it has first been noticed that, in spite of the major breakthrough of [Automated Driving Systems \(ADS\)](#), they are not yet ready for navigation in urban shared spaces. This is due to the complexity of these crowded and unstructured environments, where perception, collision risk detection and planning are highly challenging.

To address navigation in such conditions, this thesis suggested to rely on [Dynamic Occupancy Grid](#) perception that accurately models complex and dynamic environment, even under uncertainty and partial observability. As it has observed that the current state of the art of local planning does not exploit all the potential of [Dynamic Occupancy Grids](#), this thesis has investigated a local planning solution that would be tailored to operate from this specific perception format and to make the most of it. The choice of this original perception format has driven the thesis towards innovative approaches. This has motivated to permanently consider uncertainty all along the collision risk evaluation. It has induced approaches without object-based representations of the [agents](#) and of the environment. It has also inspired to build parallelizable algorithms. Above all, it has encouraged to consider jointly perception and planning throughout their design, development, deployment and evaluation.

As a result, the first contribution of this thesis is the [Predictive Collision Detector \(PCD\)](#) module that combines particle-based occupancy prediction and stochastic grid-based collision detection in order to exploit [Dynamic Occupancy Grids](#) with sampling-based local planning approaches. The [PCD](#) has been proposed within a complete collision avoidance framework that has been evaluated on several experimental platforms. This system has proven that it is capable of safe navigation, even in densely crowded and unstructured environments. It also appeared to be robust to uncertain and partial perception. But mostly, it revealed its versatility as it could address various very different driving tasks on different vehicles.

The second main contribution of this thesis is the extension of the [PCD](#) for human-aware collision avoidance. Thanks to the interpretation of semantic classification in the [PCD](#) and with an additional interaction-aware human motion prediction model, the new proposed collision avoidance solution demonstrated that it can safely and efficiently navigate a car in presence of humans, even in extremely dense crowds. While proactive planners are difficult to deploy in real-world conditions due to their inherent complexity and predictive planners are not efficient in crowded environments, the proposed solution stands as a compromise that is capable of efficient human-aware navigation and could be transferred to real-world applications.

The third and last main contribution of this thesis is a new test framework that uses [Augmented Reality \(AR\)](#) on LiDAR sensor data to challenge advanced automotive software against test scenes that combine real world and virtual elements in a realistic manner. The proposed approach relies on a real time augmentation of LiDAR point clouds that accurately

models occlusions, preserves real sensor data and maintains the format and the structure of the initial point cloud. This method has been deployed on the experimental car. It has been demonstrated that the proposed **AR** framework produces realistic test conditions and that the sensor data augmentation is seamless for the **PCD** based collision avoidance. Hence the new proposed test framework contributed to the evaluation of the **PCD**. The proposed **AR** test framework offers a smooth transition from simulation to **AR** and from **AR** to real-world. Bridging the gap between simulation and reality, it offers new opportunities for test and validation of complex automotive software.

All along this thesis, a particular care has been taken to consider and evaluate real-world applications of the proposed contributions. Some of the contributions have been deployed in real-world on several experimental platforms and all of the contributions have been applied in real-world to the **experimental automated car**. Whenever the real-world evaluation of the methods was not possible, realistic **Software-in-the-Loop (SiL)** simulations and **AR** experiments have been used.

In the end, the contributions of this thesis offer a new application of **Dynamic Occupancy Grids** to local planning. The proposed solution for collision avoidance has demonstrated its versatility, its robustness to complex environments and its capability to generate advanced driving behaviors such as risk anticipation, safety distance management and human-aware navigation. This approach without object-based representations of **agents** constitutes a new paradigm for collision avoidance that conveys new opportunities and limitations that differ from those of the related works. The proposed collision avoidance solution could contribute to leverage the current limitations of Autonomous Driving Systems.

This thesis has placed a focus on automotive applications. All of the contributions have been considered in an automotive application because it constitutes at the same time a relevant use case and a challenging application. Yet the results of this thesis can be generalized to mobile robotics.

Ultimately, with the proposition of the perception-oriented **PCD** collision avoidance system and with the development of a new **AR** framework that includes perception in the test, this thesis promotes the consideration of realistic perception constraints in the design of planning solutions.

6.2 Contributions

6.2.1 List of publications

The present thesis has motivated the following scientific publications :

- Thomas Genevois, Anne Spalanzani, and Christian Laugier. Interaction-aware predictive collision detector for human-aware collision avoidance. In *IEEE Intelligent Vehicles Symposium (IV)*, 2023
- Thomas Genevois, Lukas Rummelhard, Anne Spalanzani, and Christian Laugier. From probabilistic occupancy grids to versatile collision avoidance using predictive colli-

- sion detection. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023
- Thomas Genevois, Jean-Baptiste Horel, Alessandro Renzaglia, and Christian Laugier. Augmented reality on lidar data: Going beyond vehicle-in-the-loop for automotive software validation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2022
 - Luiz Alberto Serafim Guardini, Anne Spalanzani, Philippe Martinet, Christian Laugier, Thomas Genevois, and Anh-Lam Do. Minimal injury risk motion planning using active mitigation and sampling model predictive control. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2022
 - Andrés Gómez, Thomas Genevois, Jerome Lussereau, and Christian Laugier. Dynamic and static object detection considering fusion regions and point-wise features, 2021
 - Jean Quilbeuf, Mathieu Barbier, Lukas Rummelhard, Christian Laugier, Axel Legay, Blanche Baudouin, Thomas Genevois, Javier Ibañez-Guzmán, and Olivier Simonin. Statistical Model Checking Applied on Perception and Decision-making Systems for Autonomous Driving. In *PPNIV 2018 - 10th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pages 1–8, Madrid, Spain, October 2018

6.2.2 Scientific diffusion

Besides written publications, some contributions of this thesis have been described in the following scientific presentations :

- Thomas Genevois, Anne Spalanzani, and Christian Laugier. Interaction-aware motion planning as an alternative to proactive planning for human-aware navigation. *IEEE IV Workshop on Interaction-driven Behavior Prediction and Planning for Autonomous Vehicles*, 2023. Invited talk
- Lukas Rummelhard, Thomas Genevois, Jean-Alix David, Amaury Nègre, and Christian Laugier. Real-time Ground Estimation and Point Cloud Segmentation. *GTC - GPU Technology Conference*, March 2018. Poster, available at <https://hal.archives-ouvertes.fr/hal-01903668>

The software that has been developed within this thesis has been regularly involved in live demonstrations for academic and industrial partners. It has also been showcased in the following major events :

- at the *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* for a live demonstration of an early version of the PCD based collision avoidance. A video of this is available at <https://hal.inria.fr/medihal-01963296>
- at the *2022 World Economics Forum (WEF)* for a live demonstration of Augmented Reality on LiDAR data.

6.2.3 Transfer projects

The contributions of the present thesis have been involved in the following projects.

IRT NANOEELEC – SVA This project "Sécurité du Véhicule Autonome (SVA)" (safety of the autonomous vehicle) is part of the "Institut de Recherche Technologique NANOEELEC" which is funded by the French program "Investissement d'Avenir", ANR-10-AIRT-05. All of the research works and technical contributions of this thesis have been realized and funded within this project.

This project is carried out at Inria Grenoble-Alpes. It intends to develop, embed and promote perception software modules for the safety of autonomous vehicles. For this, the project relies on state-of-art techniques for Bayesian fusion and filtering from heterogeneous sensor data, these techniques guarantee the safety of vehicles but also provide a rich perception of the environment which is useful for autonomous navigation itself. The [experimental automated car](#) has been acquired, maintained and developed within this project. This prototype serves for software development and testing but also as a demonstrator to present technical results of the project. This long term project enabled the progressive development of a software suite for safe navigation of autonomous vehicles. The [Conditionnal Monte-Carlo Dense Occupancy Tracker \[RNL15\] \(CMCDOT\)](#) and the [Ground Elevation and Occupancy Grid Estimator \[RPNL17\] \(GEOG\)](#) have been developed within the context. More recently, the [PCD](#) and the framework for [AR](#) on sensor data, that are contributions of the present thesis, have also been realized within this project. While this project supports this continuous research, it also intends to initiate partnerships with industrial actors and transfer the technologies for industrial applications. For this, the experimental platform and the contributed software are regularly involved in live demonstrations for academic and industrial partners. Some of these demonstrations were part of major events such as the 2018 International Conference on Intelligent Robots and Systems (IROS) and the 2022 World Economic Forum (WEF). The research that has been realized within this project, including some of the contributions of the present thesis, has initiated several other projects with industrial partners.

As an engineer at Inria for 6 years, I have regularly contributed to this project. First, I have evaluated and fine tuned the drive-by-wire module of the experimental car. Then I have developed the localization system of this vehicle that fuses odometry, inertial measurement and GPS. I have also realized a virtual twin of the vehicle and virtual test environments for [SiL](#) simulation. Within this project, I have developed the [PCD](#). I deployed it on the experimental car for an application to autonomous navigation and for an application to driving assistance. I also conceived and experimented a global planner that operates from Bayesian Occupancy Grids and respects the kinematic constraints of the vehicle even in complex maneuvers. Finally, I developed the [AR](#) framework. I deployed it and tested it on the vehicle. I took part in the communication and live demonstrations of these results.

IRT Nanoelec – Light Vehicle Demonstrator This project is part of the "Institut de Recherche Technologique NANOEELEC" which is funded by the French program "Investissement d'Avenir", ANR-10-AIRT-05. This project supported the deployment of the proposed [PCD](#) on the [urban light vehicle prototype](#).

This project emerged from the project [IRT NANOEELEC – SVA](#). Its purpose is the devel-

opment of the prototype of a new type of vehicle for urban micro-mobility. This [urban light vehicle prototype](#) has been built by an industrial partner. The partner provided the vehicle and its low level automation. The localization, the perception and the navigation were to develop by Inria within this project. For this, the software that had been developed on the [experimental automated car](#) has been adapted and deployed on this new vehicle. In this context, the proposed [PCD](#) has been deployed on this [urban light vehicle prototype](#).

Being the only engineer from Inria to work on this project, I was in charge of the software of the vehicle. I have adapted the modules for localization, perception, local planning and global planning, that had been developed on the [experimental automated car](#), for their deployment on this new vehicle. I also built a virtual environment and a virtual twin of the vehicle for tests in [SiL](#) simulation. I have also tested in real-world the vehicle and its autonomous navigation capabilities. I realized a live demonstration of this.

R&D Booster MoovIT This project is part of the program "R&D Booster", it has been funded by the French region "Auvergne-Rhône-Alpes". This project supported the deployment of the proposed [PCD](#) on the [industrial logistics robot](#).

This projects intends to contribute to the flexibility of industrial production via a new solution for logistics that relies on static elements, mobile robots and a smart interface system. The industrial partner leading this project was in charge of the realization of a new mobile robot that would be tailored for this application. Inria had to develop the autonomous navigation of this vehicle. For this, the software that had been developed on the [experimental automated car](#) has been deployed on this new vehicle. In this context, the proposed [PCD](#) has been deployed on this [industrial logistics robot](#).

We were 2 engineers from Inria to contribute to this project. I have created a virtual environment and a virtual twin of the vehicle for tests in [SiL](#) simulation. I have also participated in the adaptation of the modules for perception, local planning and global planning, that had been developed on the [experimental automated car](#), for their deployment on this new vehicle.

PRISSMA The project "Plateforme de Recherche et d'Investissement pour la Sûreté et la Sécurité de la Mobilité Autonome (PRISSMA)" involves a large consortium of French industrial and academic actors. The ambition of this project is to develop new technologies and new approaches for the certification of safety and reliability of highly automated mobility systems within the corresponding legal framework. Within this global project, Inria Grenoble-Alpes investigates new approaches to validate advanced perception software such as the [CMCDOT](#) software stack. In this purpose, the framework for [Augmented Reality \(AR\)](#) on sensor data, which is a contribution of the present thesis within the project [IRT NANOEEC – SVA](#), has been reused and has a significant role in this project. Another contribution of the present thesis within the project [IRT NANOEEC – SVA](#), the [PCD](#) based collision avoidance has been involved in this project. Within the project [PRISSMA](#), Inria has realized testing campaigns that involved the [AR](#) framework, the [PCD](#) and the [experimental automated car](#) to constitute a proof of concept of the researched validation strategies.

6.3 Future works

The contributions of this thesis enable several potential future works, extensions, follow-up works and investigations of new opportunities. First, the **PCD** could be proposed as a standard interface between grid-based perception systems and sampling-based planners. Yet, for this, it would have to be evaluated with other grid-based perception and planning approaches. The Human-aware navigation of the **PCD** should be evaluated in presence of real humans. The multi-class occupancy prediction could also integrate dedicated motion prediction models for other classes of agents, such as cyclists and cars. It would enable the **PCD** based navigation to handle appropriately urban navigation scenarios that often combine various actors of different classes. Then the proposed test framework with **AR** on sensor data has been presented in this thesis with the augmentation of LiDAR point clouds. Yet it could include the augmentation of other sensing modalities. Besides, **AR** on sensor data is a only complementary tool for evaluation that must be used along with tests in simulation and real world testing. Therefore, to get the best out of it, it would require to design new validation strategies that combine tests in simulation, in Augmented Reality and in reality. More globally, this thesis has mainly considered Bayesian Occupancy Grids but the methods could be extended to other types of probabilistic **Dynamic Occupancy Grids**. Finally this thesis has proposed a new local planning solution, with a probabilistic and particle-based model of dynamic agents. This same approach could be extended to global planning and decision.

References

- [A20] Szilárd Aradi. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):740–759, 2020.
- [A21] Tesla et son autopilot visés par une enquête après une série de onze accidents aux états-unis. Le Monde, Associated Press (AP), Agence France Presse (AFP), 2021.
- [A23] ABDynamics. www.abdynamics.com, 2023.
- [AD10] Albert Albers and Tobias Düser. Implementation of a vehicle-in-the-loop development and validation platform. In *FISITA 2010 World Automotive Congress*, Budapest, Hungary, 2010.
- [AGA23] Imane Argui, Maxime Gueriau, and Samia Ainouz. A mixed-reality framework based on depth camera for safety testing of autonomous navigation systems. In *26th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023.
- [B19] Bradley Berman. The key to autonomous vehicle safety is ODD. SAE International, <https://www.sae.org/news/2019/11/odds-for-av-testing>, 2019.
- [B22] Keith Barry. Federal Regulators Investigate Honda Unexpected Braking Complaints. Consumer Reports, 2022.
- [BAB⁺22] Manuel Boldrer, Alessandro Antonucci, Paolo Bevilacqua, Luigi Palopoli, and Daniele Fontanelli. Multi-agent navigation in human-shared environments: A safe and socially-aware approach. *Robotics and Autonomous Systems*, 149:103979, 2022.
- [BMF07] Thomas Bokc, Markus Maurer, and Georg Farber. Validation of the vehicle in the loop (vil); a milestone for the simulation of driver assistance systems. In *2007 IEEE Intelligent Vehicles Symposium*, pages 612–617, 2007.
- [BW17] Ernst Bosina and Ulrich Weidmann. Estimating pedestrian speed using aggregated literature data. volume 468, pages 1–29, 2017.
- [C20] Ashwini Choudhary. Urban autonomy: The next stage of self-driving. Forbes, 2020.
- [CCMZ18] Jiyu Cheng, Hu Cheng, Max Q.-H. Meng, and Hong Zhang. Autonomous navigation by mobile robots in human environments: A survey. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018.
- [CCX⁺20] Yu Chen, Shitao Chen, Tong Xiao, Songyi Zhang, Qian Hou, and Nanning Zheng. Mixed test environment-based vehicle-in-the-loop validation - a new testing approach for autonomous vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1283–1289, 2020.

- [DCJC08] Damion D Dunlap, Emmanuel G Collins Jr, and Charmane V Caldwell. Sampling based model predictive control with application to autonomous vehicle guidance. In *Florida Conference on Recent Advances in Robotics*, 2008.
- [DZSGE⁺23] Manuel Alejandro Diaz-Zapata, David Sierra González, Özgür Erkent, Jilles Dibangoye, and Christian Laugier. Laptnet-fpn: Multi-scale lidar-aided projective transform network for real time semantic grid prediction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [ECH21] Michael Everett, Yu Fan Chen, and Jonathan P. How. Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9:10357–10377, 2021.
- [ETS19] ITS ETSI. Intelligent transport system (its); vehicular communications; basic set of applications; analysis of the collective-perception service (cps). *Intelligent Transport Systems (ITS)*, 2019.
- [EWL19] Özgür Erkent, Christian Wolf, and Christian Laugier. End-to-End Learning of Semantic Grid Estimation Deep Neural Network with Occupancy Grids. *Unmanned systems*, 7(3):171–181, July 2019.
- [F09] Chiara Fulgenzi. *Autonomous navigation in dynamic uncertain environment using probabilistic models of perception and collision risk prediction*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2009.
- [FBH⁺18] Jaime F. Fisac, Andrea Bajcsy, Sylvia L. Herbert, David Fridovich-Keil, Steven Wang, Claire J. Tomlin, and Anca D. Dragan. Probabilistically safe robot planning with confidence-based human predictions, 2018.
- [FBT97] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 1997.
- [FGS13] Gonzalo Ferrer, Anaís Garrell, and Alberto Sanfeliu. Social-aware robot navigation in urban environments. In *2013 European Conference on Mobile Robots*, pages 331–336, 2013.
- [FS98] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.
- [FS14] Gonzalo Ferrer and Alberto Sanfeliu. Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.
- [FSL09] Chiara Fulgenzi, Anne Spalanzani, and Christian Laugier. Probabilistic motion planning among moving obstacles following typical motion patterns. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

- [FYX⁺18] Yiheng Feng, Chunhui Yu, Shaobing Xu, Henry X. Liu, and Hwei Peng. An augmented reality environment for connected and automated vehicle testing and evaluation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.
- [FZY⁺20] Jin Fang, Dingfu Zhou, Feilong Yan, Tongtong Zhao, Feihu Zhang, Yu Ma, Liang Wang, and Ruigang Yang. Augmented lidar simulator for autonomous driving. *IEEE Robotics and Automation Letters*, 5(2):1931–1938, 2020.
- [GAS17] Colin Grover, Matthew Avery, and Jordan Silberling. 3D car target for future vehicle testing. In *Intern. Technical Conf. on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, 2017.
- [GGLL21] Andrés Gómez, Thomas Genevois, Jerome Lussereau, and Christian Laugier. Dynamic and static object detection considering fusion regions and point-wise features, 2021.
- [GHRL22] Thomas Genevois, Jean-Baptiste Horel, Alessandro Renzaglia, and Christian Laugier. Augmented reality on lidar data: Going beyond vehicle-in-the-loop for automotive software validation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2022.
- [GPMN16] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems (ITSC)*, 2016.
- [GRSL23] Thomas Genevois, Lukas Rummelhard, Anne Spalanzani, and Christian Laugier. From probabilistic occupancy grids to versatile collision avoidance using predictive collision detection. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023.
- [GS12] Anais Garrell and Alberto Sanfeliu. Cooperative social robots to accompany groups of people. *The International Journal of Robotics Research*, 31(13):1675–1701, 2012.
- [GSL23a] Thomas Genevois, Anne Spalanzani, and Christian Laugier. Interaction-aware motion planning as an alternative to proactive planning for human-aware navigation. *IEEE IV Workshop on Interaction-driven Behavior Prediction and Planning for Autonomous Vehicles*, 2023. Invited talk.
- [GSL23b] Thomas Genevois, Anne Spalanzani, and Christian Laugier. Interaction-aware predictive collision detector for human-aware collision avoidance. In *IEEE Intelligent Vehicles Symposium (IV)*, 2023.
- [H22] Andrew J. Hawkins. Tesla’s ‘phantom braking’ problem is getting worse, and the us government has questions. *The Verge*, 2022.
- [HE21] Carl Hildebrandt and Sebastian Elbaum. World-in-the-loop simulation for autonomous systems validation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10912–10919, 2021.

- [HLM⁺22] Jean-Baptiste Horel, Christian Laugier, Lina Marsson, Radu Mateescu, Lucie Muller, Anshul Paigwar, Alessandro Renzaglia, and Wendelin Serwe. Using formal conformance testing to generate scenarios for autonomous vehicles. In *Design, Automation and Test in Europe-Autonomous Systems Design (DATE/ASD)*. IEEE, 2022.
- [HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [HYE07] SN Huang, JK Yang, and Fredrik Eklund. Analysis of car-pedestrian impact scenarios for the evaluation of a pedestrian sensor system based on the accident data from sweden. In *2nd International Conference on ESAR "Expert Symposium on Accident Research"*, pages 136–143, 2007.
- [K86] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986.
- [KA17] Harmish Khambhaita and Rachid Alami. Assessing the social criteria for human-robot collaborative navigation: A comparison of human-aware navigation planners. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017.
- [KSM22] Maria Kabtoul, Anne Spalanzani, and Philippe Martinet. Proactive And Smooth Maneuvering For Navigation Around Pedestrians. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.
- [L08] David Luebke. Cuda: Scalable parallel programming for high-performance scientific computing. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2008.
- [L23] Corine Lesnes. Dans les rues de san francisco, les robots-taxis, sans humain au volant, sont arrivés... et ils ne font pas l'unanimité. *Le Monde*, 2023.
- [LCB⁺18] Yuanfu Luo, Panpan Cai, Aniket Bera, David Hsu, Wee Sun Lee, and Dinesh Manocha. Porca: Modeling and planning for autonomous driving among many pedestrians. *IEEE Robotics and Automation Letters*, 3, 2018.
- [LCB20] Yanjie Liu, Jiao Chen, and Xinyu Bai. An approach for multi-objective obstacle avoidance using dynamic occupancy grid map. In *IEEE International Conference on Mechatronics and Automation (ICMA)*, 2020.
- [LCL⁺21] Shuijing Liu, Peixin Chang, Weihang Liang, Neeloy Chakraborty, and Katherine Driggs-Campbell. Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [LK99] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. In *IEEE International Conference on Robotics and Automation*, 1999.

- [LK19] Kibeom Lee and Dongsuk Kum. Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map. *IEEE Access*, 2019.
- [LLG18] Thomas Laverghetta, James F. Leathrum, and Nathan D. Gonda. Integrating virtual and augmented reality based testing into the development of autonomous vehicles. In *MODSIM 2018*, 2018.
- [LRK⁺21] Johann Laconte, Elie Randriamiarintsoa, Abderrahim Kasmi, François Pomerleau, Roland Chapuis, Christophe Debain, and Romuald Aufrère. Dynamic lambda-field: A counterpart of the bayesian occupancy grid for risk assessment in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [M21] Tambiana Madiega. Artificial intelligence act. *European Parliament: European Parliamentary Research Service*, 2021.
- [M23] Mercedes-Benz Group. Autonomous and safe. <https://group.mercedes-benz.com/innovation/case/autonomous/automated-driving-in-urban-traffic.html>, 2023.
- [MCT⁺17] Hafida Mouhagir, Véronique Cherfaoui, Reine Talj, François Aioun, and Franck Guillemard. Using evidential occupancy grid for vehicle trajectory planning under uncertainty with tentacles. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017.
- [MHG⁺09] Mehdi Moussaïd, Dirk Helbing, Simon Garnier, Anders Johansson, Maud Combe, and Guy Theraulaz. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proceedings of the Royal Society B: Biological Sciences*, 2009.
- [N17] Dominik Nuss. *A random finite set approach for dynamic occupancy grid maps*. PhD thesis, Universität Ulm, 2017.
- [NHT23] National Highway Traffic Safety Administration NHTSA. Automated vehicles for safety. <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>, 2023.
- [NRL14] Amaury Nègre, Lukas Rummelhard, and Christian Laugier. Hybrid sampling bayesian occupancy filter. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1307–1312, 2014.
- [NVI23] NVIDIA. *CUDA C++ Programming Guide*, 2023.
- [O23] OSRF. *Why gazebo?*, 2023.
- [P21] Manon Prédhumeau. Simulating Realistic Pedestrian Behaviors in the Context of Autonomous Vehicles in Shared Spaces. In *20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2021.

- [PCD⁺23] Ashish Pandharipande, Chih-Hong Cheng, Justin Dauwels, Sevgi Z Gurbuz, Javier Ibanex-Guzman, Guofa Li, Andrea Piazzoni, Pu Wang, and Avik Santra. Sensing and machine learning for automotive perception: A review. *IEEE Sensors Journal*, 2023.
- [PCL20] Changwoo Park, Seunghwan Chung, and Hyeongcheol Lee. Vehicle-in-the-loop in global coordinates for advanced driver assistance system. *Applied Sciences*, 2020.
- [PDS19] Manon Prédhumeau, Julie Dugdale, and Anne Spalanzani. Adapting the social force model for low density crowds in open environments. In *Conference of the European Social Simulation Association*, pages 519–531. Springer, 2019.
- [PK05] Mihail Pivtoraiko and Alonzo Kelly. Efficient constrained path planning via search in state lattices. In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, 2005.
- [PMDS21] Manon Prédhumeau, Lyuba Mancheva, Julie Dugdale, and Anne Spalanzani. An Agent-Based Model to Predict Pedestrians Trajectories with an Autonomous Vehicle in Shared Spaces. In *AAMAS 2021 - 20th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2021.
- [PR22] Steven Posada and Stephen Ridella. Engineering analysis on autopilot & first responder scenes. Technical Report EA 22-002, National Highway Traffic Safety Administration, 2022.
- [PS05] R. Philippsen and R. Siegwart. An interpolated dynamic navigation function. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3782–3789, 2005.
- [QBR⁺18] Jean Quilbeuf, Mathieu Barbier, Lukas Rummelhard, Christian Laugier, Axel Legay, Blanche Baudouin, Thomas Genevois, Javier Ibañez-Guzmán, and Olivier Simonin. Statistical Model Checking Applied on Perception and Decision-making Systems for Autonomous Driving. In *PPNIV 2018 - 10th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pages 1–8, Madrid, Spain, October 2018.
- [R22] Olivier Razemon. Mobilité : le flop des navettes autonomes, desservies par les problèmes techniques. *Le Monde*, 2022.
- [RDGML22] Lukas Rummelhard, Jean-Alix David, Andres Gonzalez Moreno, and Christian Laugier. A cross-prediction, hidden-state-augmented approach for dynamic occupancy grid filtering. In *IEEE International Conference on Control Automation Robotics & Vision (ICARCV)*, 2022.

- [RFNV21] Nijat Rajabli, Francesco Flammini, Roberto Nardone, and Valeria Vittorini. Software verification and validation of safe autonomous cars: A systematic literature review. *IEEE Access*, 9:4797–4819, 2021.
- [RGD⁺18] Lukas Rummelhard, Thomas Genevois, Jean-Alix David, Amaury Nègre, and Christian Laugier. Real-time Ground Estimation and Point Cloud Segmentation. GTC - GPU Technology Conference, March 2018. Poster, available at <https://hal.archives-ouvertes.fr/hal-01903668>.
- [RNL15] Lukas Rummelhard, Amaury Nègre, and Christian Laugier. Conditional monte carlo dense occupancy tracker. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2485–2490. IEEE, 2015.
- [RPNL17] Lukas Rummelhard, Anshul Paigwar, Amaury Nègre, and Christian Laugier. Ground estimation and point cloud segmentation using spatiotemporal conditional random field. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017.
- [S19] Richard Schram. Euro ncap’s first step to assess automated driving systems. In *Euro NCAP Working Group on Automated Driving*, 2019.
- [S21] Zsolt Szalay. Next generation x-in-the-loop validation methodology for automated vehicle systems. *IEEE Access*, 2021.
- [SBGD22] Marcel Schreiber, Vasileios Belagiannis, Claudius Gläser, and Klaus Dietmayer. A multi-task recurrent neural network for end-to-end dynamic occupancy grid mapping, 2022.
- [SCL⁺23] Yuan Shen, Bhargav Chandaka, Zhi-hao Lin, Albert Zhai, Hang Cui, David Forsyth, and Shenlong Wang. Sim-on-wheels: Physical world in the loop simulation for self-driving. *arXiv preprint arXiv:2306.08807*, 2023.
- [SCLC20] Dan Shen, Yaobin Chen, Lingxi Li, and Stanley Chien. Collision-free path planning for automated vehicles risk assessment via predictive occupancy map. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.
- [SGSM⁺22] Luiz Alberto Serafim Guardini, Anne Spalanzani, Philippe Martinet, Christian Laugier, Thomas Genevois, and Anh-Lam Do. Minimal injury risk motion planning using active mitigation and sampling model predictive control. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2022.
- [SK16] B. Siciliano and O. Khatib. Robotics foundations. In *Springer Handbook of Robotics*. Springer International Publishing, 2016.
- [SSSS17] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars, 2017.
- [TK10] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

- [TSFA21] Phani Teja Singamaneni, Anthony Favier, and Rachid Alami. Human-aware navigation planner for diverse human-robot interaction contexts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [VRGBC21] Jose Roberto Vargas Rivero, Thiemo Gerbich, Boris Buschardt, and Jia Chen. Data augmentation of automotive lidar point clouds under adverse weather situations. *Sensors*, 21(13):4503, 2021.
- [W22] Matthew Wansley. The auto safety revolution. Social Science Research Network (SSRN), 2022.
- [W23] Volkswagen Group. Autonomous driving in the city: the hardest challenge. <https://modo.volkswagengroup.it/en/lab/autonomous-driving-in-the-city-the-hardest-challenge>, 2023.
- [WDG⁺16] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [YLCT20] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 2020.
- [ZEF⁺18] Marc René Zofka, Marc Essinger, Tobias Fleck, Ralf Kohlhaas, and J. Marius Zöllner. The sleepwalker framework: Verification and validation of autonomous vehicles by mixed reality lidar stimulation. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, pages 151–157, 2018.

Appendices

A Experimental platforms

A.1 Experimental automated car

The software contributions of this thesis have mostly been designed for deployment on [Automated Driving Systems \(ADS\)](#). For this, they have been developed and tested on a prototype of autonomous car which is displayed in [fig.52](#).



Figure 52 – The automated car prototype that has been used to experiment and demonstrate the software contributions of this thesis in the domain of [ADS](#).

This experimental platform is based on a regular commercial car, a Renault Zoé. This car has been modified such that it can be controlled from software, either for a fully automated drive, either for a shared drive between software and a human driver to experiment [Advanced Driver Assistance Systems \(ADAS\)](#). A regular computer is used to execute software, it is equipped with a NVIDIA Titan X GPU. For sensing, this vehicle is equipped with a Velodyne HDL64 LiDAR on its roof and 4 Ibeo Lux LiDARs that are placed at the height of the bumpers, 3 on the front, 1 on the rear. This set of LiDARs provides a dense observation of the surroundings at 10Hz. A monocular camera is placed inside the cabin, above the steering wheel and facing front, it is used to visualize the scene in recordings of the experiments but it is not used for perception. An Extended Kalman Filter fuses the outputs of an IMU, a RTK GPS and the wheels speeds to provide an accurate estimation of position, orientation and speed of the vehicle. A virtual twin of this vehicle has been developed with the Gazebo simulator [O23]. This twin provides localization and sensing information with the same message format than the actual vehicle. It is also driven with the same control messages. So this simulation environment can be used for [Software-in-the-Loop \(SiL\)](#) and [Hardware-in-the-Loop](#)

(HiL) simulations. An example of a simulated environment is presented in fig.53.



Figure 53 – Visualization of a simulated environment with the virtual twin of the *experimental automated car* (in gray, in the center of the image).

This experimental platform and its development have been funded by the French program ”Investissement d’Avenir” ANR-10-AIRT-05 within the scope of the project [IRT NANO-ELEC – SVA](#). The vehicle is used to develop, test and demonstrate advanced automotive software in real-world experiments. Fig.54 displays images from a test session. A focus has been made on Bayesian perception and its application to collision detection and autonomous drive, in complex urban environments. In previous works, the [Conditionnal Monte-Carlo Dense Occupancy Tracker \[RNL15\] \(CMCDOT\)](#) perception suite has been deployed on this vehicle. Within the present thesis, this experimental platform has been used to develop and evaluate the collision avoidance system which is presented in chapters 3 and 4. The proposed Augmented Reality testing framework that is introduced in chapter 5 has also been deployed and validated on this vehicle.

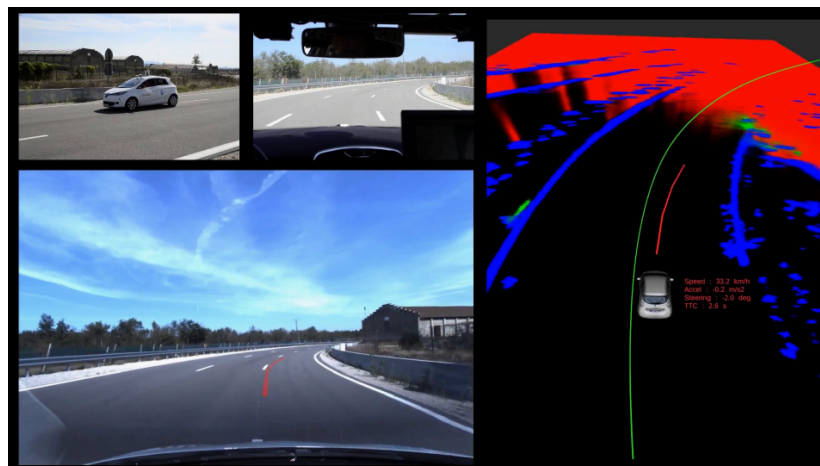


Figure 54 – Different points of view during a real-world path following experiment with the *experimental automated car*.

A.2 Urban light vehicle prototype

This thesis considered the application of advanced driving systems on a light urban vehicle that differs from the [experimental automated car](#) because it operates at lower driving speed, with different kinematics and with limited sensing and computing capabilities. This vehicle has been designed by an industrial partner as a prototype for a urban micro-mobility vehicle. It can either be manually driven or either be autonomous. The vehicle is presented on [fig.55](#).

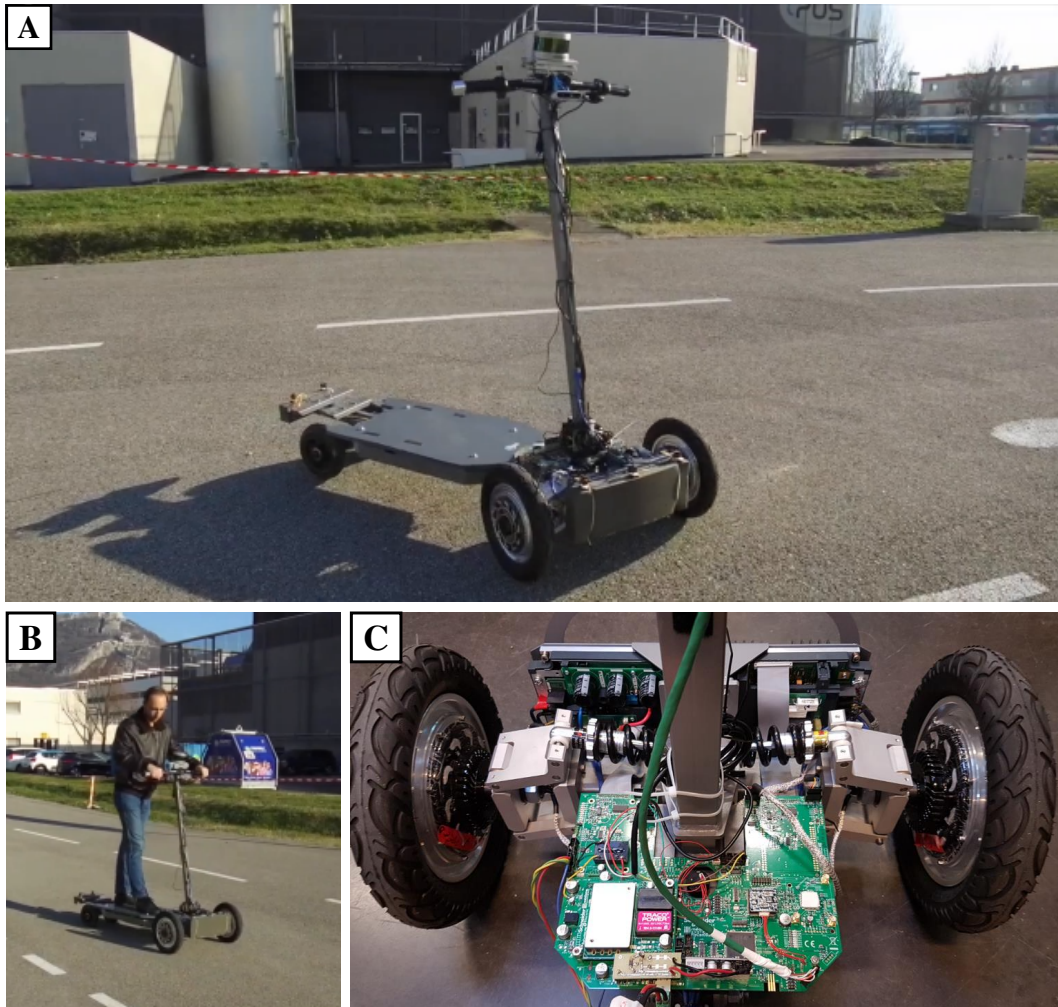


Figure 55 – (A) introduces the prototype of urban micro-mobility vehicle on which has been deployed the collision avoidance system that has been proposed within this thesis. (B) displays the vehicle with a passenger. (C) presents the front part of the vehicle, with the ECU.

This vehicle has specific kinematics. The front part of the vehicle, with the ECU, motors, batteries and sensors, is moved by 2 wheels that control motion and orientation in a differential steering fashion. Then there is a passive trailer where the passenger or the payload may stand. The computing hardware for navigation is a NVIDIA Jetson TX2, a small computer with a GPU. This small computer has been well embedded in the vehicle and has a reduced power consumption however it offers limited computing capabilities, specially in

terms of RAM and CPU cores. For its sensing of the environment, this vehicle relies on a set of ultrasonic range sensors for the close range, a VLP16 Velodyne LiDAR for perception of cars, cyclist and pedestrians and a Intel Realsense 435D stereo camera for more detailed perception in front of the vehicle. The perception uses the [CMCDOT](#) software suite to fuse the observations from these different sensing modalities. The vehicle is equipped with an IMU and a GPS that are fused within an Extended Kalman Filter to obtain a consistent estimation of position, orientation and speed. However this localization remains vague compared to the one of the [experimental automated car](#). Similarly to the virtual twin of the [experimental automated car](#), a virtual twin of this vehicle has been developed within the simulator Gazebo [O23] and it permits [SiL](#) and [HiL](#) simulation. An example of this simulation environment is shown in [fig.56](#).

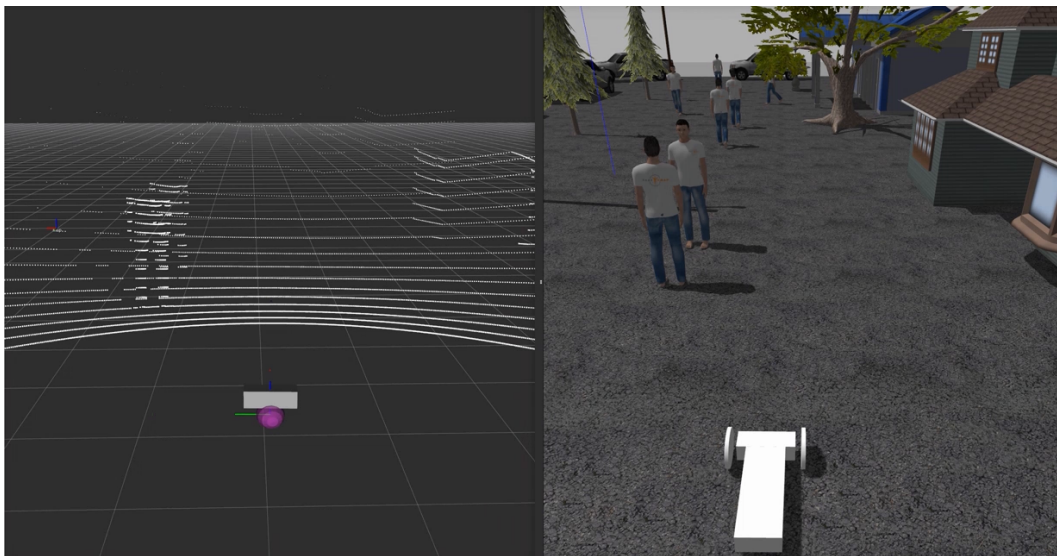


Figure 56 – Visualization of a simulated environment with the virtual twin of the [urban light vehicle prototype](#). The image on the left shows the emulated point cloud from the VLP16. The image on the right displays the simulated environment with the vehicle on the bottom.

So this light vehicle is an experimental platform for low speed navigation in urban environment, with sensing, computing and kinodynamic constraints that differ from cars. This vehicle and its development have been funded by the French program "Investissement d'Avenir" ANR-10-AIRT-05 within the scope of the project [IRT Nanoelec – Light Vehicle Demonstrator](#). Within the present thesis, this vehicle has been used to test the collision avoidance system which is presented in [chapter 3](#) under the specific constraints of this vehicle.

A.3 Industrial logistics robot

A third vehicle has been considered within this thesis. It is a small indoor robot which has been realized by an industrial partner with the purpose to convey goods in industrial logistics applications. The specificity of this robot is that it is made to be rather cheap but at the same time to be versatile. The robot itself is displayed in [fig.57](#).



Figure 57 – The prototype of industrial indoor robot that has been used to experiment and demonstrate collision avoidance software contributions of this thesis. This robot is 90 cm long and 60 cm wide.

This robot is rather small, 90 cm long and 60 cm wide, it has unicycle-like kinematics with differential steering. A NVIDIA Jetson TX2 is embedded inside it and used as computing hardware for navigation, it provides limited computing capabilities in terms of CPU and RAM. In order to ensure navigation in various environments, this robot is equipped with a multi-modal set of sensors, 2 planar LiDARs, 4 ultrasonic range sensors and a Intel Realsense L515 LiDAR depth camera. Fused with the [CMCDOT](#) perception suite, this enables versatile perception of humans, other agents and static obstacles that may be transparent or at different heights. Localization is achieved by Simultaneous Localization And Mapping (SLAM). Unfortunately this robot has not been physically available for test during the realization of the present thesis. A virtual twin of this robot has been developed within the Gazebo simulator [O23], it permitted [HiL](#) simulated tests in a realistic environment instead of actual testing with the robot. The simulated test environment is displayed in [fig.58](#).

This robot is another experimental platform that differs from the [experimental automated car](#). It has the same computing hardware than the [urban light vehicle prototype](#) and a similar set of sensor. It also navigates at similar speeds. But it operates in a different environment that presents very different constraints for navigation. This vehicle and its development have been funded by "La Région Auvergne-Rhône-Alpes" within the scope of the project [R&D Booster MoovIT](#). Within the present thesis, this experimental platform has contributed to the evaluation of the collision avoidance system which is presented in chapters [3](#) and [4](#).

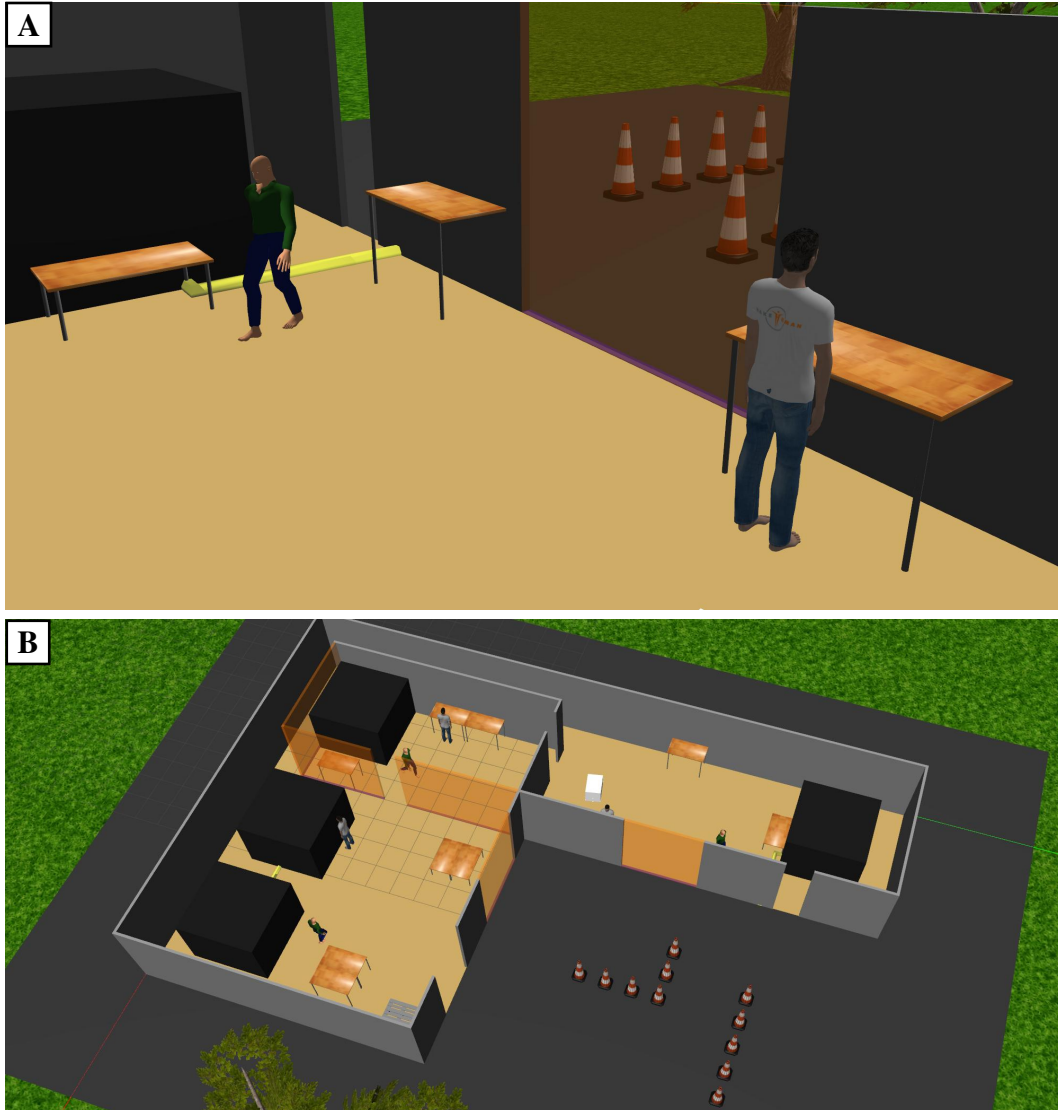


Figure 58 – (A) presents a detail of the simulated test environment of the *industrial logistics robot*. It contains various obstacles for the navigation of this robot, including standing and walking humans, transparent obstacles (not visible to LiDARs), low obstacles (below the planar LiDARs), high obstacles (tables, above the planar LiDARs) and slopes. (B) presents a global view of this environment. It is intended to reproduce the environment of a small factory. This view presents few humans but tests included more crowded environments.